

ARTIFICIAL INTELLIGENCE IN BIOINFORMATICS AND DRUG REPURPOSING: METHODS AND APPLICATIONS

EDITED BY: Pan Zheng, Shudong Wang, Xun Wang and Xiangxiang Zeng
PUBLISHED IN: *Frontiers in Genetics*



frontiers

Frontiers eBook Copyright Statement

The copyright in the text of individual articles in this eBook is the property of their respective authors or their respective institutions or funders. The copyright in graphics and images within each article may be subject to copyright of other parties. In both cases this is subject to a license granted to Frontiers.

The compilation of articles constituting this eBook is the property of Frontiers.

Each article within this eBook, and the eBook itself, are published under the most recent version of the Creative Commons CC-BY licence.

The version current at the date of publication of this eBook is CC-BY 4.0. If the CC-BY licence is updated, the licence granted by Frontiers is automatically updated to the new version.

When exercising any right under the CC-BY licence, Frontiers must be attributed as the original publisher of the article or eBook, as applicable.

Authors have the responsibility of ensuring that any graphics or other materials which are the property of others may be included in the CC-BY licence, but this should be checked before relying on the CC-BY licence to reproduce those materials. Any copyright notices relating to those materials must be complied with.

Copyright and source acknowledgement notices may not be removed and must be displayed in any copy, derivative work or partial copy which includes the elements in question.

All copyright, and all rights therein, are protected by national and international copyright laws. The above represents a summary only. For further information please read Frontiers' Conditions for Website Use and Copyright Statement, and the applicable CC-BY licence.

ISSN 1664-8714

ISBN 978-2-88974-881-5

DOI 10.3389/978-2-88974-881-5

About Frontiers

Frontiers is more than just an open-access publisher of scholarly articles: it is a pioneering approach to the world of academia, radically improving the way scholarly research is managed. The grand vision of Frontiers is a world where all people have an equal opportunity to seek, share and generate knowledge. Frontiers provides immediate and permanent online open access to all its publications, but this alone is not enough to realize our grand goals.

Frontiers Journal Series

The Frontiers Journal Series is a multi-tier and interdisciplinary set of open-access, online journals, promising a paradigm shift from the current review, selection and dissemination processes in academic publishing. All Frontiers journals are driven by researchers for researchers; therefore, they constitute a service to the scholarly community. At the same time, the Frontiers Journal Series operates on a revolutionary invention, the tiered publishing system, initially addressing specific communities of scholars, and gradually climbing up to broader public understanding, thus serving the interests of the lay society, too.

Dedication to Quality

Each Frontiers article is a landmark of the highest quality, thanks to genuinely collaborative interactions between authors and review editors, who include some of the world's best academicians. Research must be certified by peers before entering a stream of knowledge that may eventually reach the public - and shape society; therefore, Frontiers only applies the most rigorous and unbiased reviews. Frontiers revolutionizes research publishing by freely delivering the most outstanding research, evaluated with no bias from both the academic and social point of view. By applying the most advanced information technologies, Frontiers is catapulting scholarly publishing into a new generation.

What are Frontiers Research Topics?

Frontiers Research Topics are very popular trademarks of the Frontiers Journals Series: they are collections of at least ten articles, all centered on a particular subject. With their unique mix of varied contributions from Original Research to Review Articles, Frontiers Research Topics unify the most influential researchers, the latest key findings and historical advances in a hot research area! Find out more on how to host your own Frontiers Research Topic or contribute to one as an author by contacting the Frontiers Editorial Office: frontiersin.org/about/contact

ARTIFICIAL INTELLIGENCE IN BIOINFORMATICS AND DRUG REPURPOSING: METHODS AND APPLICATIONS

Topic Editors:

Pan Zheng, University of Canterbury, New Zealand

Shudong Wang, China University of Petroleum, Huadong, China

Xun Wang, China University of Petroleum, Huadong, China

Xiangxiang Zeng, Hunan University, China

Citation: Zheng, P., Wang, S., Wang, X., Zeng, X., eds. (2022). Artificial Intelligence in Bioinformatics and Drug Repurposing: Methods and Applications. Lausanne: Frontiers Media SA. doi: 10.3389/978-2-88974-881-5

Table of Contents

04	<i>Editorial: Artificial Intelligence in Bioinformatics and Drug Repurposing: Methods and Applications</i>
	Pan Zheng, Shudong Wang, Xun Wang and Xiangxiang Zeng
08	<i>Repositioning Drugs to the Mitochondrial Fusion Protein 2 by Three-Tunnel Deep Neural Network for Alzheimer's Disease</i>
	Xun Wang, Yue Zhong and Mao Ding
17	<i>Medical Image Protection Algorithm Based on Deoxyribonucleic Acid Chain of Dynamic Length</i>
	Xianglian Xue, Haiyan Jin, Dongsheng Zhou and Changjun Zhou
35	<i>Multi-Omics Data Fusion via a Joint Kernel Learning Model for Cancer Subtype Discovery and Essential Gene Identification</i>
	Jie Feng, Limin Jiang, Shuhao Li, Jijun Tang and Lan Wen
45	<i>iDNA-MT: Identification DNA Modification Sites in Multiple Species by Using Multi-Task Learning Based a Neural Network Tool</i>
	Xiao Yang, Xiucai Ye, Xuehong Li and Lesong Wei
56	<i>GADTI: Graph Autoencoder Approach for DTI Prediction From Heterogeneous Network</i>
	Zhixian Liu, Qingfeng Chen, Wei Lan, Haiming Pan, Xinkun Hao and Shirui Pan
67	<i>CLGBO: An Algorithm for Constructing Highly Robust Coding Sets for DNA Storage</i>
	Yanfen Zheng, Jieqiong Wu and Bin Wang
81	<i>Stable DNA Sequence Over Close-Ending and Pairing Sequences Constraint</i>
	Xue Li, Ziqi Wei, Bin Wang and Tao Song
96	<i>NMFNA: A Non-negative Matrix Factorization Network Analysis Method for Identifying Modules and Characteristic Genes of Pancreatic Cancer</i>
	Qian Ding, Yan Sun, Junliang Shang, Feng Li, Yuanyuan Zhang and Jin-Xing Liu
107	<i>Graph Neural Networks and Their Current Applications in Bioinformatics</i>
	Xiao-Meng Zhang, Li Liang, Lin Liu and Ming-Jing Tang
129	<i>Improving de novo Molecule Generation by Embedding LSTM and Attention Mechanism in CycleGAN</i>
	Feng Wang, Xiaochen Feng, Xiao Guo, Lei Xu, Liangxu Xie and Shan Chang
143	<i>Locally Adjust Networks Based on Connectivity and Semantic Similarities for Disease Module Detection</i>
	Jia Liu, Huole Zhu and Jianfeng Qiu
153	<i>Deep Learning Algorithms Achieved Satisfactory Predictions When Trained on a Novel Collection of Anticoronavirus Molecules</i>
	Emna Harigua-Souiai, Mohamed Mahmoud Heinhane, Yosser Zina Abdelkrim, Oussama Souiai, Ines Abdeljaoued-Tej and Ikram Guizani



Editorial: Artificial Intelligence in Bioinformatics and Drug Repurposing: Methods and Applications

Pan Zheng^{1*}, Shudong Wang², Xun Wang² and Xiangxiang Zeng³

¹Department of Accounting and Information Systems, University of Canterbury, Christchurch, New Zealand, ²College of Computer Science and Technology, China University of Petroleum, Qingdao, China, ³Department of Computer Science, Hunan University, Changsha, China

Keywords: artificial intelligence, neural network, bioinformatics, drug repurposing, drug design

Editorial on the Research Topic

Artificial Intelligence in Bioinformatics and Drug Repurposing: Methods and Applications

INTRODUCTION

The development of Artificial Intelligence (AI) pushes the boundaries of new computing paradigms to become actual realities for many science and engineering challenges. The delicacy and agility of a computing instrument do not mean anything when we cannot use it to create value and solve problems. Machine learning, a trendy subfield of AI, focuses on extracting and identifying insightful and actionable information from big and complex data using different types of neural networks. Data-hungry by its nature, machine learning algorithms usually excel in the practical fields that generate and possess abundant data. The two application areas we are interested in are drug repositioning and bioinformatics, both of which are the very fields often producing a large volume of data. This thematic issue aims to cover the recent advancement in artificial intelligence methods and applications that are developed and introduced in the field of bioinformatics and drug repurposing and provide a comprehensive and up-to-date collection of research and experiment works in these areas.

OPEN ACCESS

Edited and reviewed by:

Richard D. Emes,
University of Nottingham,
United Kingdom

*Correspondence:

Pan Zheng
pan.zheng@canterbury.ac.nz

Specialty section:

This article was submitted to
Computational Genomics,
a section of the journal
Frontiers in Genetics

Received: 07 February 2022

Accepted: 21 February 2022

Published: 17 March 2022

Citation:

Zheng P, Wang S, Wang X and Zeng X
(2022) Editorial: Artificial Intelligence in
Bioinformatics and Drug Repurposing:
Methods and Applications.
Front. Genet. 13:870795.
doi: 10.3389/fgene.2022.870795

BIOINFORMATICS

Bioinformatics is an interdisciplinary field that covers a broad spectrum of studies including fields of biology, computer science, information science, and statistics. DNA and gene data sequencing (Zou et al., 2016), coding (Yin et al., 2021), modification (Li et al., 2015; Shi et al., 2016), and structure analysis (Zhang et al., 2014; Hong et al., 2020) are predominant areas of bioinformatics. This issue includes six contributions to this topic.

DNA sequences react with each other due to complementary pairing, which reduces the number of DNA sequences that could be used for molecular hybridization. If the A-T base is located at one end of the DNA sequence and its complementary sequence, it results in a gap and leads to the decreasing of the accuracy of the calculation. Pairing Sequences Constraint (PSC) and Close-ending along with the Improved Chaos Whale (ICW) optimization algorithm are proposed by Li et al. in this Research Topic to effectively resolve the issues. The proposed method

is compared with the contemporary methods of NACST/Seq, DEPT, MO-ABC, pMO-ABC, and HSWOA on parameters such as continuity, hairpin structure, H-measure, similarity, and melting temperature. The experiment shows that the method demonstrates superior results and the ability to avoid secondary structures.

The volume of the data is growing exponentially in the big data era. Data storage is becoming a pressing challenge. The unique design and molecular structure of deoxyribonucleic acid (DNA) provide us with a mechanism to encode and store humongous amounts of data very efficiently. Zheng et al. present an algorithm of coding sets for DNA storage. It is essentially a variant of Gradient-Based Optimizer (GBO) with two mutation strategies for better performance, which are the Cauchy and Levy mutation operators. The performance of the algorithm was evaluated using CEC-2017 test function and the Wilcoxon rank-sum test. It showed that the lower bounds of DNA coding sets constructed by the CLGBO algorithm increased by 4.3–13.5% compared with previous work.

Feng et al. looked at an interesting application of utilizing gene and DNA data for cancer subtype discovery and related gene identification. The study experimented on the data collected from the Broad Institute GDAC Firehose, which contains seven common cancer datasets. Each cancer dataset consists of multiple sources of cancer data including gene expression information, isoform expression information, DNA methylation expression information, and corresponding clinical information. The raw data was initially rescaled by min-max normalization and reduced by kernel PCA. Based on the multi-omics of cancer data, three similarity kernel matrices are constructed through the Gaussian kernel function and fused into a global similarity expression matrix. Eventually, the integrated similarity kernel matrix is fed to spectral clustering, hence the predictive clusters are identified. The performance of the approach demonstrates reasonable superiority in comparison with other similar methods.

Another study of gene data by Ding et al. focuses on identifying modules and characteristic genes of pancreatic cancer. A non-NMF network analysis method (NMFNA) is introduced. Initially, the methylation network (ME), copy number variation (CNV) network, and ME-CNV network are constructed by Pearson correlation coefficient. Incorporating graph-regularized constraints, the networks are further integrated and decomposed to identify modules. Both gene ontology (GO) and pathway enrichment analyses are performed, and characteristic genes are detected by the multi-measure score to understand the biological functions of PC core modules. Compared with similar methods in the literature, the NMFNA identified more PC-related GO terms, pathways, and characteristic genes in core modules during experiments.

The intrinsic nature of the DNA structure inspires the studies of other fields, e.g., computing. Xue et al. developed an image protection algorithm based on the chain of the dynamic length of DNA. The method is tested on images generated by three popular medical imaging modalities: CT, MRI, and X-Ray. In this method, the original image is encoded into a DNA matrix dynamically using a Fractional-Order Chen Hyper Chaotic (FOCHC)

sequence. The DNA matrix is then scrambled by two other FOCHC sequences. DNA dynamical chain operations are carried out by four FOCHC sequences. To decode, the matrix is solved into a binary matrix by a FOCHC sequence, and the encrypted image is obtained after recombining the DNA chain. Eight chaotic sequences are used to complete the whole process. The eight sequences are generated by two FOCHC under different keys, which are produced using the SHA-256 algorithm and the hamming distance. The method is robust and able to be sustained under noise attack, occlusion attack, and all common cryptographic attacks.

Taking advantage of advanced computational methods to analyze DNA data conveys great value in the area of bioinformatics. Yang et al. propose a method that can identify DNA modification sites, N4-methylcytosine (4 mC) and N6-methyladenine (6 mA). The method uses multitask learning integrated with the bidirectional gated recurrent units (BGRU). The original DNA sequences are encoded into matrices by one-hot encoding as the input of the neural network. The encoded matrix is fed to a bidirectional GRU for the different levels of dependency relationships between subsequences. A max-pooling layer is used to find out features playing key roles in DNA methylation site identification in each GRU. The features learned from the max-pooling layer are sent to the task-specific output module to identify DNA modification sites. DNA datasets of four species are experimented with using the method. The performance is significantly better than the other methods using typical text classification.

DRUG REPURPOSING

Drug repurposing is an important topic in the research field of drug discovery and design. It is also known as drug repositioning, reprofiling, or re-tasking (Pushpakom et al., 2019), which is a technique or process of finding novel pharmaceutical applications of existing medicines that are not originally designed for. There are usually three types of studies to develop and examine the efficacy of a drug during the process of drug discovery and design, *in silicon*, *in vitro*, and *in vivo*. With artificial intelligence and machine learning in the role, the researches mainly focus on *in silicon* studies and some *in vitro* for proof of concept. With an increasing number of related drug databases, e.g., DrugBank, DrugMatrix, BindingDB, PubChem, ChEMBL, and KEGG to name a few, drug data analysis and study using artificial neural networks and other machine learning methods become the trend in computer-aided drug design. There are three major types of conventional studies which are Drug-Target Interaction (DTI), Drug-Drug Interaction (DDI), and Protein-Protein Interaction (PPI). Drug-Target Interaction (DTI) is the most popular type of study in this field. It investigates the binding relation, i.e., binding affinity, between the drug and target protein directly. The early studies consider DTI a binary classification problem and later various machine learning methods are used to approach DTI problems (Öztürk et al., 2018; Song et al., 2021). Drug-Drug Interaction (DDI) studies explore the effect variations of a drug when the drug is taken at the same time with another

(Kumar Shukla et al., 2020). Drug interaction profiles can be established to measure drug similarities and associations. It is believed that drug molecules with homogenous structures are probable to react on similar proteins and manifest similar physiological effects. Protein-Protein Interaction (PPI) is another type of study that probes into the drug discovery problem. PPI exists in all the biological and cellular processes, e.g., cell-signaling and cell survival. In drug discovery, finding allosteric sites and hotspots has become a popular topic of PPI studies (Liu et al., 2018; Jin et al., 2021; Yu et al., 2022). Another six contributions of drug repurposing are included in this issue.

Alzheimer's disease (AD) is common dementia that develops among the elderly. The mitochondrial fusion protein 2 (MFN2) is one of the closely relevant proteins which may cause AD. Wang et al. develop a three-tunnel deep neural network model trained on the Davis dataset and deployed it with the DrugBank database to investigate the drug-target binding affinities between drug molecules and MFN2. Fifteen drug molecules were recommended by the neural network model. Molecular docking experiments were carried out on 11 of those whose molecular weights are greater than 200. The result shows that all 11 molecules can dock with the protein successfully and five of them have a great binding effect. This work demonstrates a classical approach of DTI using neural network methods.

Liu et al. propose a novel DTI prediction method (GADTI) using graph convolutional network (GCN) and a random walk with restart (RWR). Data used in this study are first converted in the form of a graph network. DTI predictions are then transformed into link predictions of the network. The overall architecture comprises two main components, an encoder and a decoder. The encoder is formed by the GCN and the RWR, which produces embeddings for nodes of the graph. The decoder is a matrix factorization model using embedding vectors from the encoder to discover and predict DTIs. Experiments show that GADTI has an AUROC value of 0.9582 and AUPRC of 0.8611. Four popular DTI methods are used as benchmarking methods, all of which are inferior to GADTI. It is quite refreshing that the researchers endeavor to explore new neural network methods in the DTI of drug discovery and repurposing.

As graph neural networks (GNNs) gain popularity in solving various practical problems, Zhang et al. present a comprehensive survey of GNNs and their advances in bioinformatics. Three major variants of GNN, Graph Convolutional Networks, Graph Attention Networks, and Graph Autoencoder Networks are reviewed. Typical technical tasks of GNN, node classification, link prediction, and graph generation, are thoroughly discussed. The applications of GNN in literature are categorized in three bioinformatics application aspects, disease prediction, drug discovery, and biomedical imaging. Besides the merits of GNN, the survey addresses the challenges of GNN as well, e.g., data quality and method Interpretability. We noticed that there are several surveys of GNN published from 2018 to 2021, nonetheless, a dedicated GNN survey in the field of bioinformatics was lacking. The contributors of this work timely fill the gap. It has been cited 8 times in the first

6 months of its publication, thus we believe this work will attract more research attention in the future.

How to produce new reasonable molecules with desired pharmacological, physical, and chemical properties is one of the challenges of *de novo* molecular generation. Wang et al. develop a new variant of cycle generative adversarial network (CycleGAN) named LA-CycleGAN which is embedded with Long Short-Term Memory (LSTM) and Attention mechanism for molecule generation with better accuracy. With the new mechanisms added, the neural network is able to overcome long-term dependency problems in treating the commonly used SMILES input. The quantitative evaluation and experiments show that LA-CycleGAN achieves better Tanimoto similarity distribution between the generated molecules and the starting molecules in comparison with a similar variant, Mol-CycleGAN. This study focuses on a specific machine learning method and improves it so the method can better work with molecular generation problems.

Disease module identification is an important step to potential drug targets formation. Liu et al. put forward an effective disease module identification method, IDMCSS. Modifying an existing PPI network, the method adds some potential interactions and removes incorrect interactions based on the connective and semantic similarities between the given disease proteins and their neighboring proteins. The method aims to eliminate the interference of incorrect and missing links contained in the original PPI network for better disease module detection. The method is experimented on an asthma PPI network and compared with four state-of-the-art disease module identification approaches. The disease module identified by IDMCSS includes more proteins that are enriched in asthma-related GO terms, pathways, and differential expression genes than those achieved by the other four approaches.

Drug repurposing against COVID-19 attracts wide research attention in the community. Harigua-Souiai et al. propose a pipeline for Ligand-Based Drug Discovery (LBDD) against SARS-CoV-2. A dataset of 2,610 molecules having anticoronavirus effects is collected and curated. The chemical structures of these molecules were encoded through multiple systems to be readily useful as input of a set of AI methods. Seven machine learning (ML) algorithms and four deep learning (DL) algorithms were used to classify the molecules in active and inactive classes. The seven ML algorithms are Logistic Regression (LR), Support Vector Machine (SVM), Random Forests (RF), Multitask Classifier (MTC), IRV-MTC, Robust MTC, and Gradient Boosting (XGBoost). Four DL methods are the Graph Convolutional Model, the DAG model, the Graph Attention Networks model (GAT), and the GCN model. The Random Forests (RF), Graph Convolutional Network (GCN), and Directed Acyclic Graph (DAG) models achieved the best performances. A further validation experiment revealed a superior potential of DL algorithms to achieve drug repurposing against SARS-CoV-2 based, i.e., GCN and DAG. This study exhaustively used almost all suitable AI

algorithms and methods for the problem and suggested that DL methods are superior to others.

CONCLUSION

After the special thematic issue information was announced as a “research topic” on the webpage of *Frontiers in Genetics* journal, it received overwhelming attention and interest from researchers in the field of bioinformatics and drug discovery. After careful and rigid reviewing and selection process, 12 papers were eventually selected. It is a very successful collection of contributions. At the time of writing this editorial, i.e., 12 months after the first accepted paper and 2 months after the last accepted paper was published, the research topic has received 25 citations, nearly 5000 downloads, and 27,000 views.

REFERENCES

- Hong, Z., Zeng, X., Wei, L., and Liu, X. (2020). Identifying Enhancer-Promoter Interactions with Neural Network Based on Pre-trained DNA Vectors and Attention Mechanism. *Bioinformatics* 36 (4), 1037–1043. doi:10.1093/bioinformatics/btz694
- Jin, S., Zeng, X., Xia, F., Huang, W., and Liu, X. (2021). Application of Deep Learning Methods in Biological Networks. *Brief. Bioinformatics* 22 (2), 1902–1917. doi:10.1093/bib/bbaa043
- Kumar Shukla, P., Kumar Shukla, P., Sharma, P., Rawat, P., Samar, J., Moriwai, R., et al. (2020). Efficient Prediction of Drug-Drug Interaction Using Deep Learning Models. *IET Syst. Biol.* 14 (4), 211–216. doi:10.1049/iet-syb.2019.0116
- Li, X., Wang, X., Song, T., Lu, W., Chen, Z., and Shi, X. (2015). A Novel Computational Method to Reduce Leaky Reaction in DNA Strand Displacement. *J. Anal. Methods Chem.* 2015, 675827. doi:10.1155/2015/675827
- Liu, S., Liu, C., and Deng, L. (2018). Machine Learning Approaches for Protein-Protein Interaction Hot Spot Prediction: Progress and Comparative Assessment. *Molecules* 23, 2535. doi:10.3390/molecules23102535
- Öztürk, H., Özgür, A., and Ozkirimli, E. (2018). DeepDTA: Deep Drug-Target Binding Affinity Prediction. *Bioinformatics* 34 (17), i821–i829. doi:10.1093/bioinformatics/bty593
- Pushpakom, S., Iorio, F., Eyers, P. A., Escott, K. J., Hopper, S., Wells, A., et al. (2019). Drug Repurposing: Progress, Challenges and Recommendations. *Nat. Rev. Drug Discov.* 18 (1), 41–58. doi:10.1038/nrd.2018.168
- Shi, X., Wu, X., Song, T., and Li, X. (2016). Construction of DNA Nanotubes with Controllable Diameters and Patterns Using Hierarchical DNA Sub-tiles. *Nanoscale* 8, 3114785–3114792. doi:10.1039/c6nr02695h
- Song, T., Wang, G., Ding, M., Rodriguez-Paton, A., Wang, X., and Wang, S. (2021). “Network-Based Approaches for Drug Repositioning,” in *Molecular Informatics*. Hoboken, New Jersey: Wiley. 2100200. doi:10.1002/minf.202100200

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct, and intellectual contribution to the work and approved it for publication.

ACKNOWLEDGMENTS

We sincerely appreciate *Frontiers in Genetics* giving us this opportunity to organize this research topic. Special gratitude and appreciation are extended to all the contributors for their high-quality submissions and the reviewers for volunteering their time and expertise to review the scientific merit of the submitted manuscripts, without which we could not have made this Research Topic so successful.

- Yin, Q., Zheng, Y., Wang, B., and Zhang, Q. (2021). “Design of Constraint Coding Sets for Archive DNA Storage,” in *IEEE/ACM Transactions on Computational Biology and Bioinformatics*. Hoboken, New Jersey: Wiley. doi:10.1109/tcbb.2021.3127271
- Yu, X., Jiang, L., Jin, S., Zeng, X., and Liu, X. (2022). preMLI: a Pre-trained Method to Uncover microRNA-lncRNA Potential Interactions. *Brief Bioinform* 23, bbab470–1. doi:10.1093/bib/bbab470
- Zhang, Z., Li, J., Pan, L., Ye, Y., Zeng, X., Song, T., et al. (2014). A Novel Visualization of DNA Sequences, Reflecting GC-Content. *Match* 722, 533–550.
- Zou, Q., Wan, S., and Zeng, X. (2016). “HPTree: Reconstructing Phylogenetic Trees for Ultra-large Unaligned DNA Sequences via NJ Model and Hadoop,” in *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (Shenzhen: IEEE), 53–58. doi:10.1109/bibm.2016.7822492

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher’s Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2022 Zheng, Wang, Wang and Zeng. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.



Repositioning Drugs to the Mitochondrial Fusion Protein 2 by Three-Tunnel Deep Neural Network for Alzheimer's Disease

Xun Wang^{1*}, Yue Zhong¹ and Mao Ding^{2*}

¹ College of Computer Science and Technology, China University of Petroleum, Shandong, China, ² Department of Neurology Medicine, The Second Hospital, Cheeloo College of Medicine, Shandong University, Jinan, China

OPEN ACCESS

Edited by:

Quan Zou,
University of Electronic Science and
Technology of China, China

Reviewed by:

Pan Zheng,
University of Canterbury, New Zealand
Tao Song,
Polytechnic University of Madrid,
Spain

*Correspondence:

Xun Wang
wangsyun@upc.edu.cn
Mao Ding
18264181312@163.com

Specialty section:

This article was submitted to
Computational Genomics,
a section of the journal
Frontiers in Genetics

Received: 06 December 2020

Accepted: 08 January 2021

Published: 15 February 2021

Citation:

Wang X, Zhong Y and Ding M (2021)
Repositioning Drugs to the
Mitochondrial Fusion Protein 2 by
Three-Tunnel Deep Neural Network for
Alzheimer's Disease.
Front. Genet. 12:638330.
doi: 10.3389/fgene.2021.638330

Alzheimer's disease (AD) is a common neurodegenerative dementia in the elderly. Although there is no effective drug to treat AD, proteins associated with AD have been discovered in related studies. One of the proteins is mitochondrial fusion protein 2 (Mfn2), and its regulation presumably be related to AD. However, there is no specific drug for Mfn2 regulation. In this study, a three-tunnel deep neural network (3-Tunnel DNN) model is constructed and trained on the extended Davis dataset. In the prediction of drug-target binding affinity values, the accuracy of the model is up to 88.82% and the loss value is 0.172. By ranking the binding affinity values of 1,063 approved drugs and small molecular compounds in the DrugBank database, the top 15 drug molecules are recommended by the 3-Tunnel DNN model. After removing molecular weight <200 and topical drugs, a total of 11 drug molecules are selected for literature mining. The results show that six drugs have effect on AD, which are reported in references. Meanwhile, molecular docking experiments are implemented on the 11 drugs. The results show that all of the 11 drug molecules could dock with Mfn2 successfully, and 5 of them have great binding effect.

Keywords: Alzheimer's disease, drug repositioning, prediction of binding affinity values, three-tunnel deep neural network, molecular docking

1. INTRODUCTION

Alzheimer's disease (AD) is a destructive nervous system disease, which is characterized by a progressive dementia. The incidence of AD accounts for 50–70% of the total number of senile dementias. It mostly occurs in middle or late life, and the psychological skills, cognitive function, and physiological function of the patients have gradually lost (McKhann et al., 1984; Navarro et al., 2020). With the increasing aging of the population, AD has become an important world problem to be solved. However, the pathogenesis of AD is still unclear. The cascade hypothesis of amyloid β protein ($A\beta$) is the most concerned. The hypothesis holds that the formation of senile plaques by a large amount of $A\beta$ in the brain is related to cognitive dysfunction and pathological changes of AD (Lin Zhang et al., 2020). Abnormal deposition of $A\beta$ is considered to be the vital pathogenesis of AD. And $A\beta$ in cerebrospinal fluid has been included as a diagnostic marker of AD (Jia and Wei, 2018; Cui et al., 2020). A variety of AD-targeted drugs are difficult to be used in clinical practice because of poor efficacy or side effects in phase III clinical trials. More scholars focus on controlling the progression of mild cognitive impairment. Consequently, the regulation mechanism of $A\beta$ production and clearance has become an important research direction (Cui et al., 2020).

Mitochondria is the main site of cellular aerobic respiration. And mitochondrial dysfunction has effect on the production and toxicity of $A\beta$. Mitochondrial dynamics and mitochondrial dysfunction caused by abnormal mitochondrial autophagy play an important role in the pathogenesis of AD. The mitochondrial fusion protein 2 (Mfn2) is a dynamic protein expressed in the outer membrane of mitochondria. Mfn2 not only participates in mitochondrial fusion but also affects cell metabolism by regulating cell apoptosis, mitochondrial autophagy, and other biological processes. At present, Mfn2 has been proved to be closely related to the occurrence of many kinds of common diseases. Although some specific mechanisms are still unclear, Mfn2 is expected to become a new therapeutic target for some diseases (Li et al., 2020). In addition, Mfn2 involved in the regulation of protein homeostasis and pathogenesis of AD has become a research hotspot.

The approved drugs are designed and developed based on the concept of single target. Therefore, no drug has been specifically developed for Mfn2 regulation till now. It takes 10–15 years to implement the *de novo* drug design. In order to reduce the cost of drug development and the risk in the process of drug research, it has become an important strategy to repurpose the approved drugs and explore their new functions. And deep learning methods provide powerful technical supports in computing the drug-target interactions (DTIs). The prediction of DTIs is the focus of drug design and the key step of drug repositioning. However, it is obviously not accurate to divide the drug–target pairs (DT pairs) into effective and ineffective in the classification method. Therefore, more attention has been paid to the regression method, which directly predicts the binding affinity values of drug–target (DT) pairs with dissociation constant (K_d).

The DeepDTA model (Ozturk et al., 2018) considers the sequence information of drug molecules and proteins in the prediction of binding affinity values. Convolutional neural network is used in the research. It is considered to be the state-of-the-art model of predicting DTA (Huang et al., 2020). However, the model fluctuates greatly when training for many times. The GraphDTA model (Nguyen et al., 2020) uses graph convolution neural network to represent the features of drug molecules. Although its loss value is tiny, the calculation cost is too high. Recurrent neural networks (RNNs) such as gated recurrent units (GRU) (Cho et al., 2014) and long short-term memory units (LSTM) (Hochreiter and Schmidhuber, 1997) are widely used to capture temporal dependence in sequence-based data such as time series and text (Chuang et al., 2020). Extending on the use of a single RNN, the ensemble of RNNs with CNNs is a common hybrid architecture in recent applications that seeks to combine the ability of RNN in analyzing sequential data and CNN on extracting local features (Cao et al., 2020). Nonetheless, in the representation of drug molecules, the results are not better than that of CNN. The latest DeepGS model (Lin et al., 2020) inputs the sequence information and two-dimensional structure information of drug molecules as well as the protein sequence information into the model for prediction. It also has the problem of higher calculation cost. Moreover, information redundancy is inevitable as drug molecules are encoded twice by different

encoding strategies. In addition, DeepPurpose (Huang et al., 2020) provides a toolkit that integrates a variety of encoding methods of drug molecules and protein amino acid sequences. Two kinds of encoding methods are selected to input the model to predict the binding affinity values of DT pairs. The toolkit provides great convenience for future research.

In this study, we implement an approach that considers the binding affinity information and negative samples of DT pairs to reposition regulatory drugs Mfn2 as candidate medications of AD. First, a three-tunnel deep neural network (3-Tunnel-DNN) model is constructed and trained on the expanded Davis dataset using drug–protein binding affinity information. The three tunnels are protein sequences, drug molecules of positive samples, and negative samples. The accuracy of the 3-Tunnel-DNN model is 0.8882 and the loss value is 0.172 in the test set. Finally, the well-trained model is used to reposition 1,063 drugs/compounds from the DrugBank database to Mfn2 regulatory. A total of 15 drugs are recommended for Mfn2 regulation by ranking the binding affinity values of drugs/compounds from the database with Mfn2. After removing three molecules with molecular weight <200 and a topical drug, a total of 11 drug molecules are selected for literature mining and molecular docking experiments.

2. MATERIALS AND METHODS

2.1. The Extended Davis Dataset

Davis dataset contains the selective analysis of kinase protein family and related inhibitors and their respective K_d values, and it includes 30,056 binding affinity values of 442 proteins and 68 compounds (Ozturk et al., 2018; Davis et al., 2020). Negative samples are expected to be considered in our model. Davis dataset is widely used as training set in the field of drug-target binding affinity prediction, such as DeepDTA (Ozturk et al., 2018), DeepGS (Lin et al., 2020), GraphDTA (Nguyen et al., 2020), etc. Therefore, binding affinity values of DT pairs in Davis dataset are applied as training set in the 3-Tunnel DNN model as well. Besides, information of negative samples is added into Davis dataset to extend the dataset.

In the original Davis dataset, binding affinity data of DT pairs are measured by K_d values. It ranges from 0 to 10,000. The extended Davis dataset consists of four files, which are SMILES sequences file of compounds, FASTA sequences file of proteins, binding affinity values file of DT pairs, and SMILES sequences file of negative samples. The original Davis dataset consists of the first three files. The first and second files contain the sequence information needed in the model training process. The third file, in particular, is a 68×442 dimensional digital matrix [i.e., $M_{(68 \times 442)}$], in which each number $[m_{(i,j)}]$ represents the K_d value of the i -th compound and the j -th protein. The fourth file is a matrix [i.e., $NM_{(68 \times 442)}$] composed of SMILES sequences of negative samples. Each element $[nm_{(i,j)}]$ represents the SMILES sequence of the negative sample of the i -th compound and the j -th protein. In the research, the K_d value of 50 is taken as the boundary between positive and negative samples. It means that for each protein, the compounds with binding affinity value ≤ 50 are positive samples, and the compounds with > 50

are negative samples. The extended Davis dataset is given in **Supplementary Tables 1–4**.

In fact, we compare the recommendation of Zeng et al. that the boundary of positive and negative samples is set as 10 (Zeng et al., 2019) with our boundary of 50 as well. The results show that the K_d value of 50 as the boundary performs better. The results are shown in **Table 1**.

In the training process, the K_d values converted into log space (pK_d) are used as the actual binding affinity values for easier calculation of regression. The explanation is similar to Equation (1) (Huang et al., 2020).

$$pK_d = -\log_{10}(K_d \times 10^{-9} + 10^{-10}) \quad (1)$$

The DrugBank database contains common compounds (amino acids, polypeptides, choline, etc.), approved drugs (azithromycin, etc.), and approved small molecular compounds (5-fluorouridine, etc.). The first 1,063 drugs/compounds in the DrugBank database are used here as potential candidates for repositioning regulatory Mfn2. Particularly, SMILES sequences of drugs are used for calculation. Mfn2 sequence of human protein (Mfn2_Human) is used as the target protein for

repositioning. Mfn2_Human in form of FASTA sequence from the UniProt database is used for binding affinity calculations. The usage of data is shown in **Figure 1**.

2.2. Feature Extraction of Drug Molecules and Proteins

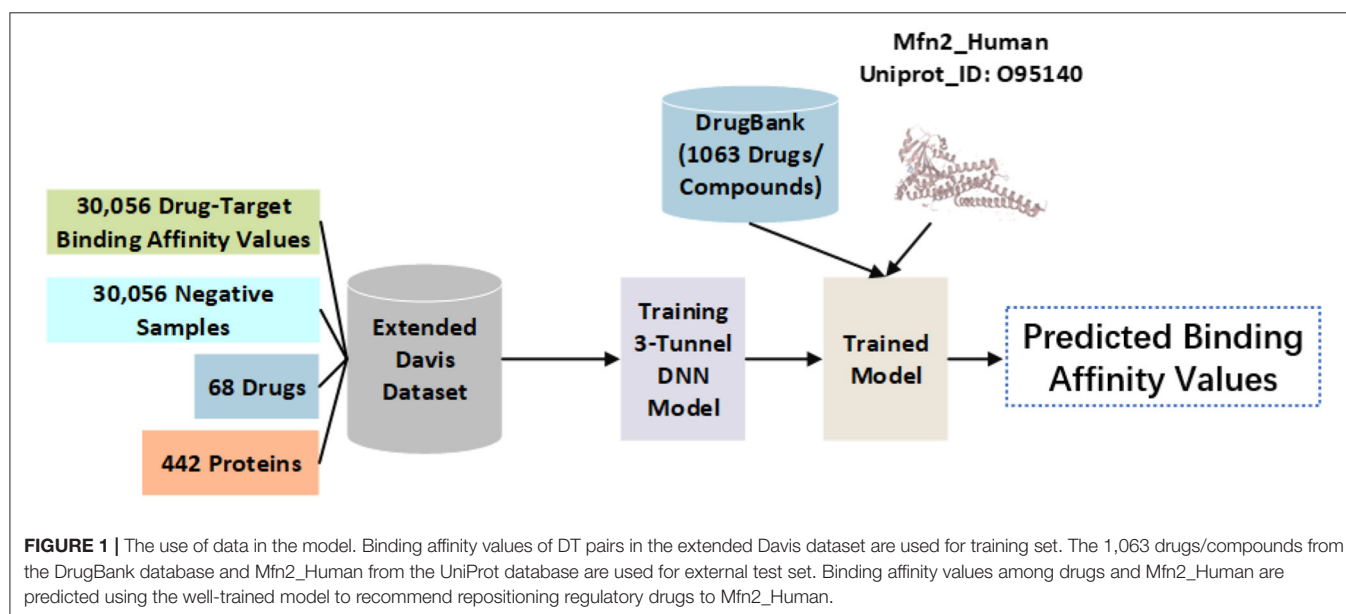
Extended-connectivity fingerprints (ECFPs) are a novel class of topological fingerprints for molecular characterization, which is a 1,024-length bits vector (Rogers and Hahn, 2010). In the study, $n=2$ (i.e., ECFP_2) is chosen as the circular radius that encodes the substructure of drug molecules. RDKit (Bento et al., 2020) is used to generate fingerprints of molecules. A multi-layer perceptron (MLP) (Chuang et al., 2020) is then applied on the binary fingerprint vector (Huang et al., 2020). In the 3-Tunnel DNN model, MLP is constructed as a four-layer neural network that the number of neurons is 1,024, 256, 64, and 256, respectively, to extract feature representations of drug molecules.

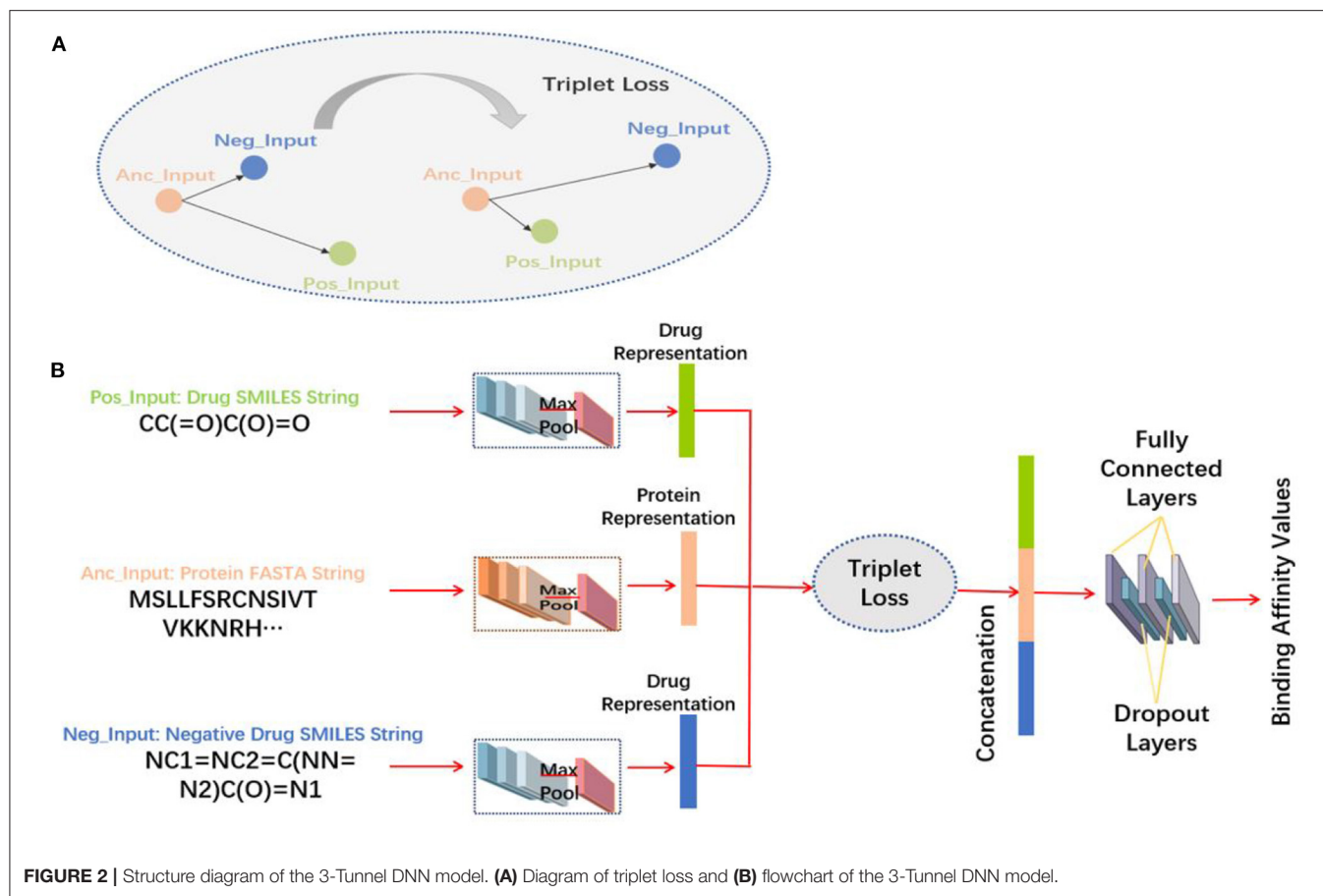
For proteins, there are 25 unique characters in protein FASTA sequence in Davis dataset (Ozturk et al., 2018). In our model, the symbol “?” is filled in the beginning of each sequence (Huang et al., 2020). Therefore, there are 26 unique characters in FASTA sequences. Each character is mapped into a unique integer, and the FASTA sequences are transformed into one-dimensional vectors. After that, the vector is extended into square data structure, in form of binary matrix with one-hot encoding strategy. The maximum length of FASTA sequences is set as 1,000 (Ozturk et al., 2018), so the matrix size of FASTA sequences is “1,000 × 26.” In particular, if the length of FASTA sequence is <1,000, the matrix is filled with 0. The matrix is input into the convolutional neural network (CNN), which consists of three layers of one-dimensional convolutional network and a global maximum pooling layer. The convolutional kernel is 32 × 1, 32 × 2, and 32 × 3, respectively (Ozturk et al., 2018; Huang et al., 2020). The activation function is Rectified Linear Unit (ReLU)

TABLE 1 | Comparison list of consistency index (CI) and mean square error (MSE) of different boundaries on test set.

Model	Mse	CI
CNN_CNN (boundary is set as 10)	0.878	0.261
CNN_CNN (boundary is set as 50)	0.881	0.245
3-Tunnel DNN (boundary is set as 10)	0.882	0.252
3-Tunnel DNN (boundary is set as 50)	0.888	0.172

Italic text indicates a better boundary between positive and negative samples. Bold text represents the best performance.





(Nair and Hinton, 2010). And then, the representation vector of the features is generated.

The feature extraction methods are shown in **Figure 2B**.

2.3. 3-Tunnel DNN Model

Information of negative samples is expected to be considered in the process of model training. Therefore, on the basis of two tunnels of drug molecules and proteins amino acid sequences, the third tunnel is added to process the data of negative samples to accurately reposition the drugs. The amino acid sequence of the protein is set as the anchor input (Anc_Input), the positive samples of the extended Davis dataset are taken as positive input (Pos_Input), and negative samples are set as negative input (Neg_Input). In the process of learning feature representations, triplet loss (Davis et al., 2020) is used to minimize the distance between Anc_Input and Pos_Input, and maximize the distance between Anc_Input and Neg_Input. The triplet loss is explained in Equation (2).

$$L = \max(\|f(X_i^{Anc_input}) - f(X_i^{Pos_input})\|_2^2 - \|f(X_i^{Anc_input}) - f(X_i^{Neg_input})\|_2^2 + M, 0) \quad (2)$$

where $f(X_i^{Anc_input})$ represents the feature representation of the i th protein amino acid sequence, $f(X_i^{Pos_input})$ represents

the feature representation of the i th positive sample, while $f(X_i^{Neg_input})$ represents the feature representation of the i th negative sample. In addition, $\|f(X_i^{Anc_input}) - f(X_i^{Pos_input})\|_2^2$ means the square of the Euclidean distance between the vectors of the i th Anc_Input and Pos_Input. Similarly, $\|f(X_i^{Anc_input}) - f(X_i^{Neg_input})\|_2^2$ means the square of the Euclidean distance between the vectors of the i th Anc_Input and Neg_Input. And M is a hyperparameter, which is set to 1 in the manuscript.

The three tunnels are used to process the FASTA sequences of proteins, the SMILES sequences of positive samples, and negative samples of drug molecules, respectively. The triplet loss (Schroff et al., 2015) is used here to obtain more accurate feature representations by maximizing the distance between proteins and negative samples and minimizing the distance between proteins and positive samples (**Figure 2A**). And then, three feature representations are concatenated together and input into the fully connected layers to make nonlinear changes to these extracted feature representations. In particular, the first two fully connected layers are followed by a dropout layer, respectively, which randomly “delete” hidden neurons to prevent over fitting, and finally map to the output space. The output of the model is the predicted binding affinity values of DT pairs. The 3-Tunnel DNN model is based on the MLP_CNN model (MLP for drugs encoding, CNN for proteins encoding) in DeepPurpose

toolkit (Lin et al., 2020), and its topologic structure is shown in **Figure 2B**.

The 30,056 DT pairs from the extended Davis dataset are taken as training set, which are divided into three subsets in the ratio of 7:1:2 (Huang et al., 2020). It means that 70% of the data are used for training, 10% for validation, and 20% for testing. We use 256 small batch data to update the weights of neural networks. The number of epochs of the 3-Tunnel DNN model is 100, as well Adam optimization algorithm (with learning rate of 10^{-4}) is applied to optimize the model.

2.4. Drug Reposition of Mfn2 by Well-Trained 3-Tunnel DNN Model

After the well-trained 3-Tunnel DNN model is saved, 1,063 SMILES sequences of drugs/compounds from the DrugBank database and Mfn2_human protein sequence in the form of FASTA sequence from the UniProt database are input into the well-trained model. These predicted values are ranked to get the drug recommendation list. After removing the molecules with molecular weight <200 and topical drugs, a total of 11 drug molecules are recommended to regulate Mfn2. Literature mining and molecular docking experiments are implemented to verify the effectiveness of these molecules.

2.5. Molecular Docking

A total of 11 structures of recommended drug molecules and Mfn2 are analyzed by molecular docking experiments. The X-ray crystal structure of Mfn2 (PDB code: 6JFK, Resolution: 2.00 Å) is downloaded from RCSB Protein Data Bank (<http://www.rcsb.org>) in PDB format, and the first conformation is chosen as the receptor structure. The three-dimensional structures of recommended molecules are downloaded from the DrugBank database (<https://www.drugbank.ca>) in PDB format as well. UCSF Chimera software (Pettersen et al., 2004) is used to prepare receptor protein binding sites, establish the three-dimensional structure of molecules, and minimize the energy.

Before the formal docking experiments, it is necessary to prepare the documents of receptor protein, binding sites, protein surface, and drug molecules. For the receptor protein, all structures of ligands and hydrogens are deleted first. Dock Prep module is used to supplement the parameters of the receptor protein. Hydrogens, AMBER ff14SB force field, and AM1-BCC charges (Jakalian et al., 2000, 2010) of receptor and ligand are added, respectively. After that, the results are saved to a file in mol2 format. Then all hydrogens are deleted again and saved in PDB format. For the binding sites in the receptor, the same operation is implemented and the result is saved in the format of mol2. The DMS tool in UCSF Chimera (Pettersen et al., 2004) is used to generate the surface of the receptor using a probe atom with a 1.4 Å radius, which is saved as the file in the format of dms. Similarly, for drug molecules, hydrogenation, and charging operations are performed, and the results are saved as a file in the format of mol2 as well.

For each binding site, the Sphgen module is used to generate a spherical collection around the active site. The grid file is generated by the Grid module, which is used for grid-based energy assessment. Then, the semi-flexible docking is

implemented with the program of Dock6.8 (Lang et al., 2009; Mukherjee et al., 2010), and 1,000 different orientations are generated. In particular, some drug molecules (e.g., adinazolam, pyrimethamine, and carbamazepine) are implemented rigid molecular docking to evaluate whether the receptor can accommodate the conformation. After that, the van der Waals force and electrostatic interaction between the ligand and the binding site are obtained, and the grid scores are also calculated. Finally, the best conformation is obtained by using cluster analysis (RMSD threshold is 2.0 Å) in semi-flexible docking, and in rigid docking, only one conformation is obtained.

3. RESULTS

3.1. Results of Model Training

In the training process, consistency index (CI) (Pahikkala et al., 2014) is used to evaluate the training performance, and mean square error (MSE) (Kansal et al., 2019) is used as the loss function to measure the error of each epoch.

We compare the performance of boundary with the K_d value of 10 (Zeng et al., 2019) and 50 as positive and negative samples. The performance of CNN_CNN model (CNN for drugs encoding, CNN for proteins encoding) and the 3-Tunnel DNN model at different boundaries are compared primarily. The results are shown in **Table 1** and **Figure 3**. The results show that the K_d value of 50 as the boundary of positive and negative samples makes the model perform better. And our 3-Tunnel DNN model performs better than CNN_CNN model.

The 3-Tunnel DNN model is compared with the performance with the state-of-the-art model at present, such as DeepDTA (Ozturk et al., 2018), GraphDTA (Nguyen et al., 2020), and the latest DeepGS (Lin et al., 2020) model. At the same time, the original CNN_CNN model (Huang et al., 2020) and other models obtained in the DeepPurpose toolkit using our extended Davis dataset, such as CNN_CNN, CNN+LSTM_CNN (CNN+LSTM for drugs encoding, CNN for proteins encoding), and CNN+GRU_CNN (CNN+GRU for drugs encoding, CNN for proteins encoding) model, are also compared with the 3-Tunnel DNN model. The performances of these models are shown in **Table 2**.

According to results of the 3-Tunnel DNN model, the value of CI on test set is 0.888 and that of MSE is 0.172, which performs best among these models. The CI value of the 3-Tunnel DNN model is improved by 0.6% compared with DeepGS model (Lin et al., 2020), that is, with best accuracy. And the MSE value is improved by 7.3% compared with GraphDTA model (Nguyen et al., 2020), that is, with minimum loss. In addition, we also find that the MSE value of models using the extended Davis dataset as the training set is significantly smaller than models trained on the original Davis dataset, except for CNN+GRU_CNN model.

3.2. Results of Recommended Repositioning Regulatory Drugs to Mfn2

For repositioning regulatory drugs to Mfn2, the well-trained 3-Tunnel DNN model is used to calculate the binding affinity value of each pair of potential drug-Mfn2 pairs. The SMILES of potential drugs are the 1,063 approved drugs in the DrugBank

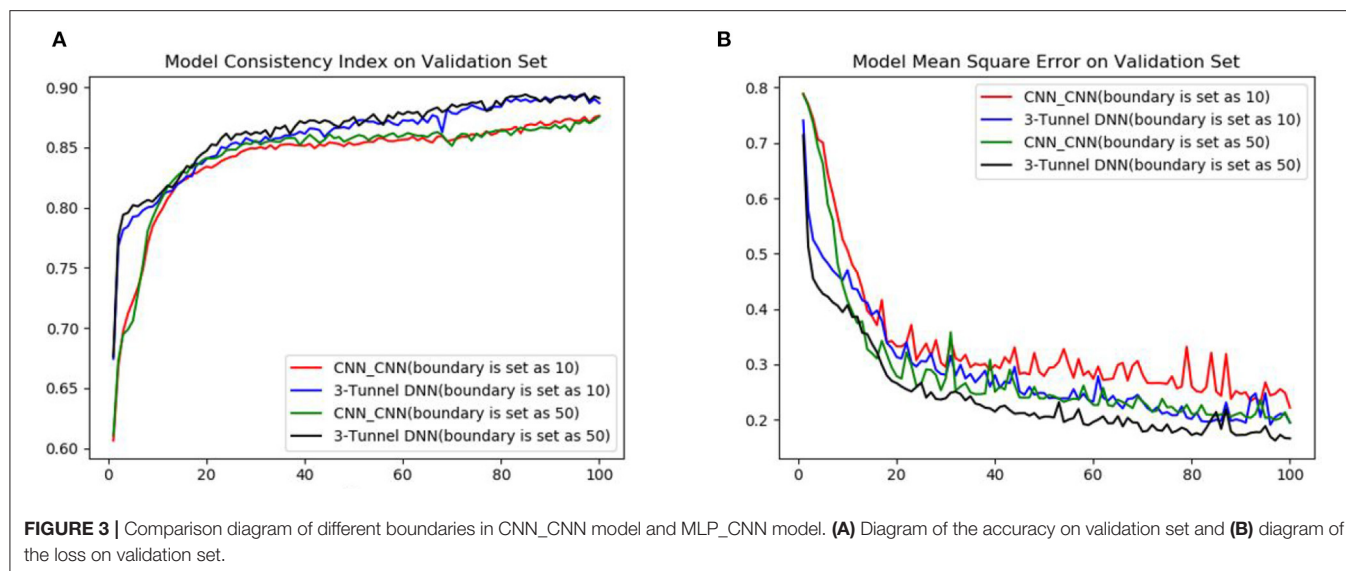


TABLE 2 | Comparison list of consistency index (CI) and mean square error (MSE) of different models on test set.

Model	Mse	CI
CNN_CNN (boundary is set as 10)	0.878	0.261
DeepDTA (Davis dataset) (Ozturk et al., 2018)	0.878	0.261
GraphDTA (Davis dataset) (Nguyen et al., 2020)	0.881	0.245
DeepGS (Davis dataset) (Lin et al., 2020)	0.882	0.252
CNN_CNN (Davis dataset) (Huang et al., 2020)	0.879	0.254
CNN_CNN (our extended Davis dataset)	0.87	0.209
CNN+LSTM_CNN (our extended Davis dataset)	0.86	0.234
CNN+GRU_CNN (our extended Davis dataset)	0.834	0.263
3-Tunnel DNN (our extended Davis dataset)	0.888	0.172

Bold text represents the best performance.

database, and Mfn2 is the amino acid sequence of human in the form of FASTA sequence from the UniProt database. According to the ranking of predicted values, the 11 drugs recommended by 3-Tunnel DNN model are shown in **Table 3** after removing drugs with molecular weight <200 (niacin, ethionamide, and acetohydroxamic acid) and an anesthetic (dyclonine). In particular, although dyclonine is reported to be effective for targets of AD in the reference (Zhang et al., 2019), it is still removed from the results because of topical drug.

We search the keywords “Alzheimer; Drug name” and find that 6 drugs have supported references. Names, DrugBank ID, original functions, and description of supported references of these drugs are shown in **Table 3**.

It is reported that neuroleptic medication appears to have modest efficacy in controlling behavioral symptoms in dementia patients (Lemke, 1995). And the three drugs (adinazolam, fluphenazine, and carbamazepine) are related to mental illness. In addition, anti-tumor drugs, anti-epilepsy drugs, anti-infection drugs, and drugs for the treatment of hypertension are included in the recommended list.

3.3. Results of Molecular Docking

Dock6.8 program (Lang et al., 2009; Mukherjee et al., 2010) is used to predict the binding patterns of 11 drug molecules in Mfn2. The value of Grid_Score is used to evaluate the molecular docking results, which represents the sum of van der Waals force and electrostatic interaction. The negative value of Grid_Score indicates that the drug molecule is bound to the target, while the positive value indicates no binding. And the smaller the Grid_Score, the stronger binding of drug molecules to Mfn2. Generally, the value of Grid_Score > -40 kcal/mol indicates poor binding, the value between -40 and -50 kcal/mol indicates medium binding, and the value < -50 kcal/mol indicates great binding (Liu et al., 2018). The Grid_Score values of each drug molecule binding to Mfn2 are shown in **Table 4**.

According to the results of molecular docking, the binding effect of bosentan and Mfn2 is the best, and it is supported by the reference (Elesber et al., 2006) as well. In addition, the Grid_Score of imatinib and pemetrexed are < -50 kcal/mol, and they also have strong binding with Mfn2. Pemetrexed is not supported by any reference, but its binding capacity to Mfn2 is slightly lower than bosentan. And sulfametopyrazine and fluphenazine have medium binding with Mfn2. However, lamotrigine, voriconazole, and nabumetone have poor binding with Mfn2 in the experiments of semi-flexible docking. Lamotrigine, in particular, has the lowest score among these drug molecules, although it is supported by reference (Tsolaki et al., 2000). In addition, the Grid_Scores of adinazolam, pyrimethamine, and carbamazepine are not satisfactory. We speculate that it is caused by the rigid molecular docking. Since atomic bonds cannot rotate, there is only one conformation in the rigid docking.

4. DISCUSSION

In this study, we construct a 3-Tunnel DNN model based on the original drug-target binding affinity prediction model to consider the influence of negative samples. The binding affinity

TABLE 3 | Recommended drugs by the three-tunnel deep neural network (3-Tunnel DNN) model.

Drug name	Drug ID	Disease treated with the drug	References & Descriptions
Imatinib	DB00619	Antineoplastic	It is confirmed that imatinib-mediated control of neprilysin could indeed be accounted for its effect on activation induced cell death (Bauer et al., 2011)
Lamotrigine	DB00555	Antiepileptic	
Sulfametopyrazine	DB00664	The treatment of respiratory, urinary tract infections, and malaria	The study shows that lamotrigine is an effective and safe monotherapy in patients with cognitive disorders and AD (Tsolaki et al., 2000)
Adinazolam	DB00546	Anxiolytic, anticonvulsant, sedative, and antidepressant	
Pyrimethamine	DB00205	Antimalarial or the treatment of toxoplasmosis	The study shows that bosentan can play a pathophysiological role in the endogenous endothelin system of AD (Elesber et al., 2006)
Bosentan	DB00559	The treatment of pulmonary hypertension	
Voriconazole	DB00582	Antifungal	Fluphenazine and other depot neuroleptics are used in AD patients to treat behavioral disorders (Gottlieb et al., 1988)
Fluphenazine	DB00623	The treatment of psychoses	
Pemetrexed	DB00642	Antineoplastic	It provides a method for treating and preventing dementia such as AD, which comprises administering an effective, nontoxic amount of nabumetone or 6MNA (Clark, 1997)
Nabumetone	DB00461	Anti-inflammatory	
Carbamazepine	DB00564	Anticonvulsant and analgesic	The result shows that carbamazepine may be useful to treat agitated AD patients (Gleason and Schneider, 1990)

values of DT pairs are trained on the extended Davis dataset (i.e., the positive and negative samples are divided by the K_d value of 50 on the original Davis dataset). Then, the binding affinity values of 1,063 drug molecules with Mfn2 protein are predicted using a well-trained deep learning model. Literature mining and molecular docking experiments are implemented on the recommended 11 molecules. The values of accuracy and loss of the model are obviously better than the existing models, especially the loss value is 0.172. Six of the 11 molecules have been reported by other researchers. The results of molecular docking show that all of the 11 drug molecules can dock with Mfn2 successfully. And five drug molecules have medium or strong binding force. In particular, bosentan has the best performance of molecular docking, which is also supported by the reference (Elesber et al., 2006). In addition, pemetrexed and imatinib are prospect drugs as well. Specially, pemetrexed has not been used in the treatment of AD, and its molecular docking result is just tiny lower than bosentan. In the following work, we will evaluate the pharmacology and toxicology of pemetrexed, and *in vitro* experiments could be prepared to verify its effectiveness.

Although we find that positive and negative samples in Davis dataset with the K_d value of 50 as the boundary is better

TABLE 4 | Results of molecular docking.

Drug name	Drug ID	Grid score	Is supported by references
Imatinib	DB00619	−51.678650	Yes
Lamotrigine	DB00555	−28.860310	Yes
Sulfametopyrazine	DB00664	−43.703491	No
<i>Adinazolam</i>	<i>DB00546</i>	−29.337458	<i>No</i>
<i>Pyrimethamin</i>	<i>DB00205</i>	−31.183176	<i>No</i>
Bosentan	DB00559	−55.177814	Yes
Voriconazole	DB00582	−39.047768	No
Fluphenazine	DB00623	−49.189960	Yes
Pemetrexed	DB00642	−54.729557	No
Nabumetone	DB00461	−38.957726	Yes
<i>Carbamazepine</i>	<i>DB00564</i>	−34.177979	Yes

Bold text shows the drugs with Grid_Score < −50 kcal/mol, and italic text represents rigid molecular docking.

than 10, its specific value is still worthy to study. In addition, other datasets (such as KIBA and BindingDB) are expected to be extended and implemented in the model in our future

work. And, it is a promising research to add gene information (Chen et al., 2013; Zhang et al., 2013; Shi et al., 2014; Tan et al., 2014) into the drug–target relationships and explore the relationships between genes (Li et al., 2015, 2016; Shi et al., 2015) and drugs. Furthermore, the training speed and accuracy of the feature representations of drug molecules extracted by molecular fingerprint is obviously better than that of SMILES sequences. For further research, a better feature extraction method for protein characteristics is expected to be obtained. And spiking neural P systems (Song et al., 2013, 2015a,b; Song and Pan, 2015) is also considered to be implemented in the future.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/**Supplementary Material**, further inquiries can be directed to the corresponding author/s.

AUTHOR CONTRIBUTIONS

XW conceived and designed the experiments. YZ performed the experiments and wrote the code. MD analyzed the

data. XW and YZ drafted the work and revised it. All authors contributed to the article and approved the submitted version.

FUNDING

This work was supported by National Natural Science Foundation of China (Grant Nos. 61873280, 61873281, 61672033, 61672248, 61972416), Taishan Scholarship (tsqn201812029), Natural Science Foundation of Shandong Province (No. 2019GGX101067, ZR2019MF012), Fundamental Research Funds for the Central Universities (18CX02152A, 19CX05003A-6), and Foundation of Science and Technology Development of Jinan (201907116).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/fgene.2021.638330/full#supplementary-material>

Supplementary Tables 1–4 | The extended Davis dataset.

REFERENCES

- Bauer, C., Pardossi-Piquard, R., Dunys, J., Roy, M., and Checler, F. (2011). γ -secretase-mediated regulation of neprilysin: influence of cell density and aging and modulation by imatinib. *J. Alzheimer's Dis.* 27, 511–520. doi: 10.3233/JAD-2011-110746
- Bento, A. P., Hersey, A., Felix, E., Landrum, G., Gaulton, A., Atkinson, F., et al. (2020). An open source chemical structure curation pipeline using RDKit. *J. Cheminform.* 12:51. doi: 10.1186/s13321-020-00456-1
- Cao, Y., Geddes, T. A., Yang, J. Y. H., and Yang, P. (2020). Ensemble deep learning in bioinformatics. *Nat. Mach. Intell.* 2, 1–9. doi: 10.1038/s42256-020-0217-y
- Chen, Z., Song, T., Huang, Y., and Shi, X. (2013). Solving vertex cover problem using DNA tile assembly model. *J. Appl. Math.* 407816, 2541–2565. doi: 10.1155/2013/407816
- Cho, K., van Merriënboer, B., Gulcehre, C., Fethi Bougares, D. B., Schwenk, H., and Bengio, Y. (2014). “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” in *Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)* (Doha). doi: 10.3115/v1/D14-1179
- Chuang, K. V., Gunsalus, L. M., and Keiser, M. J. (2020). Learning molecular representations for medicinal chemistry. *J. Med. Chem.* 63, 8705–8722. doi: 10.1021/acs.jmedchem.0c00385
- Clark, M. S. G. (1997). Use of nabumetone or 6-methoxynaphthyl acetic acid for the treatment of dementia. US5695774 A.
- Cui, Y., Wang, Y., and Wang, P. (2020). Regulation of amyloidogenesis and clearance of β -amyloid in Alzheimer's disease. *Chinese J. Biochem. Mol. Biol.* 1–7. doi: 10.13865/j.cnki.cjbmb.2020.11.1384
- Davis, M. I., Hunt, J. P., Herrgard, S., Ciceri, P., Wodicka, L. M., Pallares, G., et al. (2020). Comprehensive analysis of kinase inhibitor selectivity. *Nat. Biotechnol.* 29, 1046–1051. doi: 10.1038/nbt.1990
- Elesber, A. A., Bonetti, P. O., Julie E Woodrums, X. Z., Lerman, L. O., Pallares, G., et al. (2006). Bosentan preserves endothelial function in mice overexpressing app. *Neurobiol. Aging* 27, 446–450. doi: 10.1016/j.neurobiolaging.2005.02.012
- Gleason, R. P., and Schneider, L. S. (1990). Carbamazepine treatment of agitation in Alzheimer's outpatients refractory to neuroleptics. *J. Clin. Psychiatry* 51, 115–118.
- Gottlieb, G. L., McAllister, T. W., and Gur, R. C. (1988). Depot neuroleptics in the treatment of behavioral disorders in patients with Alzheimer's disease. *J. Am. Geriatr. Soc.* 36, 619–621. doi: 10.1111/j.1532-5415.1988.tb06157.x
- Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.* 9, 1735–1780. doi: 10.1162/neco.1997.9.8.1735
- Huang, K., Fu, T., Xiao, C., Glass, L. M., and Sun, J. (2020). Deepurpose: a deep learning library for drug-target interaction prediction and applications to repurposing and screening. *Bioinformatics* 1–3. doi: 10.1093/bioinformatics/btaa1005
- Jakalian, A., Bush, B. L., Jack, D. B., and Bayly, C. I. (2000). Fast, efficient generation of high-quality atomic charges. AM1-BCC model: I. Method. *J. Comput. Chem.* 21, 132–146. doi: 10.1002/(SICI)1096-987X(20000130)21:2<132::AID-JCC5>3.0.CO;2-P
- Jakalian, A., Jack, D. B., and Bayly, C. I. (2010). Fast, efficient generation of high-quality atomic charges. AM1-BCC model: II. Parameterization and validation. *J. Comput. Chem.* 23, 1623–1641. doi: 10.1002/jcc.10128
- Jia, J., and Wei, C. (2018). 2018 chinese guidelines for the diagnosis and treatment of dementia and cognitive impairment, guidelines for the diagnosis and treatment of Alzheimer's disease. *Chinese J. Med.* 13, 971–977. doi: 10.3760/cma.j.issn.0376-2491.2018.13.004
- Kansal, S., Bansod, P. P., and Kumar, A. (2019). Prediction of instantaneous heart rate using adaptive algorithms. *Int. J. Adapt. Innov. Syst.* 2, 267–281. doi: 10.1504/IJAIS.2019.108397
- Lang, P. T., Brozell, S. R., Mukherjee, S., Pettersen, E. F., Meng, E. C., Thomas, V., et al. (2009). Dock 6: combining techniques to model RNA-small molecule complexes. *RNA* 15, 1219–1230. doi: 10.1261/rna.1563609
- Lemke, M. R. (1995). Effect of carbamazepine on agitation in Alzheimer's inpatients refractory to neuroleptics. *J. Clin. Psychiatry* 56, 354–357.
- Li, H., Lei, L., and Han, S. (2020). Role of mitofusin 2 in major diseases. *Biol. Chem. Eng.* 6, 160–162.
- Li, X., Song, T., Chen, Z., Shi, X., Chen, C., and Zhang, Z. (2016). A universal fast colorimetric method for DNA signal detection with DNA strand displacement and gold nanoparticles. *J. Nanomater.* 16, 464–469. doi: 10.1155/2015/407184
- Li, X., Wang, X., Song, T., Lu, W., Chen, Z., and Shi, X. (2015). Novel computational method to reduce leaky reaction in DNA strand displacement. *J. Analyt. Methods Chem.* 2015:675827. doi: 10.1155/2015/675827
- Lin Zhang, D. F., Zhao, D., Wang, Y., and Liu, C. (2020). Study on uptake of Puerarin by SH-SY5Y cells and improvement of α 1-42 induced cell damage. *Chinese Arch. Tradit. Chinese Med.* 1–7. Available online at: <http://kns.cnki.net/kcms/detail/21.1546.R.20201106.1612.028.html>
- Lin, X., Zhao, K., Xiao, T., Quan, Z., Wang, Z., and Yu, P. S. (2020). “Deepgs: Deep representation learning of graphs and sequences for drug-target binding affinity

- prediction,” in *The 24th European Conference on Artificial Intelligence (ECAI 2020)*, (Santiago de Compostela).
- Liu, T., Liu, M., Chen, F., Chen, F., Tian, Y., Huang, Q. (2018). A small-molecule compound has anti-influenza A virus activity by acting as a 'pb2 inhibitor'. *Mol. Pharm.* 15, 4110–4120. doi: 10.1021/acs.molpharmaceut.8b00531
- McKhann, G., Drachman, D., Folstein, M., Katzman, R., Price, D., and Stadlan, E. M. (1984). Clinical diagnosis of Alzheimer's disease: Report of the NINCDS-ADRDA work group under the auspices of department of health and human services task force on Alzheimer's disease. *Nat. Neurol.* 34, 939–944. doi: 10.1212/WNL.34.7.939
- Mukherjee, S., Balus, T. E., and Rizzo, R. C. (2010). Docking validation resources: protein family and ligand flexibility experiments. *J. Chem. Inform. Model.* 50, 1986–2000. doi: 10.1021/ci1001982
- Nair, V., and Hinton, G. E. (2010). “Rectified linear units improve restricted boltzmann machines,” in *Proceedings of the 27th International Conference on International Conference on Machine Learning (Haifa)*, 807–814.
- Navarro, J. F., Croteau, D. L., Jurek, A., Andrusivova, Z., Yang, B., Wang, Y., et al. (2020). Associated with dysregulated mitochondrial functions and stress signaling in Alzheimer disease. *iScience* 23:101556. doi: 10.1016/j.isci.2020.101556
- Nguyen, T., Le, H., and Venkatesh, S. (2020). Graphdta: prediction of drug-target binding affinity using graph convolutional networks. *Bioinformatics* btaa921. doi: 10.1093/bioinformatics/btaa921
- Ozturk, H., Ozkirimli, E., and Ozgur, A. (2018). Deepdta: Deep drug-target binding affinity prediction. *Bioinformatics* 34, 821–829. doi: 10.1093/bioinformatics/bty593
- Pahikkala, T., Airola, A., Sami Pietilä, S. S., Shakyawar, S., Szwajda, A., Tang, J., et al. (2014). Toward more realistic drug-target interaction predictions. *Brief. Bioinformatics* 16, 325–337. doi: 10.1093/bib/bbu010
- Pettersen, E. F., Goddard, T. D., Huang, C. C., Couch, G. S., Greenblatt, D. M., Meng, E. C., et al. (2004). UCSF chimera-a visualization system for exploratory research and analysis. *J. Comput. Chem.* 25, 1605–1612. doi: 10.1002/jcc.20084
- Rogers, D., and Hahn, M. (2010). Extended-connectivity fingerprints. *J. Chem. Inform. Model.* 50, 742–754. doi: 10.1021/ci100050t
- Schroff, F., Kalenichenko, D., and Philbin, J. (2015). “Facenet: A unified embedding for face recognition and clustering,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Boston, MA), 815–823. doi: 10.1109/CVPR.2015.7298682
- Shi, X., Chen, C., Li, X., Song, T., Chen, Z., Zhang, Z., et al. (2015). Size controllable DNA nanoribbons assembled from three types of reusable brick single-strand DNA tiles. *Soft Matter* 11, 8484–8492. doi: 10.1039/C5SM00796H
- Shi, X., Wang, Z., Deng, C., Song, T., Pan, L., and Chen, Z. (2014). A novel bio-sensor based on DNA strand displacement. *PLoS ONE* 9:e108856. doi: 10.1371/journal.pone.0108856
- Song, T., and Pan, L. (2015). Spiking neural p systems with rules on synapses working in maximum spikes consumption strategy. *IEEE Trans. Nanobiosci.* 14, 38–44. doi: 10.1109/TNB.2014.2367506
- Song, T., Pan, L., and Paun, G. (2013). Asynchronous spiking neural P systems with local synchronization. *Inform. Sci.* 219, 197–207. doi: 10.1016/j.ins.2012.07.023
- Song, T., Xu, J., and Pan, L. (2015a). On the universality and non-universality of spiking neural P systems with rules on synapses. *IEEE Trans. Nanobiosci.* 14, 960–966. doi: 10.1109/TNB.2015.2503603
- Song, T., Zou, Q., Zeng, X., and Liu, X. (2015b). Asynchronous spiking neural p systems with rules on synapses. *Neurocomputing* 151, 1439–1445. doi: 10.1016/j.neucom.2014.10.044
- Tan, G., Song, T., and Chen, Z. (2014). Spiking neural p systems with anti-spikes and without annihilating priority as number acceptors. *J. Syst. Eng. Electron.* 25, 464–469. doi: 10.1109/JSEE.2014.00053
- Tsolaki, M., Kourtis, A., Divanoglou, D., Bostanzopoulou, M., and Kazis, A. (2000). Monotherapy with lamotrigine in patients with Alzheimer's disease and seizures. *Am. J. Alzheimer's Dis. Other Dement.* 15, 74–79. doi: 10.1177/153331750001500209
- Zeng, X., Zhu, S., Liu, X., Zhou, Y., and Cheng, F. (2019). deepdr: A network-based deep learning approach to *in silico* drug repositioning. *Bioinformatics* 35, 5191–5198. doi: 10.1093/bioinformatics/btz418
- Zhang, B., Pang, X., Jia, H., Wang, Z., Liu, A., and Du, G. (2019). Repositioning drug discovery for Alzheimer's disease based on global marketed drug data. *Acta Pharm. Sin.* 54, 1214–1224. doi: 10.16438/j.0513-4870.2019-0165
- Zhang, K., Huang, X., Shi, X., Qiang, X., Song, T., Xinzhu, S., et al. (2013). A dynamic programming algorithm for circular single-stranded DNA tiles secondary structure prediction. *Appl. Math. Inform. Sci.* 15, 2533–2538. doi: 10.12785/amis/070649

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2021 Wang, Zhong and Ding. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.



Medical Image Protection Algorithm Based on Deoxyribonucleic Acid Chain of Dynamic Length

Xianglian Xue^{1,2}, Haiyan Jin^{1,3}, Dongsheng Zhou⁴ and Changjun Zhou^{5*}

¹ School of Computer Science and Engineering, Xi'an University of Technology, Xi'an, China, ² Sections of Computer Teaching and Research, Shaanxi University of Chinese Medicine, Xianyang, China, ³ Shaanxi Key Laboratory for Network Computing and Security Technology, Xi'an University of Technology, Xi'an, China, ⁴ Key Laboratory of Advanced Design and Intelligent Computing, Ministry of Education, Dalian University, Dalian, China, ⁵ College of Mathematics and Computer Science, Zhejiang Normal University, Jinhua, China

OPEN ACCESS

Edited by:

Pan Zheng,
University of Canterbury, New Zealand

Reviewed by:

Enqiang Zhu,
Guangzhou University, China
Xuncai Zhang,
Zhengzhou University of Light
Industry, China

*Correspondence:

Changjun Zhou
zhou-chang231@163.com

Specialty section:

This article was submitted to
Computational Genomics,
a section of the journal
Frontiers in Genetics

Received: 17 January 2021

Accepted: 09 February 2021

Published: 04 March 2021

Citation:

Xue XL, Jin HY, Zhou DS and
Zhou CJ (2021) Medical Image
Protection Algorithm Based on
Deoxyribonucleic Acid Chain
of Dynamic Length.
Front. Genet. 12:654663.
doi: 10.3389/fgene.2021.654663

Current image encryption algorithms have various deficiencies in effectively protecting medical images with large storage capacity and high pixel correlation. This article proposed a new image protection algorithm based on the deoxyribonucleic acid chain of dynamic length, which achieved image encryption by DNA dynamic coding, generation of DNA dynamic chain, and dynamic operation of row chain and column chain. First, the original image is encoded dynamically according to the binary bit from a pixel, and the DNA sequence matrix is scrambled. Second, DNA sequence matrices are dynamically segmented into DNA chains of different lengths. After that, row and column deletion operation and transposition operation of DNA dynamic chain are carried out, respectively, which made DNA chain matrix double shuffle. Finally, the encrypted image is got after recombining DNA chains of different lengths. The proposed algorithm was tested on a list of medical images. Results showed that the proposed algorithm showed excellent security performance, and it is immune to noise attack, occlusion attack, and all common cryptographic attacks.

Keywords: FOCHC system, DNA dynamic encoding, DNA dynamic chain, medical image encryption, deletion and transposition operation

INTRODUCTION

Nowadays, technologies such as telemedicine, tele-surgery, and tele-radiology have been enormously developed and are in the preparation stage for clinical usage (Priyanka and Maheshkar, 2017). Patient information may be exposed to network transmission with these technologies. Especially, medical images (MRI, CT, and X-ray) with large data storage, redundancy and high pixel correlation are easily attacked and tampered by unauthorized access. Therefore, it is necessary to develop efficient high-performance medical image encryption method.

Since the ground-breaking work on DNA computing conducted and reported by Adleman (1994). DNA computing has attracted ever increasing attention of researchers worldwide, due to its superior characteristics of large concurrency, mass storage and low energy consumption (Li et al., 2020; Liu et al., 2020; Wang B. et al., 2020; Zhu et al., 2020). In 2009, DNA coding theory was used in the field of image information security by Zhang et al. (Xue et al., 2010a,b; Zhang et al., 2010; Liu et al., 2012; Zhang and Wei, 2013), which opened a new window for the DNA cryptography.

The main encryption ideas were using the DNA operations (addition, subtraction, XOR, and DNA complement operations) and combination with some chaotic systems to achieve image encryption. Their novel methods and better encryption effects were often emulated and affirmed by researchers. However, Zhang et al.'s method was criticized as being unsafe in recent years. For instance, Zhu et al. (2017) and Hermassi et al. (2014) pointed out that the DNA addition operation proposed by Zhang et al. (2010) was irreversible. Besides, the encryption algorithm proposed by Zhang et al. (2012) has been deciphered by Belazi et al. (2014), Liu et al. (2014), and Wang et al. (2015) with chosen plaintext attack (CPA), respectively.

To improve the security, some researchers combined the complex chaotic systems with the DNA coding. For instance, Mondal and Mandal (2017) used two Logistic chaotic systems, and Zhang Y.Q. et al. (2016) used MLNCML system embedded logistic, to strengthen the existing algorithm. All of them combined the chaotic system with the DNA coding operations (addition and subtraction) to encrypt images. Zhang and Gao (2016) proposed an image encryption method which used hyper-chaotic system to control the DNA complement operation. However, because they adopted the technique of fixed DNA coding and fixed operation rules, the security of their algorithm was quite fragile. Further, the encryption key had not been associated with the original image. As a result, although complex chaotic systems was used to improve the security of the algorithm, the encrypted images could still be easily deciphered by the CPA and brute force attack (BFA) or the known plaintext attack (KPA) (Dou et al., 2017). Note that chaotic systems play a major role in such encryption methods, while the DNA coding operation without chaotic mapping was equivalent to the calculation of binary bits, and its security was not guaranteed. For example, Kumar et al. (2016) proposed a technique using DNA coding combined with elliptic curve Diffie-Hellman for image encryption, while it was deciphered by Akhavan et al. (2017) at no much cost using the chosen plaintext attack.

For these reasons, researchers used more efficient DNA coding mechanisms and DNA operations to achieve better performance chaotic systems for image encryption. In terms of DNA coding, Kalpana and Murali (2015), Zhen et al. (2016), Chai et al. (2017), Rehman et al. (2018), Dagadu et al. (2019a), and Hossein et al. (2020), etc. proposed different DNA dynamic coding, respectively. These methods gave DNA bases higher levels of encryption. However, all of the above dynamic DNA coding were based on image blocks or based on pixel-by-pixel. In addition, some of them (Dagadu et al., 2019a; Hossein et al., 2020) could not resist CPA. In terms of the DNA operations, dynamic addition operation (Zhang J. et al., 2016) and complement operation (Belazi et al., 2019), and cellular automata operation (Zhou et al., 2016; Chai et al., 2017) were proposed. Their encryption effects were better but the algorithms were more complex. In terms of the chaotic system, because the hyper-chaotic system obtained by fractional order calculation had low correlation and more complex dynamic characteristics (Zhu et al., 2014), it was favored by researchers. For example, Zhang L.M.

et al. (2017) used the fractional-order hyper-chaotic system (FOHC) to scramble the DNA sequence, and achieved better image encryption effect. Li et al. (2017) used the fractional-order Lorenz hyper chaotic mapping (FOLHC) to direct the DNA operations (XOR, addition, subtraction). However, these methods were complex and the keys used were independent of the original images.

In this article, a medical image protection method based on dynamic deoxyribonucleic acid chain operation is proposed. The algorithm is tested against three kinds of medical images, and the performance, safety, efficiency of the developed algorithm evaluated against existing algorithms reported in the literature. The general arrangement for this article is as follows: First, DNA dynamic coding, FOHC, and DNA chain operation are introduced in section "Background Knowledge." Then, section "The Proposed Algorithm" introduces the proposed method. Next, section "Simulation Results" simulates the results. Security analyses are shown in the section "Security Analyses" and the conclusion in the section "Conclusion."

BACKGROUND KNOWLEDGE

DNA Coding Rule

There are four bases in a deoxyribonucleic acid chain. They are adenine (A), cytosine (C), guanine (G), thymine (T), in which A and T complement with each other, so do C and G. The four bases are denoted by the binary numbers of 00, 01, 10, and 11, normally. A total of 24 types of coding have been list (Xue et al., 2010b). However, because 0 and 1 complement with each other in binary, so do 01 and 10, and 00 and 11. Thus, only 8 of the 24 DNA coding rules satisfy the principle of base complementarity, as shown in **Table 1**.

By summarizing and categorizing the existing dynamic coding cases, it is observed that the existing cases fall into three categories: (1) those are based on the image block (column/row) dynamic coding (Akhavan et al., 2017); (2) those are based on the pixel dynamic encoding (Kalpana and Murali, 2015; Dagadu et al., 2019b; Wang X.Y. et al., 2020); and (3) those are based on the binary bit dynamic coding (Zhang J. et al., 2016). Because a DNA chain contains four bases, theoretically each base should appear in a random image with 25% of probability. The following equation can be used to calculate the base distribution rate of the above different DNA dynamic coding. The result is shown in **Table 2**.

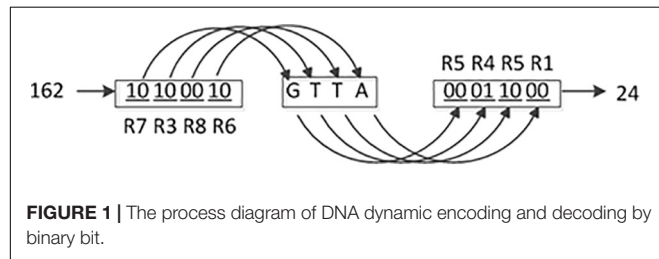
$$AP = \text{count}(A) \div (M \times N \times 4) \times 100\% \quad (1)$$

TABLE 1 | Eight kinds of DNA coding rules.

Binary	R1	R2	R3	R4	R5	R6	R7	R8
00	A	A	C	C	G	G	T	T
01	C	G	A	T	A	T	C	G
10	G	C	T	A	T	A	G	C
11	T	T	G	G	C	C	A	A

TABLE 2 | The base distribution rate of different kinds of dynamic coding.

Base distribution	By row (%)	By pixel (%)	By bit (proposed) (%)
AP	24.50	24.90	24.98
TP	24.65	24.90	25.04
CP	25.42	25.33	24.97
GP	25.43	24.87	25.01

**FIGURE 1** | The process diagram of DNA dynamic encoding and decoding by binary bit.

This equation uses base “A” as the example, where M and N are the numbers of row and column in the image; count (A) is the counting function of base “A.” A pixel consists of eight bits of binary, so four bases can represent one pixel. Here $M \times N \times 4$ is the total number of the possible base appearance. The distribution rate of “T,” “C,” and “G” can be calculated similarly.

From **Table 2**, it is found that the values of the base distribution rate of the DNA dynamic coding by binary bit are close to 25%, and the maximum deviation of the base distribution rates from 25% is 0.04%. Consequently, the DNA dynamic coding by binary bit is used for encoding and decoding in this study. A detailed coding example is shown in **Figure 1**. Where (R7, R3, R8, and R6) and (R5, R4, R5, and R1) are the encoding and decoding rules from **Table 1**, respectively, and they are controlled by the chaotic map. It can be seen that the image pixel value changes from 162 to 24. To our knowledge, this is the best DNA coding for image encryption.

Fractional-Order Chen Hyper Chaotic (FOHC) System

It is well known that hyper chaotic systems have much advantage over low-dimensional chaotic systems or multi-chaotic combination systems. Also the fractional-order hyper chaotic systems are superior to integer-order hyper chaotic systems in several aspects, including cross-correlation, self-correlation amplitude, pseudo-randomness, and the correlation and so on (Zhu et al., 2014).

Among the common fractional-order decomposition algorithms to solve the fractional-order chaotic system, the Adomian decomposition algorithm is the best choice since it has high precision, low complexity, and high computational efficiency (Donato and Giuseppe, 2008). Thus the Adomian decomposition algorithm is chosen to solve the FOHC in this study, and the generated chaotic sequence is then used for image encryption. The FOHC system model is described below:

$$\begin{cases} \frac{d^q}{dt^q}x = a(y-x) + w \\ \frac{d^q}{dt^q}y = bx - xz + cy \\ \frac{d^q}{dt^q}z = xy - dz \\ \frac{d^q}{dt^q}w = yz - ew \end{cases} \quad (2)$$

When $a = 38$, $b = 7$, $c = 12$, $d = 3$, $e = 0.7$, the system is in chaotic state and four chaotic sequences x , y , z , w are generated. The chaotic attractors for $q = 0.98$ are shown in **Figure 2**.

The Definition of DNA Chain Operation

The DNA chain is defined as:

$$C_n = C_h C_{h-1} \dots C_2 C_1 \quad (h \leq n)$$

Here, C_n is a DNA chain with length m . It can be broken into smaller DNA chains of $C_h, C_{h-1}, \dots, C_2, C_1$, with different lengths of $L_h, L_{h-1}, \dots, L_2, L_1$, respectively. Apparently, $m = L_h + L_{h-1} + \dots + L_2 + L_1$. In the DNA computing, there are several operations on the DNA chain to achieve base scrambling, including the deletion, the insertion, and the transposition operations. Their operating principles are shown in **Figure 3**.

THE PROPOSED ALGORITHM

The proposed new method in this study includes the following four steps. Firstly, the original image is encoded into a DNA matrix dynamically, by using a FOHC sequence. Secondly, the DNA matrix is scrambled by two other FOHC sequences. Thirdly, DNA dynamical chain operations are carried out by four FOHC sequences. At last, the DNA matrix generated is decoded into a binary matrix by a FOHC sequences, and the encrypted image is obtained after recombining the DNA chain. Eight chaotic sequences are used to complete the above four steps. The eight sequences are generated by two FOHC under different keys, which are obtained using the SHA-256 algorithm and the hamming distance. The detailed steps and the flowchart are shown in section “Key Generation,” section “Key Generation by SHA-256,” section “Key Generation by Hamming Distance, Generation of FOHC sequences,” section “Scrambling of the DNA sequence matrix,” section “The proposed algorithm Based on the DNA dynamic chains operation,” section “Generation of the dynamic DNA chains,” section “Deletion operation on the dynamic DNA chain,” section “Transposition operation on the dynamic DNA chain,” section “Insertion operation on the dynamic DNA chain,” and section “The proposed algorithm” and **Figure 4**.

Key Generation

Two kinds of keys are used as the initial values of the FOHC sequences, and they are generated by the SHA-256 algorithm

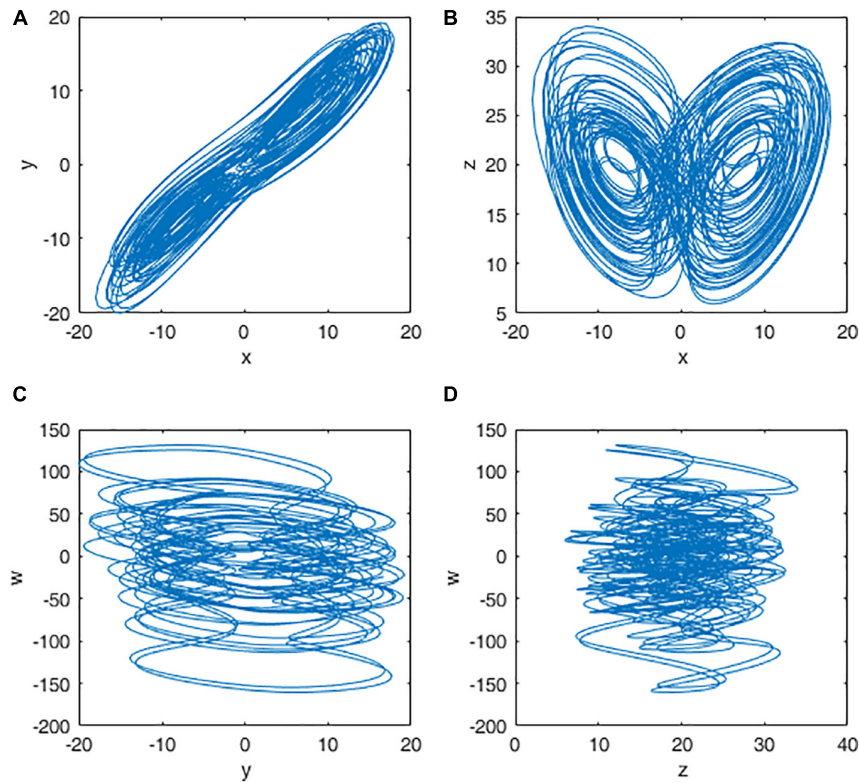


FIGURE 2 | Attractors of the fractional-order Chen hyper chaotic system: (A) x-y plane; (B) x-z plane; (C) y-w plane; and (D) z-w plane.

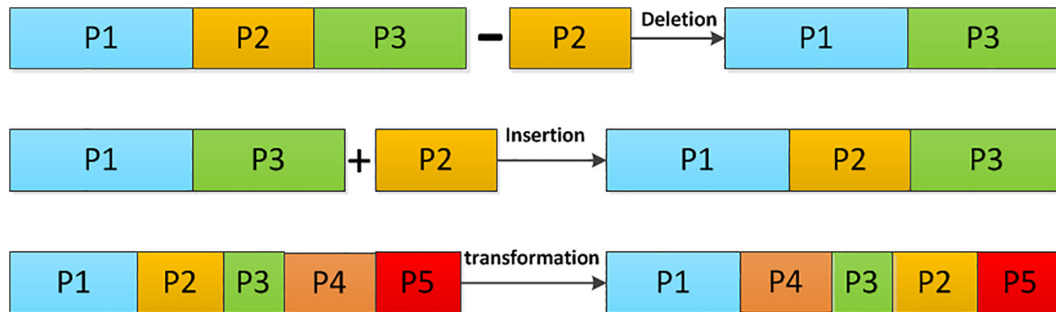


FIGURE 3 | The operating principle of DNA chains.

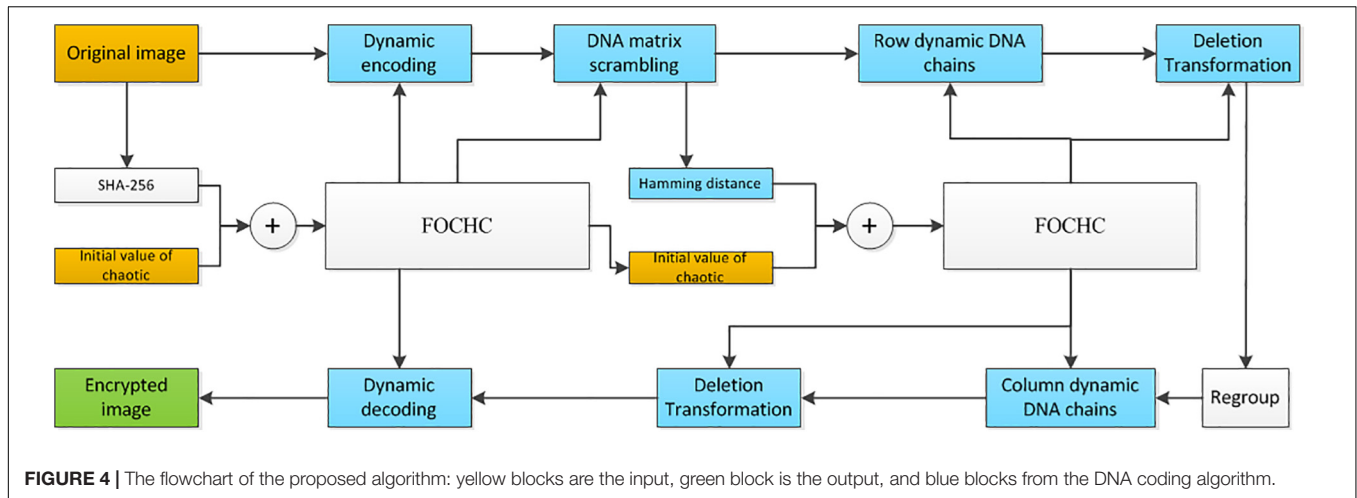
and the hamming distance, respectively. The former effectively defends against the KPA and CPA, and the latter enhances the diffusion ability of the bases.

Key Generation by SHA-256

The key generation in this algorithm depends on the SHA-256 function proposed in Belazi et al. (2019), because the result of the SHA-256 function is sensitive to the original image changes, since even one pixel change can result in a completely different hash value. In this study the 256-bit hash value is obtained by applying the SHA-256 function to the original image at first. The value is then converted into decimal numbers in groups of eight bits, and a decimal sequence K of length 32 is obtained, which

can be expressed as $K = \{K_1, K_2 \dots K_{32}\}$. The initial values are obtained via K ; the detail equation is defined below.

$$\begin{cases} k_1 = (K_1 \oplus K_2 \oplus K_3 \oplus K_4)/256 \\ k_2 = (K_5 \oplus K_6 \oplus K_7 \oplus K_8)/256 \\ k_3 = (K_9 \oplus K_{10} \oplus K_{11} \oplus K_{12})/256 \\ k_4 = (K_{13} \oplus K_{14} \oplus K_{15} \oplus K_{16})/256 \\ k_5 = (K_{17} \oplus K_{18} \oplus K_{19} \oplus K_{20})/256 \\ k_6 = (K_{21} \oplus K_{22} \oplus K_{23} \oplus K_{24})/256 \\ k_7 = (K_{25} \oplus K_{26} \oplus K_{27} \oplus K_{28})/256 \\ k_8 = (K_{29} \oplus K_{30} \oplus K_{31} \oplus K_{32})/256 \end{cases} \quad (3)$$



$$\begin{cases} x_s = k1 + x0 \\ y_s = k2 + y0 \\ z_s = k3 + z0 \\ w_s = k4 + w0 \end{cases} \quad (4)$$

Where x_s, y_s, z_s, w_s are the initial values obtained, and $x0, y0, z0$, and $w0$ are the initial values given.

Key Generation by Hamming Distance

There are four steps for key generation by the hamming distance, as detailed in the following:

Step 1: For a DNA matrix $A_DNA_matrix(m, n \times 4)$, calculate the hamming distance for every two rows and every two columns of the matrix, respectively. The row hamming distance $R_H = \{r_{h1}, r_{h2} \dots, r_{hi}, \dots, r_{hm/2}\}$ and the column hamming distance $C_H = \{c_{h1}, c_{h2} \dots, c_{hi} \dots, c_{hn} = 4/2\}$ are obtained. The equation for calculating the hamming distance is:

$$\begin{cases} D(M, N) = \sum_{i=0}^L d(m'_i, n'_i) \\ d(m'_i, n'_i) = \begin{cases} 0, & \text{if } m'_i = n'_i \\ 1, & \text{if } m'_i \neq n'_i \end{cases} \end{cases} \quad (5)$$

Where m'_i and n'_i are the i th base of the DNA chains M and N , respectively, and $D(M, N)$ is the hamming distance between M and N ;

Step 2: Calculate the average values of R_H and C_H , which are R'_h and C'_h , respectively.

Step 3: Extract the decimal parts p and q from R'_h and C'_h .

Step 4: The new initial values of FOCHC are obtained by Eq. (6). $x0', y0', z0'$, and $w0'$ are the given initial values, $k5, k6, k7, k8$ are calculated as described in section "Key Generation by SHA-256."

$$\begin{cases} X_h = k5 + p + x0' \\ Y_h = k6 + p + y0' \\ Z_h = k7 + q + z0' \\ W_h = k8 + q + w0' \end{cases} \quad (6)$$

Generation of FOCHC Sequences

The initial values generated using the FOCHC sequences, as described in section "Key generation," was input into the FOCHC system to produce four groups of chaotic sequences X, Y, Z, W after being iterated for $1000 + m \times n \times 4$ times. To eliminate the transient effects in the chaotic systems, the chaotic sequences were recalculated for 1,000 iterations before being used, and their length were $m \times n \times 4$.

Scrambling of the DNA Sequence Matrix

Step 1: Input the DNA sequence matrix $A (m, n \times 4)$, whose size is $(m, n \times 4)$;

Step 2: Use the following equations to transform the chaotic sequences:

$$\begin{cases} YY = \text{abs}(Y1 - \text{fix}(Y1)) \\ ZZ = \text{abs}(Z1 - \text{fix}(Z1)) \end{cases} \quad (7)$$

Where, $Y1$ and $Z1$ are the FOCHC sequences. The length of $Y1$ is m , and the length of $Z1$ is $n \times 4$. $\text{fix}(\cdot)$ is the rounding function. $\text{abs}(\cdot)$ is the absolute value function.

Step 3: Sort YY and ZZ , respectively, to obtain the index values By and Bz .

$$\begin{cases} [\sim, By] = \text{sort}(YY) \\ [\sim, Bz] = \text{sort}(ZZ) \end{cases} \quad (8)$$

Step 4: Use the following equation to scramble A , and obtain the matrix $A_scrambling$.

$$A_scrambling(i, j) = A(By(i), Bz(j)); \quad (9)$$

Where $i = 1, 2 \dots m, j = 1, 2 \dots n \times 4$.

The Proposed Algorithm Based on the DNA Dynamic Chains Operation

Through DNA dynamic chain operation, the algorithm proposed changes the position of the base, which leads to changes of the pixel values in the image. As shown in Figure 5, each row in the DNA sequence matrix is divided into chains of

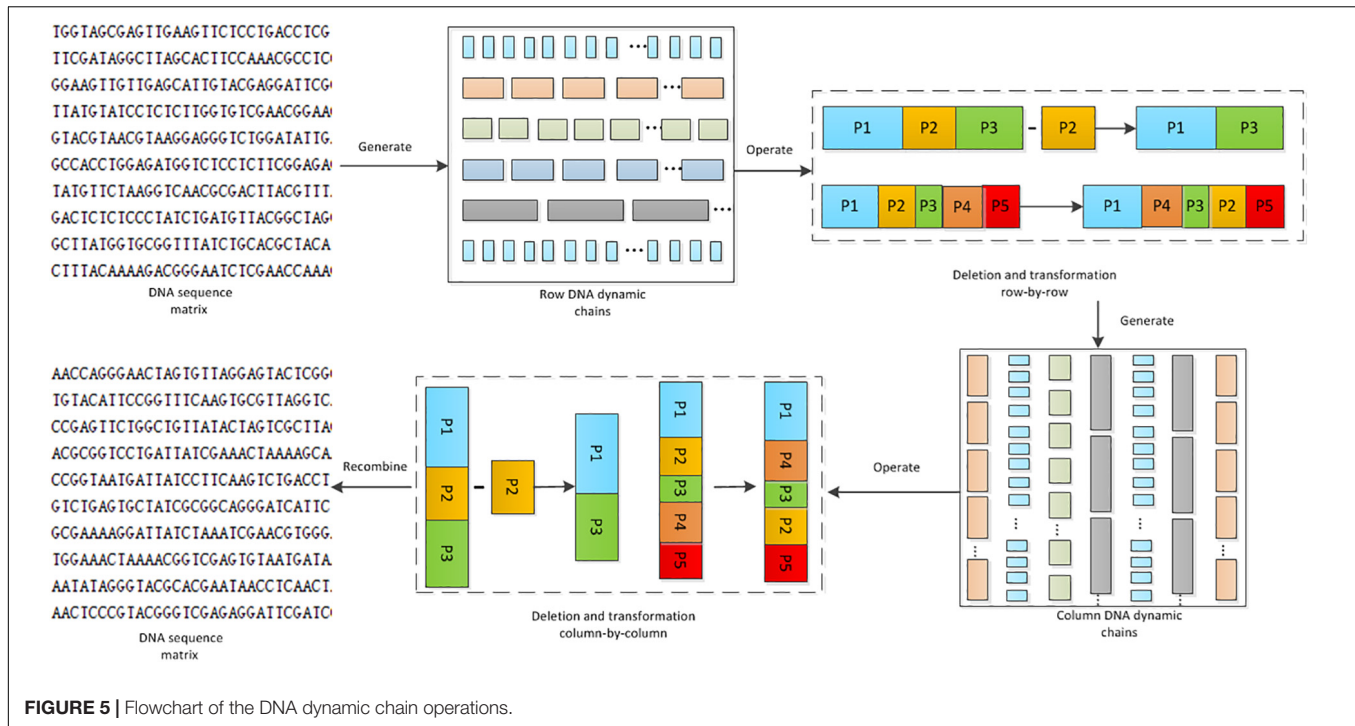


FIGURE 5 | Flowchart of the DNA dynamic chain operations.

different lengths. Deletion operation and transposition operation are applied on these chains. This is the first shuffle process on the DNA sequence matrices. To eliminate the block effect, generation of the column DNA dynamic chain, and the deletion and transposition operations are conducted again. Thus the DNA sequence matrices are scrambled for the second time. In the whole process, the length of the DNA chain is dynamic, so is the operation, and the DNA base is completely disrupted.

Generation of the Dynamic DNA Chains

DNA sequence matrix is divided into DNA chains of different lengths by rows or columns. The length of the chain is controlled by the hyper-chaotic sequence. The following explains the detailed steps for generating DNA dynamic chains row-by-row. Generation of the DNA dynamic chains column-by-column can be implemented in a similar way, except that the DNA matrix needs to be transposed before the generation.

Step 1: A FOCHC sequence x_1 is transformed using the following equation:

$$x_1 = \text{mod}(\text{fix}(\text{abs}(x_1 - \text{fix}(x_1)) \times 10^{14}), 4) + 1 \quad (10)$$

Where, $\text{fix}(\cdot)$ is the rounding function, and $\text{abs}(\cdot)$ is the absolute value function.

Step 2: The length of chains $a(i)$ is determined with the following equation:

$$a(i) = \begin{cases} 16, & \text{if } x_1(i) = 1 \\ 8, & \text{if } x_1(i) = 2 \\ 4, & \text{if } x_1(i) = 3 \\ 2, & \text{if } x_1(i) = 4 \end{cases} \quad (11)$$

Where, $i = 1, 2, 3 \dots m$, m is the size of the rows in the DNA sequence matrix.

Step 3: DNA dynamic chain matrices are obtained by decomposing each row of the DNA matrix sequences, according to different lengths, as shown in the following equation:

$$\text{Row_chain} = \text{DNA_decompose}(\text{DNA_matrix}(i, :), a(i)) \quad (12)$$

Where $\text{DNA_decompose}(\cdot)$ is a generation function of the DNA dynamic chain, which means that the i th row in the DNA_matrix is decomposed into chains whose lengths are defined in $a(i)$.

Deletion Operation on the Dynamic DNA Chain

Deletion operation on the DNA dynamic chains for each row or column is implemented using the chaotic sequence. The deletion operation function $\text{deletion}(A, X)$ is defined as following:

In the function $\text{deletion}(A, X)$, A is a chain set of a row, which can be represented as $A = \{a_1, a_2, a_3 \dots a_n\}$, where n is the number of chains. a_i is the i th DNA chain. X is a chaotic sequence, which can be represented as $X = \{x_1, x_2, x_3 \dots x_n\}$. x_i ($x_i \in (0, 1)$) is the i th element in the chaotic sequence. Note that the length of the chaotic sequence and that of the DNA chain set are the same. Carry out deletion for a_i when $x_i < 0.5$, otherwise save the chain. Supposing that the a_i chain has been deleted, the a_i chain is moved to the end of the DNA chain. Other deleted chains can be processed in the same way.

Transposition Operation on the Dynamic DNA Chain

Transposition operation on the DNA dynamic chains for each row or column is conducted using the chaotic sequence. The Transposition operation function $\text{Transposition}(A, X)$ is defined as following:

In the function $\text{Transposition}(A, X)$, the definitions of A and X are the same as those in section “Deletion Operation on the Dynamic DNA Chain.” A new sequence X' is obtained by transposing X , with a_i and $a_{i'}$ exchanged, where i' is the location of the i th element in X' .

Insertion Operation on the Dynamic DNA Chain

Insertion operation is used for the decryption process, the insertion operation function $\text{insertion}(A, X)$ is defined as following:

In the function $\text{insertion}(A, X)$, A is a chain set of a row, which can be represented as $A = \{a_1, a_2, a_3 \dots a_n\}$ where n is the number of chains. a_i is the i th DNA chain. X is a chaotic sequence, which can be represented as $X = \{x_1, x_2, x_3 \dots x_n\}$. x_i ($x_i \in (0, 1)$) is the i th element in the chaotic sequence. Note that the length of the chaotic sequence and that of the DNA chain set are the same. Set $\text{count} = 0$, when $x_i < 0.5$, carry out $\text{count} = \text{count} + 1$, count is the number of deleted DNA chains in the encryption process. Carry out $e_j = a_{n-\text{count}+j}$, where $j = 1, 2, \dots, \text{count}$, here e_j is the j th deleted DNA chain. Set $q = 1, j = 1$, if $x_i < 0.5$, $f_i = e_j, j++$; else $f_i = a_q, q++$. $A1 = \{f_1, f_2, f_3 \dots f_n\}$ is obtained after insertion operation. Other inserted chains can be processed in the same way.

The Proposed Algorithm

The detailed steps of the proposed algorithm are listed below.

Step 1: Input the initial values x_0, y_0, z_0, w_0 and an 8-bit image A (m, n), where m and n define the size of the image. A binary matrix $A'(m, n \times 8)$ is obtained by transforming A (m, n).

Step 2: Use the SHA-256 function to generate the chaotic initial values x_s, y_s, z_s, w_s , as explained in section “Key Generation.”

Step 3: Produce four chaotic sequences X, Y, Z, W using FOCHC with the initial values x_s, y_s, z_s, w_s , as detailed in section “Generation of FOCHC Sequences.”

Step 4: Generate the matrix $A_{\text{encode}}(m, n \times 4)$ using the chaotic sequence $X1$ to encode $A'(m, n \times 8)$, as detailed in section “DNA Coding Rule.” $X1$ is obtained using the following equation:

$$X1 = \text{mod}(\text{fix}(\text{abs}(X - \text{fix}(X)) \times 10^{14}), 8) + 1 \quad (13)$$

Where $\text{fix}(\cdot)$ is the rounding function, and $\text{abs}(\cdot)$ is the absolute value function.

Step 5: Scramble $A_{\text{encode}}(m, n \times 4)$ by using two chaotic sequences Y, Z , as explained in section “Scrambling of the DNA Sequence Matrix.” This produces the matrix $A_{\text{DNA_scrambling}}(m, n \times 4)$.

Step 6: Calculate the hamming distance of $A_{\text{DNA_scrambling}}$ to obtain the new initial values x_h, y_h, z_h, w_h , as explained in section “Key

Generation by Hamming Distance,” which are then used to generate four chaotic sequences X', Y', Z', W' .

Step 7: Divide $A_{\text{DNA_scrambling}}$ into different lengths of DNA dynamic chains row-by-row by using the chaotic sequence X' , as explained in section “Generation of the Dynamic DNA Chains.” This produces the DNA chain matrix $A_{\text{Row_Chain}}$.

Step 8: Conduct the deletion and transposition operations on $A_{\text{Row_Chain}}$ using the chaotic sequence Y' , as described in section “Deletion Operation on the Dynamic DNA Chain” and section “Transposition Operation on the Dynamic DNA Chain.” After recombining the data, this produces the matrix $A_{\text{Row_operation}}$.

Step 9: Divide $A_{\text{Row_operation}}$ into different lengths of DNA dynamic chains column-by-column using the chaotic sequence Z' , as described in section “Generation of the Dynamic DNA Chains.” This produces the DNA chain matrix $A_{\text{Column_Chain}}$.

Step 10: Conduct the deletion and transposition operations on $A_{\text{Column_Chain}}$ using the chaotic sequence W' , as explained in section “Deletion Operation on the Dynamic DNA Chain” and section “Transposition Operation on the Dynamic DNA Chain.” After recombining the data, this produces the matrix $A_{\text{Column_operation}}$.

Step 11: Decode the matrix $A_{\text{Column_Chain}}$ dynamically using the chaotic sequence $W1$, as explained in section “DNA Coding Rule.” This produces the new matrix A_{decode} . $W1$ is calculated using the following equation:

$$W1 = \text{mod}(\text{fix}(\text{abs}(W - \text{fix}(W)) \times 10^{14}), 8) + 1 \quad (14)$$

Step 12: Recombine A_{decode} to obtain the encrypted image B .

The decryption algorithm is the inverse of the encryption algorithm detailed above; also the delete operation needs to be replaced with the insert operation.

SIMULATION RESULTS

The proposed algorithm explained above is then tested on three kinds of medical images of MRI, CT, and X-ray. All of the experimental data are 512×512 images extracted from the database¹. Matlab 2019a is used to code the proposed algorithm, and the code is running in the 64-bit Window 7 environment with 8GB RAM and the i5-7200U CPU. The keys of the encryption algorithm presented in this article are composed of the hash value, the row and column hamming distance values, and two sets of chaotic initial values, as shown in **Table 3**. **Table 4** lists the experimental results using the extracted images.

As illustrated in **Table 4**, no any useful information can be drawn from the encrypted images for all the three types of medical images tested, while the decrypted images show no difference when compared with the original images. From

¹<https://medpix.nlm.nih>

TABLE 3 | The key of the proposed algorithm.

Composition of the key	The key of encryption and decryption
Hash value	c515f75a2b612d728e3356b7b53925 32044172d647291f10f00075107f161bd9
Hamming distance	$R_h' = 393425, C_h' = 393058$
Initial value of two	$x_0 = 0.12, y_0 = 0.35, z_0 = 0.68, w_0 = 0.42,$ $x_0' = 0.37, y_0' = 0.54, z_0' = 0.89, w_0' = 0.76$
FOCHC system	

the point of view of visual inspection, the proposed algorithm worked satisfactorily.

Security Analyses

Key Space Evaluation

Section “Simulation Results” shows that the keys consist of three parts: the hash value, the hamming distances and the chaotic initial value. Therefore, in addition to the hash value and the hamming distances, there are another eight keys in the proposed algorithm. They are $x_0, y_0, z_0, w_0, x_0', y_0', z_0',$ and w_0' . All of them have 14 bits precision, so the key space is $(10^{14})^7 = 10^{112} \approx 2^{372}$. The SHA-256 value with the complexity of the finest attack is 2^{128} . All of them are larger than 2^{100} (Wang X.Y. et al., 2020). Thus the key space is large enough to withstand BFA. Moreover, generation of the key depends on the original image, which forms a one-time pad scheme, and makes it difficult for the attacker to predict the encryption key.

Key Sensitivity Evaluation

The keys are used in the encryption and decryption process, and the key sensitivity means that when the encryption keys each change slightly, the image generated will be completely different to the initial encrypted image. Similarly, when the decryption keys each change slightly, the correct decryption image cannot be obtained. This article tests the sensitivity of the keys from the aspects of encryption and decryption separately. **Figure 6** and **Table 5** show the results.

Figure 6B is the encrypted image of the “MRI-Knee-joint” with the keys listed in **Table 3**. **Figure 6C** is the same as **Figure 6B** beside x_0 is changed to x_0+t , where $t = 0.000000000000001$. **Figure 6D** shows that **Figures 6B,C** have much difference. To further observe the sensitivity of the keys, one key is changed slightly (add t) and the other keys kept unchanged, after that, these keys are used to encrypt the images “MRI-knee,” “CT-abdominal,” and “X-ray-pelvic,” respectively. At last, the difference rate of the encrypted images before and after the slight key changes is calculated. All the average difference rates in **Table 5** are above 99.50%, which is very close to 100%. On the other hand, the original images are restored using three types of keys (x_0, R_h' and the hash value) changed slightly. **Figure 6E** is the decrypted image with $x_0 = x_0+t$ and other keys unchanged. **Figure 6F** is the decrypted image with $x_0 = R_h' + 1$ and other keys unchanged. **Figure 6G** is the decrypted image with the hash value + ‘1’ and other keys unchanged. **Figures 6E–G** show that the original image cannot be restored when the keys change slightly. The original image can only be recovered when the keys

are correct, as shown in **Figure 6H**. This test proved that the key sensitivity is very high, and the algorithm is robust against the exhaustive attacks.

Statistical Analysis Evaluation

The Histogram Evaluation of the Decrypted Image

The distribution of the histogram is evaluated by observing the histogram for the encrypted image and calculating the variance of the histogram. The more uniform the histogram distribution for the encrypted image, the stronger the ability of anti-statistical analysis. The variance of the histogram is defined as follow:

$$\text{Var}(M) = \frac{1}{n^2} \times \sum_{i=1}^n \sum_{j=1}^n \frac{1}{2} (m_i - m_j)^2 \quad (15)$$

Where n is the gray scale value, here it is set as $n = 256$. m_i is the number of pixels whose gray values are equal to i , with i being the value in the histogram value. m_i is the same as m_j .

Figures 7A,C,E show the histogram of three kinds of original images, and **Figures 7B,D,F** show the histogram corresponding to the images after encrypting. Observation suggests that **Figures 7B,D,F** looks very uniform. In **Table 6**, the variance values for the encrypted images are significantly reduced. In addition, the average of the variance values is lower than other algorithm, as detail in the **Table 7**. In summary, it is difficult to extract original information through statistical analysis on the histogram.

Correlation Coefficient Evaluation


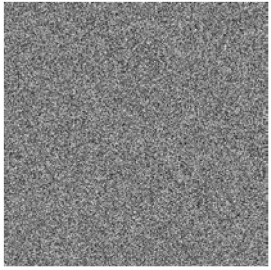

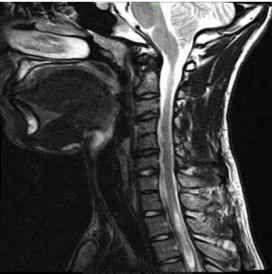
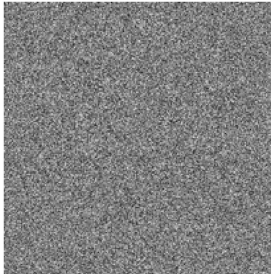
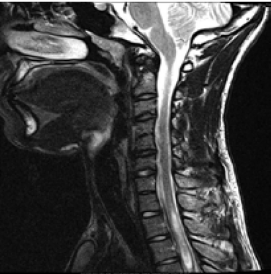

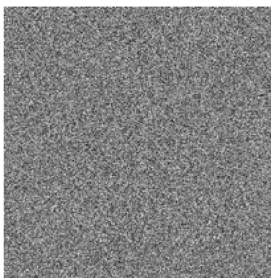

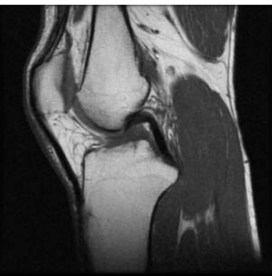
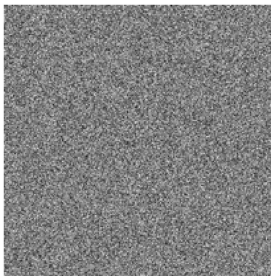


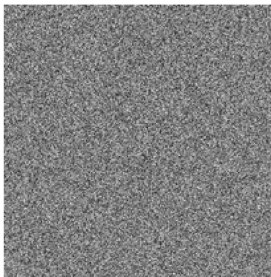

Usually, the recognizable images have high correlation, so correlation evaluation is one of the effective means to measure the encryption effect. The closer the correlation coefficient to 0, the better the encryption result. The correlation coefficient is defined as follows:

$$r_{xy} = \frac{\frac{1}{N} \times \sum_{i=1}^N \left(x_i - \frac{1}{N} \times \sum_{i=1}^N x_i \right) \left(y_i - \frac{1}{N} \times \sum_{i=1}^N y_i \right)}{\sqrt{\frac{1}{N} \times \sum_{i=1}^N \left(x_i - \left(\frac{1}{N} \times \sum_{i=1}^N x_i \right)^2 \right)} \sqrt{\frac{1}{N} \times \sum_{i=1}^N \left(y_i - \left(\frac{1}{N} \times \sum_{i=1}^N y_i \right)^2 \right)}} \quad (16)$$

Where x_i and y_i are adjacent pixels selected randomly in three directions (horizontal, vertical, and diagonal). For evaluation 8,000 pairs of adjacent pixels are chosen for the test.


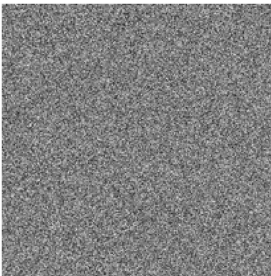


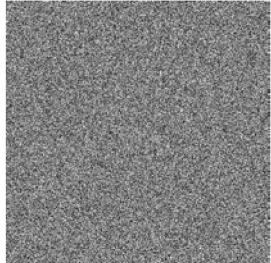
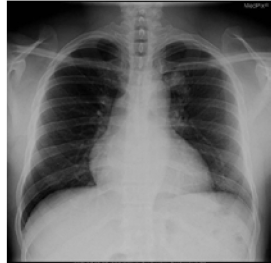
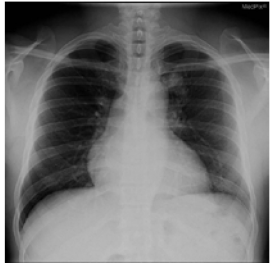

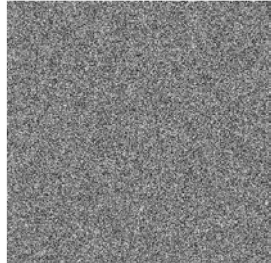
Figures 8A–C shows the correlation coefficients in the three directions, respectively. Obviously, the distribution of point sets is concentrated in the left subfigure of **Figures 8A–C**. On the contrary, the distribution of point sets is discrete in the right subfigure of **Figures 8A–C**. The values of correlation coefficients are shown in **Table 8**. The correlation coefficients of the encrypted image are very close to 0. From the comparison results, it is also better than other algorithms, which are shown in **Table 9**. This sufficiently demonstrates that it is difficult for attackers to obtain a cipher image by statistical pixel correlation.

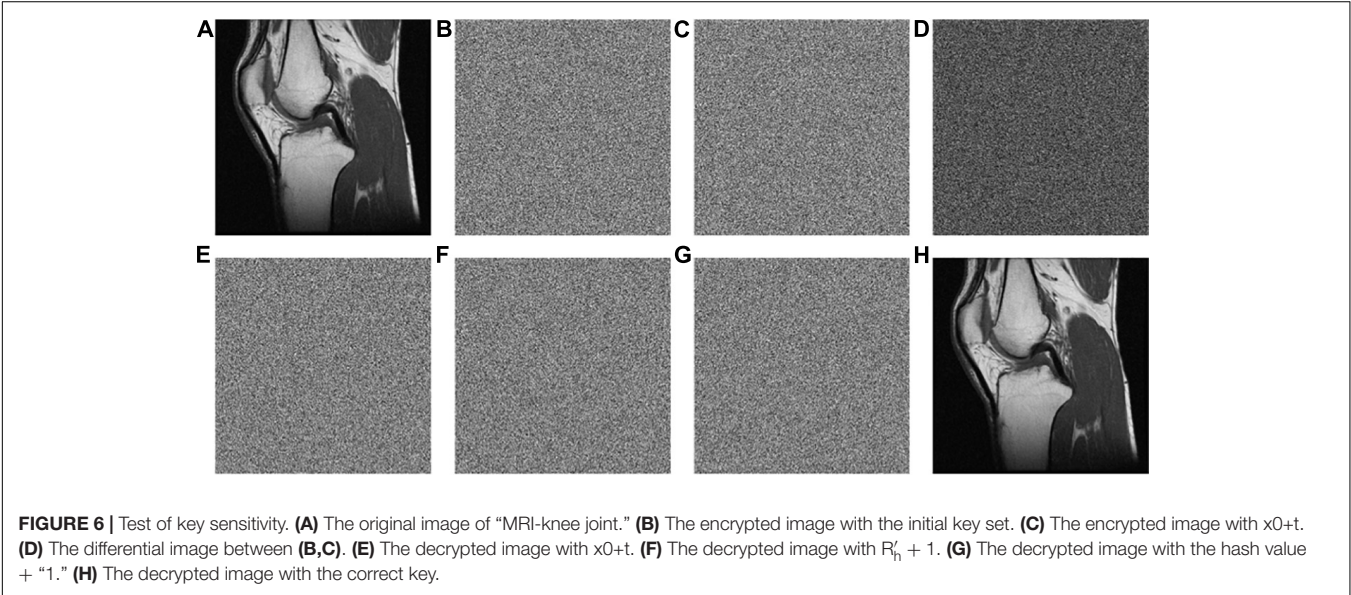
TABLE 4 | Results of the proposed algorithm.

Types	Name	Original image	Encrypted image	Decrypted image
MRI	A child's brain			
	Vertebrae cervicales			
	Brain			
	Knee-joint			
	Chest reinforcement			
CT				

(Continued)

TABLE 4 | Continued

Types	Name	Original image	Encrypted image	Decrypted image
	Abdominal reinforcement			
X-ray	Chest			
	Pelvic			



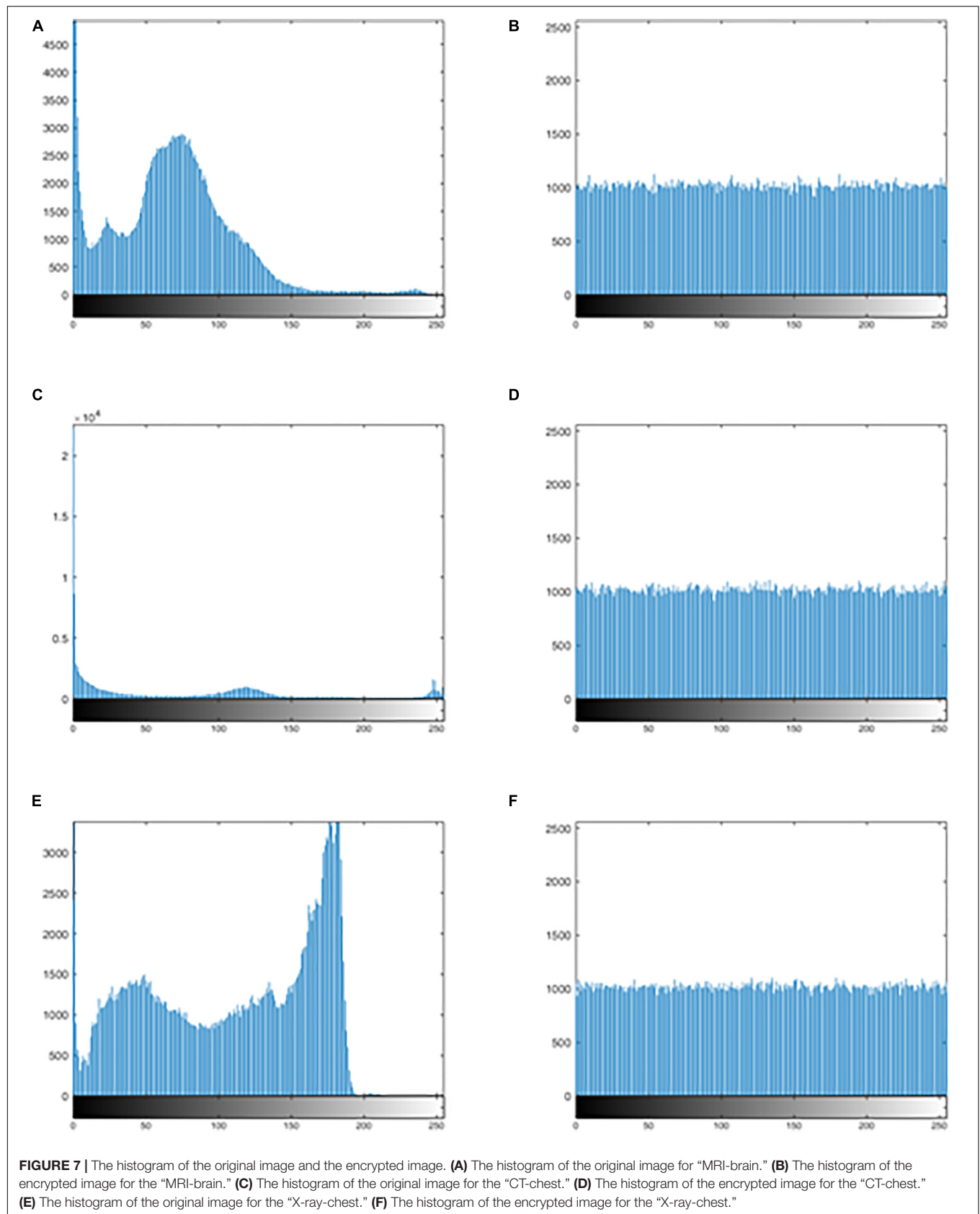


TABLE 5 | Difference rate of two encrypted image obtained by slightly different keys.

Image	Difference rate (%)							
	x0+t	y0+t	z0+t	w0+t	x0'+t	y0'+t	z0'+t	w0'+t
MRI-knee	99.6101	99.6326	99.6025	99.6014	99.5232	99.5304	99.5224	99.5136
CT-abdominal	99.5964	99.6101	99.6307	99.6044	99.5209	99.4858	99.5049	99.5335
X-ray-Pelvic	99.6227	99.6174	99.6014	99.6106	99.5197	99.5182	99.5537	99.5308
Average	99.6097	99.6200	99.6115	99.6054	99.5220	99.5115	99.5270	99.5260

TABLE 6 | The variance of the histogram.

Image	Original image	Encrypted image
MRI-child's brain	1.9859×10^7	1.0956×10^3
MRI-cervical vertebra	2.0992×10^6	1.0379×10^3
MRI-brain	2.8428×10^6	1.1222×10^3
MRI-knee-joint	2.6840×10^6	871.1016
CT-chest	8.0452×10^7	1.0736×10^3
CT-abdominal	4.4843×10^7	907.8516
X-ray-chest	7.8361×10^5	1.0098×10^3
X-ray-pelvic	2.0195×10^6	843.0469
Average	2.1938×10^7	995.1375

TABLE 7 | The variance of the histogram comparison.

Algorithm	Variance
Proposed	995.1375
Chai et al., 2019	1051
Liu et al., 2019	1341

Global and Local Information Entropy Evaluation

There is redundancy in any image, which is related to the probability or uncertainty of each pixel in the image. Usually, this uncertainty is measured with the global and the local information entropy (Zhang et al., 2012; Wu et al., 2013). The global information entropy is a measure of the distribution of all pixels in an image, while the local information entropy is a measure of the distribution of pixel values in an image block. Compared with the global information entropy, the local information entropy is

more efficient, accurate and consistent in judging the pixel values distribution situation of the image.

It is known that the global information entropy of an ideal random image is 8. Also Wu et al., 2013 shows that the local entropy values for the ideal random image blocks of 16×16 and 32×32 are 7.1749 and 7.8087, respectively. **Table 10** lists the global and local information entropy of all the encrypted images processed using the proposed algorithm. It is observed that the average global information entropy of all the encrypted image is 7.9993, and the average local entropy are 7.1715 for the 16×16 block and 7.8016 for the 32×32 block. All of them are close to the ideal values. Furthermore, comparison with other algorithms is shown in **Table 11**, which also includes the comparison of global information entropy corresponding to the image encryption algorithm. Apparently, the information entropy of the proposed algorithm in this study is superior to those for other algorithms.

Plaintext Sensitivity (Differential Attack)

Differential attack is one of the common attack methods used by cryptanalysts. Its main idea is to encrypt two original images with tiny change and no change, respectively, then compare the relationship between the encrypted image before and after change, and predict the encryption key, so as to decipher the encryption algorithm. NPCR and UACI in the Reference (Wang X.Y. et al., 2020) are used here to test the ability of the algorithms to resist differential attack.

The NPCR and UACI values of the encrypted images obtained from the two slightly changed images are shown in **Table 12**. The average values obtained are 99.6191% and 33.4815%,

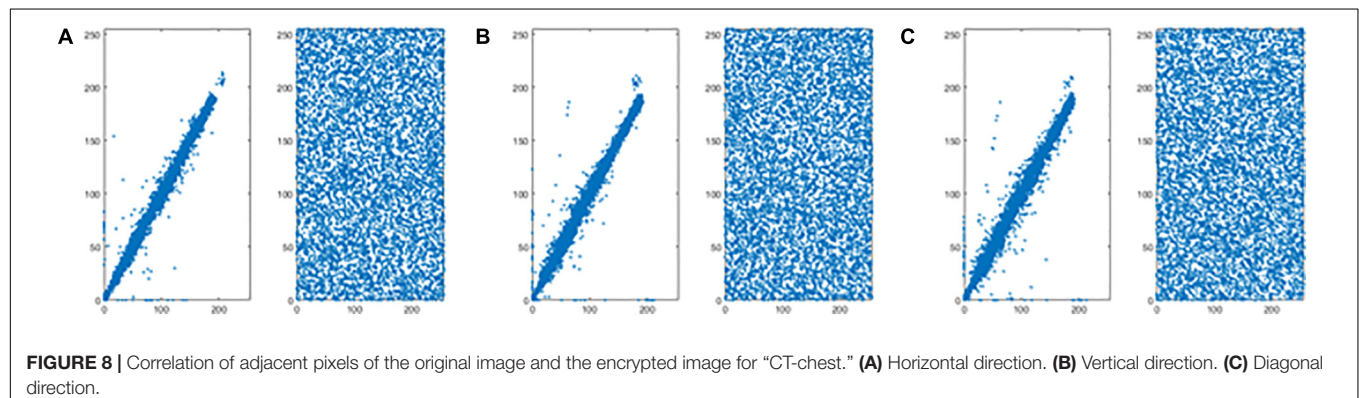


TABLE 8 | Correlation coefficients of two adjacent pixels of the original and the encrypted images.

Image	Original image			Encrypted image		
	H	V	D	H	V	D
MRI-child's brain	0.9757	0.9806	0.9563	0.0013	6.5787×10^{-4}	-0.0049
MRI-cervical vertebra	0.9714	0.9738	0.9495	-1.6397×10^{-4}	9.0951×10^{-4}	-0.0015
MRI-brain	0.9812	0.9809	0.9607	0.0036	-0.0075	0.0021
MRI-knee-joint	0.9928	0.9970	0.9912	-9.6693×10^{-4}	-6.6741×10^{-4}	4.7456×10^{-4}
CT-chest	0.9746	0.9603	0.9455	0.0012	-2.7242×10^{-4}	0.0011
CT-abdominal	0.9788	0.9827	0.9656	-9.1875×10^{-4}	3.8182×10^{-4}	-0.0018
X-ray-chest	0.9946	0.9779	0.9774	0.0011	6.4868×10^{-4}	0.0049
X-ray-pelvic	0.9340	0.9529	0.9114	0.0012	-2.0257×10^{-4}	-4.7550×10^{-4}

TABLE 9 | Correlation coefficients comparison.

Algorithm	H	V	D
Proposed (CT-chest)	0.0012	-0.0003	0.0011
Hossein et al., 2018 (medical image)	0.0031	0.0029	0.0013
Belazi et al., 2019 (medical image)	0.0013	-0.0049	0.0057
Dagadu et al., 2019a (medical image)	-0.0016	0.0043	-0.0061
Wang X.Y. et al., 2020	-0.0021	0.0009	0.0003
Hossein et al., 2020	0.0059	0.0029	0.0018
Wu et al., 2019	0.0158	0.0023	-0.0336
Zhang X.C. et al., 2017	0.0082	0.0032	0.0150

TABLE 10 | The entropy of eight medical encrypted images.

Image	Entropy	Local entropy (16 × 16)	Local entropy (32 × 32)
MRI-child's brain	7.9992	7.1530	7.7962
MRI-Vertebrae cervicales	7.9993	7.1705	7.7995
MRI-Brain	7.9992	7.1800	7.8037
MRI-Kneejoint	7.9994	7.1935	7.8010
CT-Chest	7.9993	7.1688	7.8034
CT-Abdominal	7.9994	7.1567	7.8007
X-ray-Chest	7.9993	7.1615	7.8037
X-ray-Pelvic	7.9994	7.1883	7.8046
Average	7.9993	7.1715	7.8016

which are higher than the values for other algorithms, as detailed in **Table 13**.

Noise Attack

Images can be contaminated by noise during transmission. To analyze the anti-noise capability, the same encrypted image is attacked by the salt and the pepper noise with the density of 0.002, 0.005, 0.05, 0.1, 0.25, and 0.5, respectively, **Figures 9A–F** lists the decrypted images after being attacked with the salt and the pepper noise. All of them can clearly show the outline and texture of the original image. Further, equation (17) is used to calculate the PSNR between the original image and **Figures 9A–F**. The PSNR of the proposed algorithm is then compared with that of other algorithms. Results in **Figure 9** and **Table 14** show that the

TABLE 11 | Global information entropy comparison.

Algorithm	Entropy
Proposed	7.9993
Hossein et al., 2018 (medical image)	7.9990
Belazi et al., 2019 (medical image)	7.9974
Dagadu et al., 2019a (medical image)	7.9993
Hua et al., 2018 (medical image)	7.9981
Wang X.Y. et al., 2020	7.9971
Azimi and Ahadpour, 2020	7.9988
Hossein et al., 2020	7.9989
Wu et al., 2019	7.99895
Yang et al., 2019	7.9964

TABLE 12 | The result of differential attack (NPCR, UACI).

Image	NPCR (%)	UACI (%)
MRI-child's brain	99.6124	33.4986
MRI-Vertebrae cervicales	99.6162	33.4415
MRI-Brain	99.6185	33.4285
MRI-Knee joint	99.6220	33.4547
CT-Chest	99.6254	33.4474
CT-Abdominal	99.6334	33.5747
X-ray-Chest	99.6059	33.4359
X-ray-Pelvic	99.6193	33.5704
Average	99.6191	33.4815

proposed algorithm is immune to the salt and the pepper noise.

$$PSNR = 10 \lg \frac{255 \times 255 MN}{\sum_{i=1}^M \sum_{j=1}^N |x'(i, j) - x(i, j)|^2} \quad (17)$$

Occlusion Attack

To analyze the anti-occlusion capability of the proposed algorithm, the same cipher image is occluded with 1/16, 1/8, 1/4, and 1/2, respectively. Then, the blocked images are decrypted with the proposed algorithm. Take the image “CT-chest” as the example, which are shown in **Figures 10A–H**. As shown, all the encrypted images which are occluded with different area are recovered successfully. In all of them the information of the

TABLE 13 | Comparison of the average differential attack (NPCR,UACI) by different encryption algorithms.

Algorithm	NPCR (%)	UACI (%)
Proposed	99.6191	33.4815
Hossein et al., 2018 (medical image)	99.1349	33.1633
Belazi et al., 2019 (medical image)	99.6536	33.4121
Dagadu et al., 2019a (medical image)	99.6100	33.5075
Wang X.Y. et al., 2020	99.5956	33.4588
Wang et al., 2018	99.5700	32.3800
Hossein et al., 2020	99.5438	33.4742
Wu et al., 2019	99.5666	33.3966
Yang et al., 2019	99.6105	33.4694

original image can be identified. Additionally, comparison of the ability to resist occlusion attack for the proposed and other algorithm by PSNR is shown in **Table 15**. Obviously the proposed algorithm is superior to others.

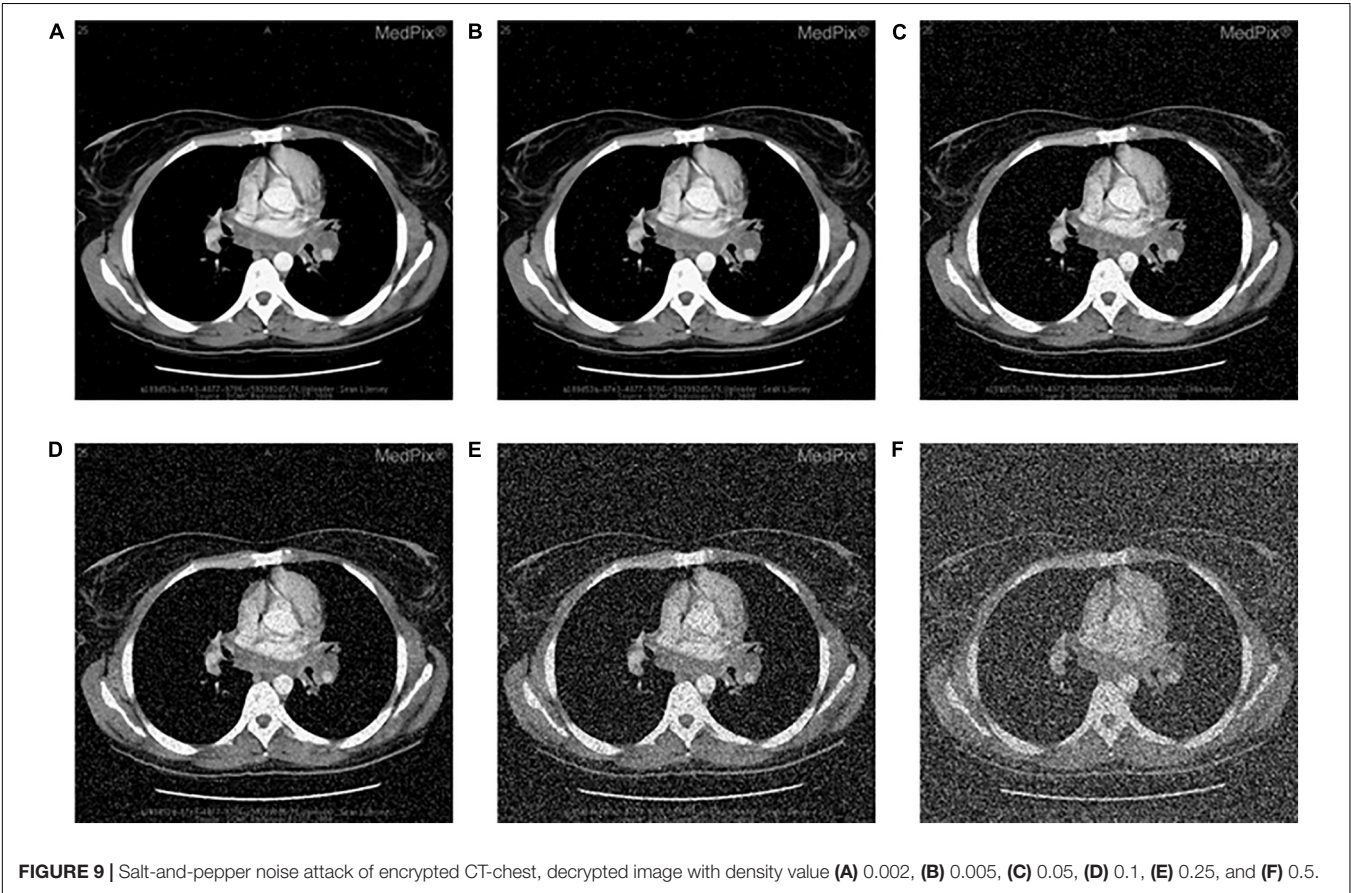
Known-Plaintext and Chosen-Plaintext Attacks

Kerckhoffs’ principles in cryptography state that encryption and decryption algorithms are known or transparent in a cryptosystem. Therefore, the security of the cryptosystem

TABLE 14 | PSNR (db) between the original and the decrypted images under noise.

Algorithm	Density of salt & pepper noise					
	0.002	0.005	0.05	0.1	0.25	0.5
Proposed	33.8870	29.6681	19.7353	16.6353	12.4552	9.1951
Belazi et al., 2019	32.8396	28.7068	18.8395	15.8599	12.2262	9.8903
Zhou et al., 2015	26.1682	21.9976	12.8812	10.6900	8.8973	8.5504
Hua and Zhou, 2017	8.5900	8.5625	8.5514	8.5476	8.5454	8.5428
Hua et al., 2018	29.8380	25.6571	15.8923	13.1335	10.2166	8.8271
Liu et al., 2016	/	19.1553	19.5829	11.9524	/	/

depends on the key rather than the encryption algorithm itself. By exploring the relationship between the key and the ciphertext or the plaintext and ciphertext, the attacker obtains the valid equivalent key, and then decrypts the original image. The main methods include the ciphertext-only attack, KPA, CPA, and chosen ciphertext attack (CCA). Among these attack methods, CPA is recognized as the strongest attack method, so the ability of the current algorithm to resist CPA is analyzed here. As for the key generation, the encryption keys of this algorithm are generated by the SHA-256 function and the



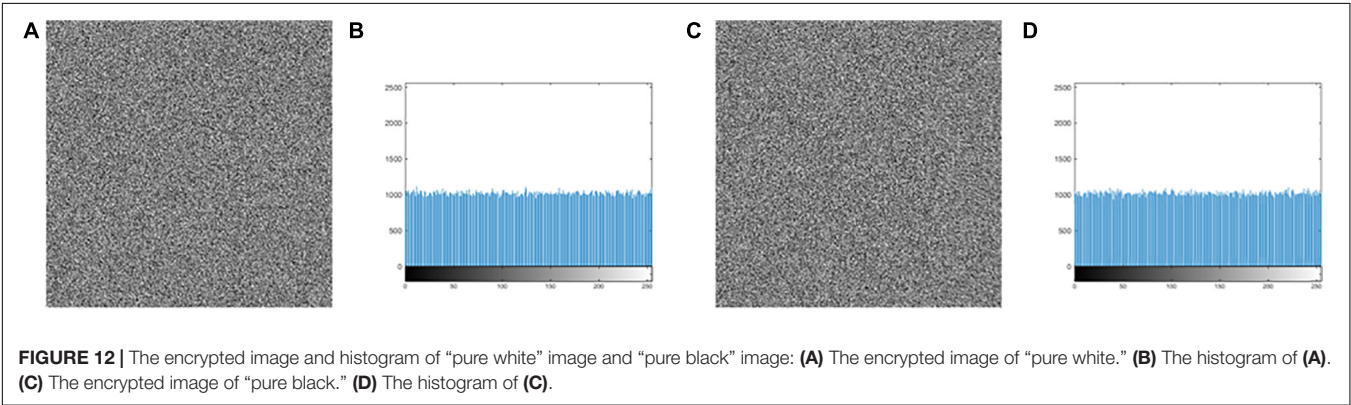
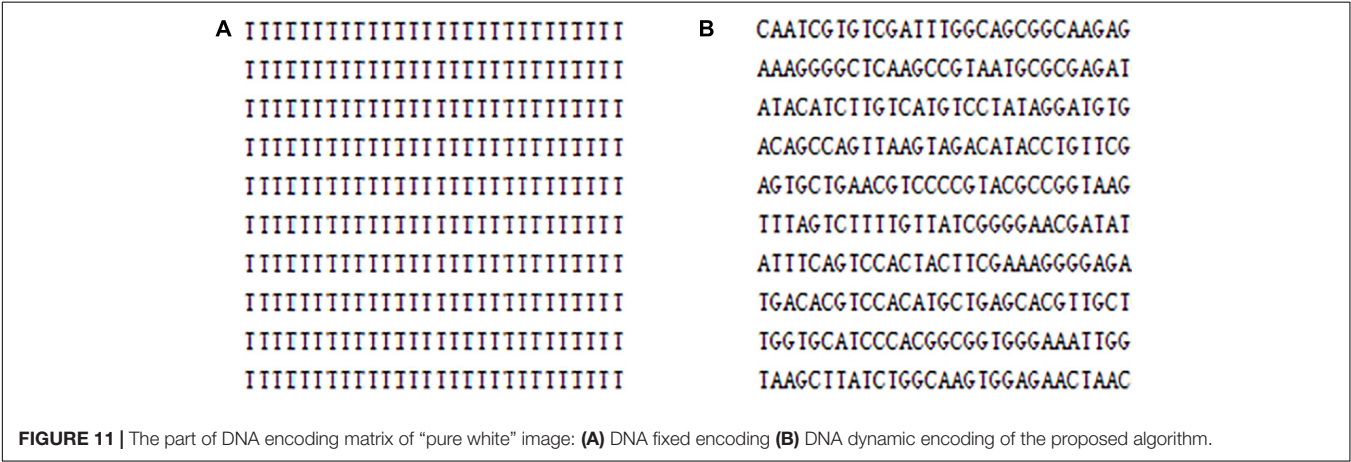
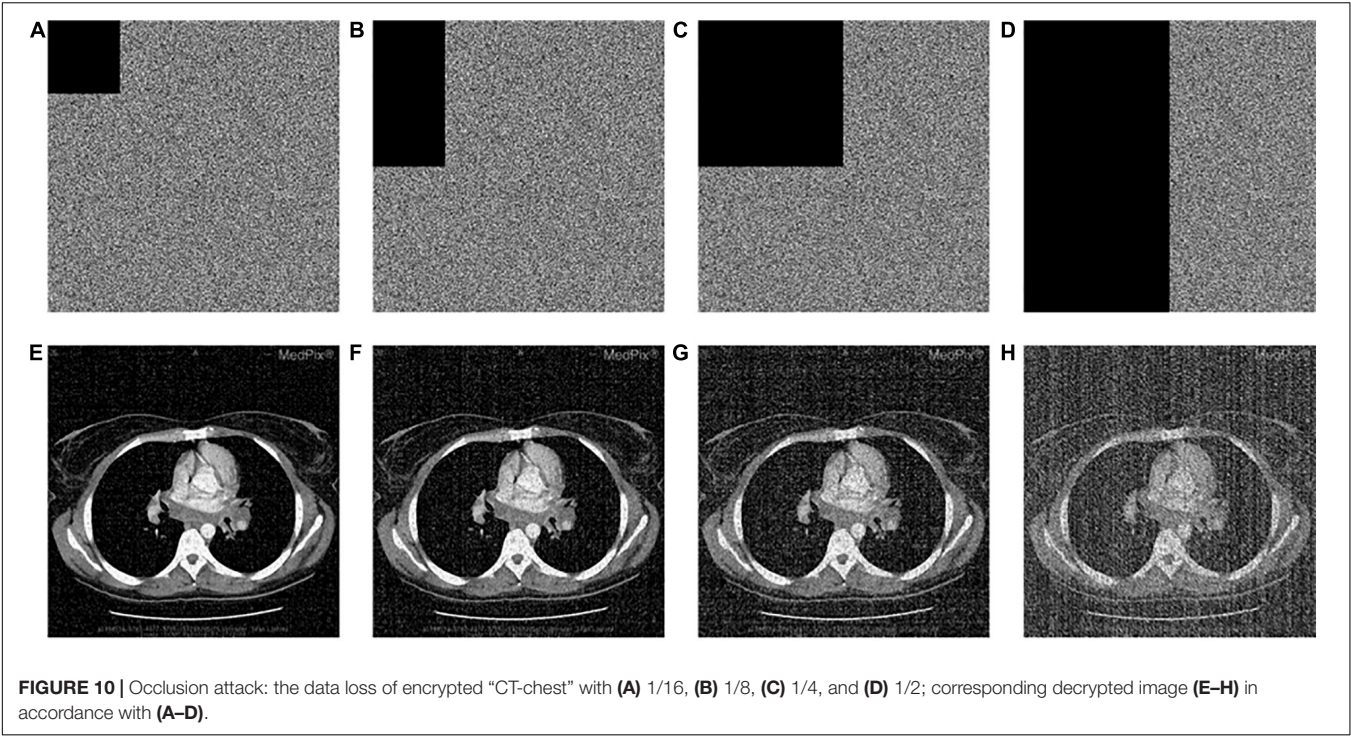


TABLE 15 | PSNR (db) between the original and decrypted images under occlusion.

Algorithm	Occlusion			
	1/16	1/8	1/4	1/2
proposed	18.7275	15.6445	12.3947	9.1317
Belazi et al., 2019	26.6301	17.6447	14.6193	11.6147
Zhou et al., 2015	12.0881	10.0969	8.8968	8.5539
Hua and Zhou, 2017	8.5675	8.5540	8.5502	8.5480
Hua et al., 2018	17.7218	14.8610	12.1275	9.7698

hamming distances. Because the calculation of the SHA-256 function and the hamming distances are closely related to the plaintext, the key is very sensitive to the plaintext. In other words, a small change in the plaintext image produces a completely different key, as detailed in section “Key Sensitivity Evaluation.” In the DNA encoding, this article uses the DNA dynamic encoding by binary bit, compared with the traditional fixed DNA coding and other existing DNA dynamic encoding methods, the base distribution is more uniform, which can be found in **Table 2**. Additionally, for pure white or pure black images, encoding with DNA fixed can cause multiple base repeats, as shown in **Figure 11**. Clearly, **Figure 11A** gives an attacker an opportunity, but the DNA bases in **Figure 11B** are irregular. From this whole encryption system, both “pure white” and “pure black” images of the encrypted images and the corresponding histograms are derived, which are shown in **Figure 12**. Moreover, **Figures 12A,C** are evaluated in **Table 16**. The histogram in **Figure 12** is evenly distributed. The information entropy in **Table 16** is 7.9994. Their NPCR and UACI are both higher than 99.6% and 33.4%. The correlation coefficients are close to 0. It can be shown that it is difficult for an attacker to analyze the equivalent key by choosing pure white or black images. To sum up, the proposed algorithm is robust in defending against the chosen plaintext attack.

Randomness Detection

Randomness detection examines whether the detected sequence demonstrates the characteristics of the random sequence, using the techniques of probability statistics. The most authoritative package for the randomness test is the Special Publication 800-22, provided by the National Institute of Standards and Technology (NIST) of the United States (Khawaja and Khan, 2019). This test package uses the P-value returned for different aspects of the evaluation process for making the judgment. Only when each P-value is greater than 0.01, the test sequence is recognized as a random sequence. In this study, the randomness of the encrypted image of “MRI-Brain” is examined here as an example, and the results are shown in **Table 17**. The results of all the test items in **Table 17** are “success,” which proves that the

TABLE 17 | NIST randomness test of encrypted images.

Test	P-values	Results
Frequency	0.139830	Success
Block frequency	0.747300	Success
Rank	0.944274	Success
Run ($M = 10,000$)	0.240022	Success
long runs of ones	0.937168	Success
Linear complexity	0.618749	Success
Overlapping templates	0.446549	Success
Non-overlapping templates	all P-value > 0.01	Success
FFT	0.967619	Success
Approximate entropy	0.801709	Success
Universal	0.507906	Success
Serial P values 1	0.275633	Success
Serial P values 2	0.295743	Success
Cumulative sums forward	0.168961	Success
Cumulative sums reverse	0.075333	Success
Random excursions	all P-value > 0.01	Success
Random excursions variant	all P-value > 0.01	Success

TABLE 18 | Comparison of efficiency.

Algorithm	Complexity
Proposed	$O(41MN + 5M + 20N)$
Hua and Zhou, 2017	$O(108MN + 72L4)$
Sun, 2018	$O(579MN)$
Belazi et al., 2019	$O(124MN)$

encrypted image obtained by using the proposed algorithm has good randomness.

Efficiency of the Proposed Algorithm

The efficiency of the algorithm is determined by the time expense of the algorithm. The time cost of the proposed algorithm is $O(41MN + 5M + 20N)$. **Table 18** lists the results comparison with other algorithms. From **Table 18**, it is concluded that the encryption efficiency of the proposed algorithm is higher than other ones in the literature.

CONCLUSION

For medical images with large storage space and high pixel redundancy, the encryption effect, security and efficiency of encryption algorithm should have higher standards. The proposed algorithm combines the SHA-256 and the hamming distances to obtain the keys, uses the excellent FHCOC system to realize the best DNA dynamic coding, to generate the DNA dynamic chains of different lengths, to carry out dynamic deletion

TABLE 16 | The performance of the encrypted “pure white” image and the encrypted “pure black” image.

Encrypted image	Entropy	variance	Horizontal	Vertical	Diagonal	NPCR	UACI
White	7.9994	816.5000	0.0018	−0.0014	0.0016	99.6078	33.4375
Black	7.9994	880.3828	0.0029	−0.0013	−0.0037	99.6136	33.5343

operation and dynamic transposition operation of DNA chains. Test results show that the full diffusion of bases causes the pixels of medical images to be completely disorganized; the efficiency is higher and can resist all common attacks. Of course, the proposed algorithm is not only suitable for medical image encryption, but also suitable for other image encryption scenarios. For future research, the proposed algorithm can be applied to large storage space and parallelism of DNA computing for the protection of medical images.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

REFERENCES

- Adleman, L. M. (1994). Molecular computation of solutions to combinatorial problems. *Science* 266, 1021–1024. doi: 10.1126/science.7973651
- Akhavan, A., Samsudin, A., and Akhshani, A. (2017). Cryptanalysis of an image encryption algorithm based on DNA encoding. *Optics Laser Technol.* 95, 94–99. doi: 10.1016/j.optlastec.2017.04.022
- Azimi, Z., and Ahadpour, S. (2020). Color image encryption based on DNA encoding, and pair coupled chaotic maps. *Multimedia Tools Appl.* 79, 1727–1744. doi: 10.1007/s11042-019-08375-6
- Belazi, A., Hermassi, H., Rhouma, R., and Belghith, S. (2014). Algebraic analysis of a RGB image encryption algorithm based on DNA encoding, and chaotic map. *Nonlin. Dyn.* 76, 1989–2004. doi: 10.1007/s11071-014-1263-y
- Belazi, A., Talha, M., Kharbech, S., and Xiang, W. (2019). Novel medical image encryption scheme based on chaos, and DNA encoding. *IEEE Access.* 7, 36667–36681. doi: 10.1109/ACCESS.2019.2906292
- Chai, X. L., Gan, Z. H., Yang, K., Chen, Y. R., and Liu, X. X. (2017). An image encryption algorithm based on the memristive hyperchaotic system, cellular automata, and DNA sequence operation. *Signal Process.* 52, 6–19. doi: 10.1016/j.image.2016.12.007
- Chai, X. L., Gan, Z. H., Yuan, K., Chen, Y. R., and Liu, X. X. (2019). A novel image encryption scheme based on DNA sequence operations, and chaotic systems. *Neural Comput. Appl.* 31, 219–237. doi: 10.1007/s00521-017-2993-9
- Dagadu, J. C., Li, J. P., and Aboagye, E. O. (2019a). Medical image encryption based on hybrid chaotic DNA diffusion. *Wireless Pers. Commun.* 108, 591–612. doi: 10.1007/s11277-019-06420-z
- Dagadu, J. C., Li, J. P., Aboagye, E. O., and Deynu, F. K. (2019b). Medical image encryption scheme based on multiple chaos, and DNA coding. *Int. J. Netw. Secur.* 21, 83–90.
- Donato, C., and Giuseppe, G. (2008). Bifurcation, and chaos in the fractional-order Chen system via a time-domain approach. *Int. J. Bifurc. Chaos* 18, 1845–1863. doi: 10.1142/S0218127408021415
- Dou, Y. Q., Liu, X. M., Fan, H. J., and Li, M. (2017). Cryptanalysis of a DNA, and chaos based image encryption algorithm. *Optik* 145, 456–464. doi: 10.1016/j.ijleo.2017.08.050
- Hermassi, H., Belazi, A., and Rhouma, R. (2014). Security analysis of an image encryption algorithm based on a DNA addition combining with chaotic maps. *Multimedia Tools Appl.* 72, 2211–2224. doi: 10.1007/s11042-013-1533-6
- Hossein, N., Rasul, E., Homayun, M., Frederico, G. G., and Vitor, N. C. (2018). Medical image encryption using a hybrid model of modified genetic algorithm, and coupled map lattices. *Optics Lasers Eng.* 110, 24–32. doi: 10.1016/j.optlaseng.2018.05.009
- Hossein, N., Rasul, E., Mehdi, Y., Malrey, L., and Gisung, J. (2020). Binary search tree image encryption with DNA. *Optik* 202:163505. doi: 10.1016/j.ijleo.2019.163505

AUTHOR CONTRIBUTIONS

XLX and CJZ: conceptualization. XLX: methodology. HYJ and DSZ: formal analysis. XLX: investigation and writing – original draft preparation. HYJ and CJZ: writing – review and editing. DSZ and CJZ: funding acquisition. All authors have read and agreed to the published version of the manuscript.

FUNDING

This work was supported in part by the National Natural Science Foundation of China under the grant number 61672121, in part by the program for the Liaoning Distinguished Professor, the Science and Technology Innovation Fund of Dalian (No. 2018J12GX036).

- Hua, Z., Yi, S., and Zhou, Y. (2018). Medical image encryption using high-speed scrambling, and pixel adaptive diffusion. *Signal. Process.* 144, 134–144. doi: 10.1016/j.sigpro.2017.10.004
- Hua, Z., and Zhou, Y. (2017). Design of image cipher using block-based scrambling, and image filtering. *Inf. Sci.* 396, 97–113. doi: 10.1016/j.ins.2017.02.036
- Kalpana, J., and Murali, P. (2015). An improved color image encryption based on multiple DNA sequence operations with DNA synthetic image, and chaos. *Optik* 126, 5703–5709. doi: 10.1016/j.ijleo.2015.09.091
- Khawaja, M. A., and Khan, M. (2019). Application based construction, and optimization of substitution boxes over 2D mixed chaotic maps. *Int. J. Theor. Phys.* 58, 3091–3117. doi: 10.1007/s10773-019-04188-3
- Kumar, M., Iqbal, A., and Kumar, P. (2016). A new RGB image encryption algorithm based on DNA encoding, and elliptic curve Diffie-Hellman cryptography. *Signal Process.* 125, 187–202. doi: 10.1016/j.sigpro.2016.01.017
- Li, T. Y., Yang, M. G., Wu, J., Jing, X., and Elsaid, A. (2017). A novel image encryption algorithm based on a fractional-order hyperchaotic system, and DNA computing. *Complexity* 2017, 1–13. doi: 10.1155/2017/9010251
- Li, X., Wang, B., Lv, H., Yin, Q., Zhang, Q., and Wei, X. P. (2020). Constraining DNA sequences with a triplet-bases unpaired. *IEEE Trans. Nanobiosci.* 19, 299–307. doi: 10.1109/TNB.2020.2971644
- Liu, C. J., Liu, Y., Zhu, E. Q., and Zhang, Q. (2020). Cross-inhibitor: a time-sensitive molecular circuit based on DNA strand displacement. *Nucleic Acids Res.* 48, 10691–10701. doi: 10.1093/nar/gkaa835
- Liu, H., Zhao, B., and Huang, L. Q. (2019). A remote-sensing image encryption scheme using DNA bases probability, and two-dimensional logistic map. *IEEE Access.* 7, 65450–65459. doi: 10.1109/ACCESS.2019.2917498
- Liu, L. L., Zhang, Q., and Wei, X. P. (2012). A RGB image encryption algorithm based on DNA encoding, and chaos map. *Comput. Electr. Eng.* 38, 1240–1248. doi: 10.1016/j.compeleceng.2012.02.007
- Liu, Y., Wang, J., Fan, J. H., and Gong, L. H. (2016). Image encryption algorithm based on chaotic system, and dynamic S-boxes composed of DNA sequences. *Multimedia Tools Appl.* 75, 4363–4382. doi: 10.1007/s11042-015-2479-7
- Liu, Y. S., Tang, J., and Xie, T. (2014). Cryptanalyzing a RGB image encryption algorithm based on DNA encoding, and chaos map. *Optics Laser Technol.* 60, 111–115. doi: 10.1016/j.optlastec.2014.01.015
- Mondal, B., and Mandal, T. (2017). A light weight secure image encryption scheme based on chaos, and DNA computing. *J. King Saud Univ. Comput. Inform. Sci.* 29, 499–504. doi: 10.1016/j.jksuci.2016.02.003
- Priyanka, and Maheshkar, S. (2017). Region-based hybrid medical image watermarking for secure telemedicine applications. *Multimedia Tools Appl.* 76:36173647. doi: 10.1007/s11042-016-3913-1
- Rehman, A., Liao, X. F., Hahsmi, M. A., and Haider, R. (2018). An efficient mixed inter-intra pixels substitution at 2bits-level for image encryption technique using DNA, and chaos. *Optik* 53, 117–134. doi: 10.1016/j.ijleo.2017.09.099

- Sun, S. (2018). A novel hyperchaotic image encryption scheme based on DNA encoding, pixel-level scrambling, and bit-level scrambling. *IEEE Photon J.* 10, 1–14. doi: 10.1109/JPHOT.2018.2817550
- Wang, B., Xie, Y. J., Zhou, S. H., Zheng, X. D., and Zhou, C. J. (2018). Correcting errors in image encryption based on DNA coding. *Molecules* 23:1878. doi: 10.3390/molecules23081878
- Wang, B., Zhang, Q., and Wei, X. P. (2020). Tabu variable neighborhood search for designing DNA barcodes. *IEEE Trans. NanoBiosci.* 19, 127–131. doi: 10.1109/TNB.2019.2942036
- Wang, X. Y., Wang, Y., Zhu, X. Q., and Luo, C. (2020). A novel chaotic algorithm for image encryption utilizing one-time pad based on pixel level, and DNA level. *Optics Lasers Eng.* 125, 105851. doi: 10.1016/j.optlaseng.2019.105851
- Wang, Y., Lei, P., Yang, H. Q., and Cao, H. Y. (2015). Security analysis on a color image encryption based on DNA encoding, and chaos map. *Comput. Electr. Eng.* 46, 433–446. doi: 10.1016/j.compeleceng.2015.03.011
- Wu, T. Y., Fan, X. N., Wang, K. H., Lai, C. F., Xiong, N., and Wu, J. M.-T. (2019). A DNA computation based image encryption scheme for cloud CCTV systems. *IEEE Access* 181434–181443. doi: 10.1109/ACCESS.2019.2946890
- Wu, Y., Zhou, Y. C., Saveriades, G., Agaian, S., Noonana, J. P., and Natarajan, P. (2013). Local Shannon entropy measure with statistical tests for image randomness. *Inform. Sci.* 222, 323–342. doi: 10.1016/j.ins.2012.07.049
- Xue, X. L., Zhang, Q., Wei, X. P., Guo, L., and Wang, Q. (2010a). A digital image encryption algorithm based on DNA sequence, and multi-chaotic maps. *Neural Netw. World* 20, 285–296.
- Xue, X. L., Zhang, Q., Wei, X. P., Guo, L., and Wang, Q. (2010b). An image fusion encryption algorithm based on DNA sequence, and multi-chaotic maps. *J. Comput. Theor. Nanosci.* 7, 397–403. doi: 10.1166/jctn.2010.1372
- Yang, Y. G., Guan, B. W., Li, J., Li, D., Zhou, Y. H., and Shi, W. M. (2019). Image compression-encryption scheme based on fractional order hyperchaotics systems combined with 2D compressed sensing, and DNA encoding. *Optics Laser Technol.* 119:105661. doi: 10.1016/j.optlastec.2019.105661
- Zhang, J., Hou, D. Z., Ren, H. G., and Islam, N. (2016). Image encryption algorithm based on dynamic DNA coding, and Chen's hyperchaotic system. *Math. Probl. Eng.* 126, 1–11. doi: 10.1155/2016/6408741
- Zhang, L. M., Sun, K. H., Liu, W. H., and He, S. B. (2017). A novel color image encryption scheme using fractional-order hyperchaotic system, and DNA sequence operations. *Chin. Phys. B* 26:100504. doi: 10.1145/3127404
- Zhang, Q., Guo, L., and Wei, X. P. (2010). Image encryption using DNA addition combining with chaotic maps. *Math. Comput. Model.* 52, 2028–2035. doi: 10.1016/j.mcm.2010.06.005
- Zhang, Q., and Wei, X. P. (2013). RGB color image encryption method based on Lorenz Chaotic system, and DNA computation. *IETE Tech. Rev.* 30, 404–409. doi: 10.4103/0256-4602.123123
- Zhang, Q., Xue, X. L., and Wei, X. P. (2012). A novel image encryption algorithm based on DNA chain operation. *Sci. World J.* 2012:286741. doi: 10.1100/2012/286741
- Zhang, S., and Gao, T. G. (2016). An image encryption scheme based on DNA coding, and permutation of hyper-image. *Multimedia Tools Appl.* 75, 17157–14170. doi: 10.1007/s11042-015-2982-x
- Zhang, X. C., Han, F., and Niu, Y. (2017). Chaotic image encryption algorithm based on bit Permutation, and dynamic DNA encoding. *Hindawi Comput. Intell. Neurosci.* 2017:6919675. doi: 10.1155/2017/6919675
- Zhang, Y. Q., Wang, X. Y., Liu, J., and Chi, Z. L. (2016). An image encryption scheme based on the MLNCML system using DNA sequences. *Optics Lasers Eng.* 82, 95–103. doi: 10.1016/j.optlaseng.2016.02.002
- Zhen, P., Zhao, G., Min, L. Q., and Jin, X. (2016). Chaos-based image encryption scheme combining DNA coding, and entropy. *Multimedia Tools Appl.* 75, 1–17. doi: 10.1007/s11042-015-2573-x
- Zhou, S. H., Wang, B., Zheng, X. D., and Zhou, C. J. (2016). An image encryption scheme based on DNA computing, and cellular automata. *Discr. Dyn. Nat. Soc.* 2016:5408529. doi: 10.1155/2016/5408529
- Zhou, Y., Hua, Z., Pun, C. M., and Chen, C. L. P. (2015). Cascade chaotic system with applications. *IEEE Trans. Cybern.* 45, 2001–2012. doi: 10.1109/TCYB.2014.2363168
- Zhu, E. Q., Chen, C. Z., Rao, Y. S., and Xiong, W. C. (2020). Biochemical logic circuits based on DNA combinatorial displacement. *IEEE Access* 8, 34096–34103. doi: 10.1109/ACCESS.2020.2974024
- Zhu, S. Q., Li, J. Q., and Wang, W. H. (2017). Security analysis of improved image encryption method based on DNA coding, and chaotic map. *Appl. Res. Comput.* 34, 3090–3093.
- Zhu, W., Yang, G., Chen, L., and Chen, Z. Y. (2014). An improved image encryption algorithm based on double random phase encoding, and chaos. *Acta Opt. Sin.* 34:0607001. doi: 10.3788/AOS201434.0607001

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2021 Xue, Jin, Zhou and Zhou. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.



Multi-Omics Data Fusion via a Joint Kernel Learning Model for Cancer Subtype Discovery and Essential Gene Identification

Jie Feng¹, Limin Jiang^{1*}, Shuhao Li¹, Jijun Tang^{1,2,3} and Lan Wen^{4*}

¹ School of Computer Science and Technology, College of Intelligence and Computing, Tianjin University, Tianjin, China,

² School of Computational Science and Engineering, University of South Carolina, Columbia, SC, United States, ³ Key Laboratory of Systems Bioengineering (Ministry of Education), Tianjin University, Tianjin, China, ⁴ Changsha Municipal Center of Disease Control, Changsha, China

OPEN ACCESS

Edited by:

Xiangxiang Zeng,
Hunan University, China

Reviewed by:

Qi Zhao,
University of Science and Technology
Liaoning, China
Yang Yang,
Shanghai Jiao Tong University, China

*Correspondence:

Limin Jiang
jianglm@tju.edu.cn
Lan Wen
87903499@qq.com

Specialty section:

This article was submitted to
Computational Genomics,
a section of the journal
Frontiers in Genetics

Received: 29 December 2020

Accepted: 02 February 2021

Published: 04 March 2021

Citation:

Feng J, Jiang L, Li S, Tang J and
Wen L (2021) Multi-Omics Data
Fusion via a Joint Kernel Learning
Model for Cancer Subtype Discovery
and Essential Gene Identification.
Front. Genet. 12:647141.
doi: 10.3389/fgene.2021.647141

The multiple sources of cancer determine its multiple causes, and the same cancer can be composed of many different subtypes. Identification of cancer subtypes is a key part of personalized cancer treatment and provides an important reference for clinical diagnosis and treatment. Some studies have shown that there are significant differences in the genetic and epigenetic profiles among different cancer subtypes during carcinogenesis and development. In this study, we first collect seven cancer datasets from the Broad Institute GDAC Firehose, including gene expression profile, isoform expression profile, DNA methylation expression data, and survival information correspondingly. Furthermore, we employ kernel principal component analysis (PCA) to extract features for each expression profile, convert them into three similarity kernel matrices by Gaussian kernel function, and then fuse these matrices as a global kernel matrix. Finally, we apply it to spectral clustering algorithm to get the clustering results of different cancer subtypes. In the experimental results, besides using the *P*-value from the Cox regression model and survival analysis as the primary evaluation measures, we also introduce statistical indicators such as Rand index (RI) and adjusted RI (ARI) to verify the performance of clustering. Then combining with gene expression profile, we obtain the differential expression of genes among different subtypes by gene set enrichment analysis. For lung cancer, GMPS, EPHA10, C10orf54, and MAGEA6 are highly expressed in different subtypes; for liver cancer, CMYA5, DEPDC6, FAU, VPS24, RCBTB2, LOC100133469, and SLC35B4 are significantly expressed in different subtypes.

Keywords: cancer subtype, kernel PCA, spectral clustering, survival analysis, GSEA

INTRODUCTION

Cancer is the important leading cause of death in the world and is responsible for an estimated 9.6 million deaths in 2018. Unlike most other diseases, cancer is not a sort of single disease but is a group of diseases involving abnormal cell growth with the potential to invade or spread to other parts of the body. In the same type of cancer, patients usually have the same or similar external

appearances, but in most cases, universal drugs and universal treatment methods do not produce good prognosis in all cases. The multiple sources of cancer determine its multiple causes, and the same cancer can be composed of many different subtypes. The discovery and identification of cancer subtypes are a key part of personalized cancer treatment and provide an important reference for clinical diagnosis and treatment (de Kruijf et al., 2013).

The Cancer Genome Atlas (TCGA) is the largest open cancer genome database to date initiated by the US government, which aims to catalog and discover major cancer-causing genome alterations in large cohorts of over 30 human tumors through large-scale genome sequencing and integrated multidimensional analyses. It covers a variety of omics expression data including genomics, transcriptomics, copy number variation, DNA methylation, proteomics, and clinical information of follow-up cases (Tomczak et al., 2015; Jiang et al., 2019a), which provide great support for the detection of cancer subtypes by computational methods.

Recently, many methods for cancer subtypes recognition and marker extraction have been proposed (Yeoh et al., 2002; Lapointe et al., 2004; Figueroa et al., 2010; Yang et al., 2017a; Pan et al., 2019). Some models are based on single expression data, including gene expression (Yeoh et al., 2002; Lapointe et al., 2004), microRNA (miRNA) expression (Yang et al., 2017a,b; Liu and Yang, 2018), copy number variation (Pan et al., 2019), and DNA methylation (Figueroa et al., 2010). Lapointe et al. (2004) identified three subclasses of prostate tumors based on distinct patterns of gene expression. Yang et al. (2017a) clustered miRNAs based on Fisher linear discriminant analysis (FDA), using representative cluster member combinations as potential biomarkers. Pan et al. (2019) used copy number variation, a biomarker more likely to be used for cancer diagnosis than mRNA biomarkers, to further reveal differences between various breast cancer subtypes. Figueroa et al. (2010) examined the methylation profiles of 344 patients with acute myeloid leukemia (AML). Clustering of these patients by methylation data segregated patients into 16 groups. Five of these groups defined new AML subtypes. Also, there are methods to analyze and predict cancer subtypes by considering multiple expression data (Shen et al., 2009; Wang et al., 2014; Ge et al., 2016; Jiang et al., 2019b). The iCluster is a latent variable model-based clustering algorithm proposed by Shen et al. (2009). It uses multiple sources of data for integrated analysis to identify tumor subtypes. Similarity network fusion (SNF) is a network fusion method integrating multicomponent data, which was proposed by Wang et al. (2014). SNF builds a similar network of sample pairs on different histological data (gene, methylation, and miRNA) and then integrates the network to predict cancer subtypes.

Furthermore, due to the high dimensionality of research data, we need to find effective and suitable dimensionality reduction methods. Some methods, such as principal component analysis (PCA) and non-negative matrix factorization (NMF), have been used to combine clustering algorithms (Alter et al., 2000; Holter et al., 2000; Brunet et al., 2004). However, for the high-dimensional and non-linear gene data,

the performance is not always good. In order to better handle these, we consider a non-linear version of PCA, kernel PCA (Schölkopf et al., 1998), which introduces a non-linear mapping function that can map data in the raw space to high-dimensional space. It can make the distribution of all mapped data linearized and simplified in high-dimensional space, and then PCA can be used to construct features.

In this study, inspired by SNF, we combine gene expression profile with isoform expression profile and DNA methylation expression data. We propose a novel method: first, take kernel PCA to extract features for each profile, then convert them into three similarity kernel matrices, and fuse them into one. Finally, we apply it to spectral clustering algorithm to get the clustering results of different cancer subtypes.

MATERIALS AND METHODS

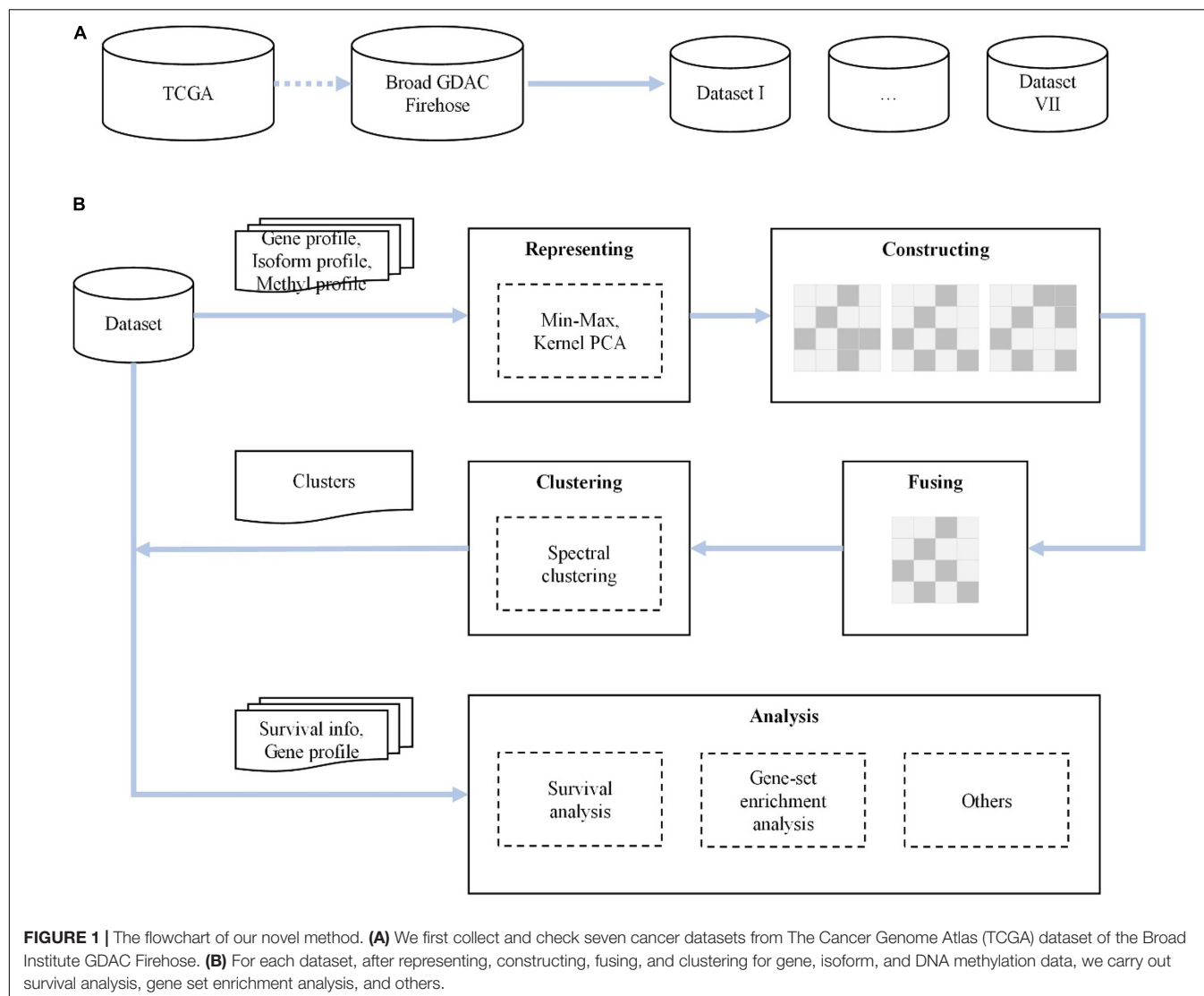
We propose a novel method for analyzing various cancer subtypes. First, we rescale the raw expression data by min-max normalization and reduce the dimensionality of data via kernel PCA, with a minimal loss of information. Then, in each cancer dataset, based on the expression of gene profile, isoform profile, and methylation data, we construct three similarity kernel matrices through the Gaussian kernel function and fuse them into a global similarity expression matrix. Finally, the integrated similarity kernel matrix is fed to spectral clustering, and the predictive clusters are identified. The flowchart of our proposed method is shown in **Figure 1**.

Data Sources

In our study, all research data are collected from the Broad Institute GDAC Firehose¹ (Center BITGDA, 2016). Firehose is an analytical infrastructure created at the Broad Institute based on the data of TCGA project (Tomczak et al., 2015), which provides genome-scale transcriptome data for various cancers and different levels of processed data for cancer analysis. Firehose gives a corresponding visual web platform, Firebrowse², which can easily access TCGA open access layer data. This greatly lowers the threshold for experimenters to operate the TCGA database and also makes the data for analyzing as consistent as possible. Here, we extract seven common cancer datasets: BLCA, BRCA, COAD, KIDNEY (KICH&KIRC&KIRP), LIHC, LUNG (LUAD&LUSC), and STAD. For each cancer dataset, it consists of gene expression information (gdac_rnaseqv2_genes_RSEM_normalized_Level_3, 2016-02-18), isoform expression information (gdac_rnaseqv2_isoforms_RSEM_normalized_Level_3, 2016-02-18), DNA methylation expression information (gdac_Methylation_Preprocess_mean_Level_3, 2016-02-18), and corresponding clinical information (gdac_Clinical_Pick_Tier1_Level_4, 2016-02-18). The clinical data are used in subsequent survival analyses, while the other three expression profiles are used to construct a suitable

¹<https://gdac.broadinstitute.org>

²<http://firebrowse.org/>



global similarity kernel matrix. We check redundant cases (reserve cases with number 01–09) in the four profiles at each cancer datasets and extract all valid cases that contain the above expression information. And then, we obtain the experimental input data for cluster analysis. A summary of our datasets is shown in **Table 1**.

TABLE 1 | Description of seven datasets.

Datasets	No. samples	Gene	Isoform	Methylation
BRCA	780	20,531	73,599	20,106
COAD	275	20,531	73,599	20,116
KIDNEY	658	20,531	73,599	20,119
LUNG	824	20,531	73,599	20,116
STAD	372	20,531	73,599	20,101
BLCA	408	20,531	73,599	20,109
LIHC	371	20,531	73,599	20,105

Data Representation

Min–Max Normalization

Since we collect high-quality, standardized datasets directly from Firehose, the size of the data values can intuitively reflect the expression abundance of gene, isoform, or methylation. However, to eliminate the influence of digital distribution in three expression profiles and make them fused reasonable, we use min–max normalization to rescale the values. The general formula for a min–max of [0, 1] is defined as Eq. 1:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (1)$$

where x is an original expression value and x' is the rescaled value.

Kernel Principal Component Analysis

The kernel PCA is a method for performing a non-linear form of PCA proposed in Schölkopf et al. (1998). Through using kernel PCA, the dimensionality of complex, non-linear features can be

reduced effectively. Kernel PCA transforms the raw linear input space R into a high-dimensional feature space F by using some non-linear mapping, like a dot product matrix defined as Eq. 2:

$$K(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle \quad (2)$$

and calculates the principal components in F . Then compute projections onto the eigenvectors obtained by diagonalizing K to extract the principal components corresponding to the k of K (Schölkopf et al., 1998; Devi et al., 2014). In this study, we use the kernel PCA method and take polynomial kernel defined as Eq. 3:

$$K_{poly}(x_i, x_j) = (x_i^T x_j + 1)^3 \quad (3)$$

as the non-linear mapping. We adopt the default parameter, x_i and x_j are the expression vector of i -th case and j -th, and all the non-zero components are preserved. After performing the above rescale and reduction on the gene, isoform, and methylation expression profiles in the dataset, we gain the necessary input to construct a similarity kernel matrix.

Similarity Kernel Matrix Kernel Construction

The kernel methods map data points into possibly high-dimensional feature space, where the distribution of all mapped data is linearized and simplified (Vert et al., 2004; Mei and Fei, 2010). Assume mapping function $\Phi(x)$, the computation of the inner product $\langle \Phi(x_i), \Phi(x_j) \rangle$ in the high-dimensional feature space F can be implemented in the original space R using kernel trick, $K(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$, such that no explicit mapping function or even explicit feature representation is required. The size of the matrix used to represent the profile of N cancer cases is always N by N . This allows us to comprehensively consider the expression of three profiles for one specific cancer, which perform a more accurate cluster analysis (Vert et al., 2004). Here, we use the Gaussian kernel and the adjusted parameter γ , as Eq. 4:

$$K_{gaussian}(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \quad (4)$$

Kernel Fusion

Data fusion from multidimensional expression profiles has been shown to produce better results than considering a single expression information. Jiang et al. (2019b) and Li et al. (2020) have researched multi-omics data fusion and achieved good results. We fuse three different expression profiles (gene, isoform, and methylation) to construct a global similarity kernel matrix for each cancer. Therefore, we integrate three Gaussian kernel matrices. In our example, we adopt an average fusion strategy as Eq. 5:

$$K_{fuse} = \frac{1}{3}(K_{gene}, K_{isoform}, K_{methyl}) \quad (5)$$

where K_{gene} , $K_{isoform}$, and K_{methyl} represent the similarity kernel matrix constructed by gene, isoform, and methylation expression profiles, respectively.

TABLE 2 | Results with or without kernel principal component analysis (KPCA).

Datasets	P-value without KPCA	P-value with KPCA
BRCA(4)	0.137	9.71e-06
COAD(10)	0.465	2.39e-03
KIDNEY(9)	0.381	7.15e-04
LUNG(4)	0.386	9.06e-03
STAD(6)	0.977	2.35e-03
BLCA(9)	0.290	1.88e-05
LIHC(7)	0.038	1.23e-07

TABLE 3 | P-values of taking single kernel and fused kernel.

Datasets	Gene	Isoform	Methyl	Fusion
BRCA(4)	0.011	0.342	0.755	9.71e-06
COAD(10)	0.024	0.693	0.254	2.39e-03
KIDNEY(9)	0.057	0.475	0.116	7.15e-04
LUNG(4)	0.667	0.565	0.142	9.06e-03
STAD(6)	0.063	0.552	0.252	2.35e-03
BLCA(9)	0.090	0.116	0.192	1.88e-05
LIHC(7)	0.001	0.004	0.071	1.23e-07

TABLE 4 | Performance between Li's and our method.

Datasets	Li's method	Our method
BRCA(4)	1.12e-05	9.71e-06
COAD(10)	1.12e-07	2.39e-03
KIDNEY(9)	1.80e-02	7.15e-04
LUNG(4)	1.59e-06	9.06e-03
STAD(6)	2.00e-03	2.35e-03
BLCA(9)	–	1.88e-05
LIHC(7)	–	1.23e-07

TABLE 5 | Rand index (RI) and adjusted Rand index (ARI) on two datasets between Li's and our method.

Datasets	RI of Li	Our RI	ARI of Li	Our ARI
KIDNEY(9)	0.59	0.66	0.07	0.21
LUNG(4)	0.5	0.64	0	0.28

Spectral Clustering

Spectral clustering is a clustering method based on graph theory algorithm; the basic idea is to use the similarity matrix of the samples to obtain the feature vector of the feature decomposition for cluster analysis (Von Luxburg, 2007). Because of its excellent algebraic graph foundation, it can get a global loose solution for complex cluster structure (Jia et al., 2014). We use it as the core algorithm for cluster analysis. The process of spectral clustering algorithm is taken as follows. First, based on K_{fuse} , calculate the Laplacian matrix L . Then, construct the normalized Laplacian matrix $D^{-1/2}LD^{-1/2}$. D is a diagonal matrix whose diagonal element is the sum of the row elements of K_{fuse} . And compute the eigen vectors y corresponding to the eigen values of $D^{-1/2}LD^{-1/2}$. The matrices composed of corresponding eigen vectors y are standardized on a row basis to form the $N_{case} \times N_{feature}$

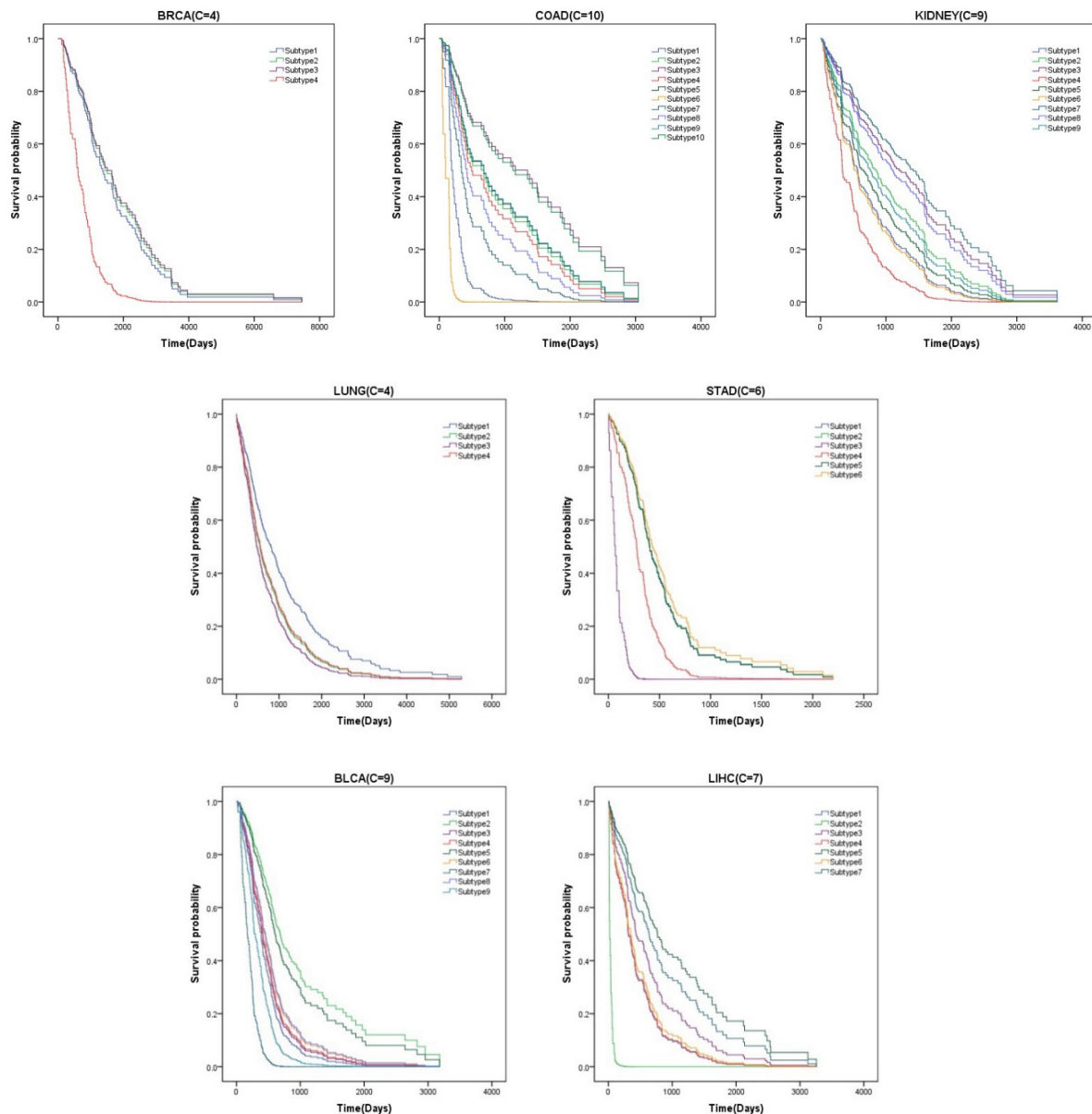


FIGURE 2 | Survival curves of various subtypes for seven cancer datasets.

feature matrix Y . Finally, each row in Y is taken as a sample, which is clustered by discrete method to obtain cluster partition $C(C_1, C_2, \dots, C_j)$. Each partition will represent a cancer subtype. The whole process of spectral clustering can be transformed into solving the optimization problem as Eq. 6:

$$\begin{aligned} \min \operatorname{tr} \left(Y^T D^{-\frac{1}{2}} L D^{-\frac{1}{2}} Y \right) \\ \text{s.t. } Y^T Y = I \end{aligned} \quad (6)$$

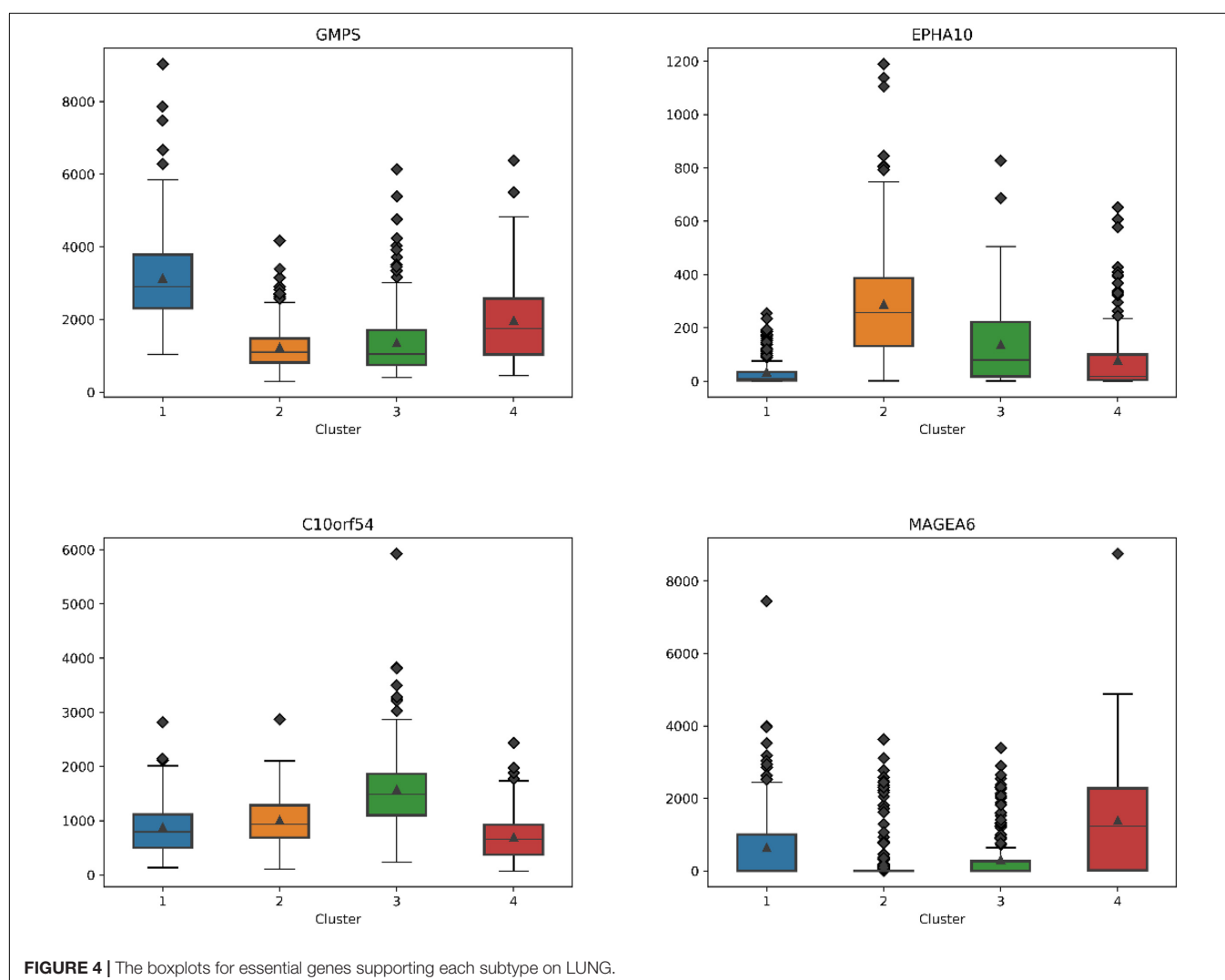
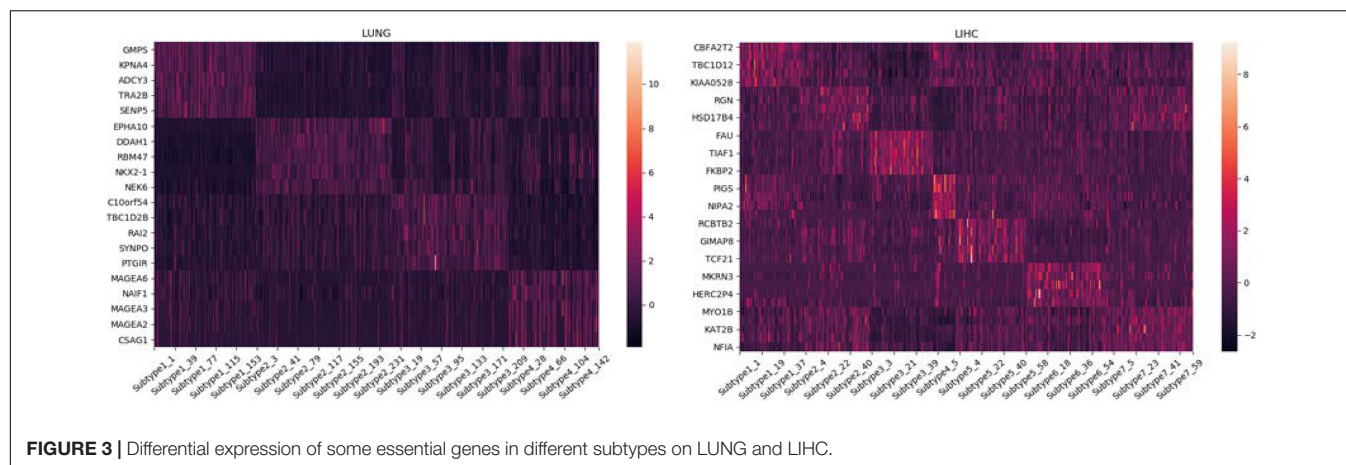
where Y is the eigen matrix for the eigen values of $D^{-1/2} L D^{-1/2}$, D is the degree matrix of K_{fuse} , and L is the Laplacian matrix of K_{fuse} .

RESULTS

In this section, we evaluate and compare the performance of our proposed method in multiple dimensions, using P -values and survival curves as the primary criteria and taking indexes such as RI and ARI into consideration. Finally, through gene set enrichment analysis (GSEA), some key genes supporting each subtype are obtained and displayed using heat maps and boxplots.

Evaluation Novel Method

We use the P -value of Cox regression model to evaluate the performance of several key steps of the proposed



method (Pölsterl et al., 2015, 2016a,b). It includes applying kernel PCA to reduce the original data dimension, using similarity kernel fusion strategy to obtain feature input, and employing spectral clustering as the core clustering method to

obtain the final clustering result. We calculate the P -values for the clusters on the seven datasets. A lower P -value indicates a more significant result. Here, we use 0.05 as the threshold for evaluation.

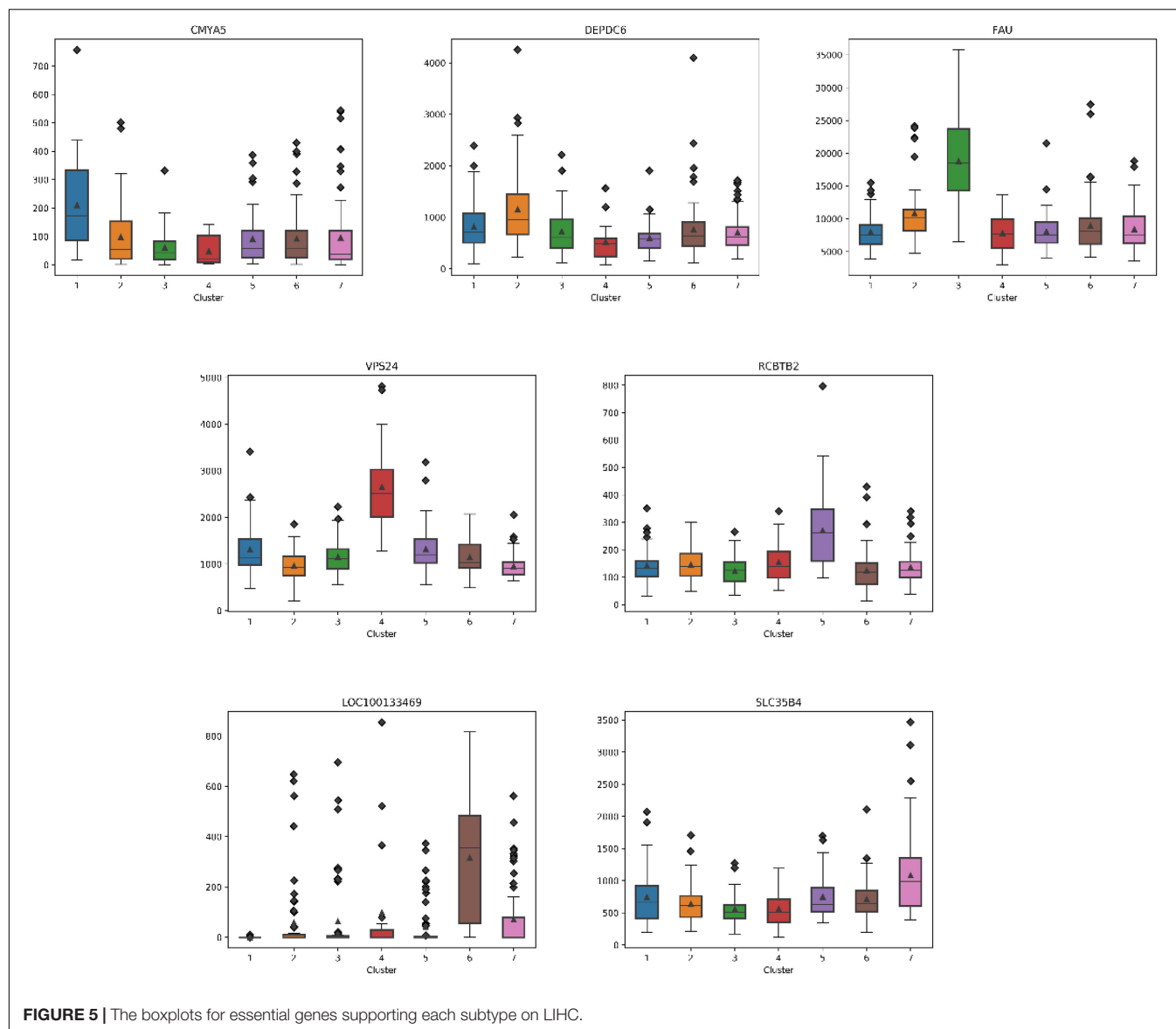


FIGURE 5 | The boxplots for essential genes supporting each subtype on LIHC.

Performance of Kernel Principal Component Analysis

The kernel PCA is a non-linear version of PCA widely used in linear dimensionality reduction methods. Using kernel PCA, the dimensionality of complex, non-linear features can be reduced effectively. For the features with tens of thousands of dimensions in the original data, it can be reduced to only a few hundred. We compare all datasets, with and without kernel PCA, and the resulting *P*-values are shown in **Table 2**. According to the data in **Table 2** (the number of clusters in the table is the optimal results with the lowest *P*-values), kernel PCA greatly improves the performance of the method and makes the results more reliable and stable.

Performance of Fusion Strategy

Here, we compare the performance of using a single kernel directly with the use of a kernel fusion strategy on seven datasets, and the results are shown in **Table 3**. After using similarity kernel

fusion, the *P*-values on seven datasets have been significantly improved. And for the dataset LIHC, although the performance on single kernel has already performed well, kernel fusion will further enhance the clustering results. We therefore conclude that the strategy for similarity kernel fusion is necessary.

Comparing With Other Methods

We compare the results with those of the method of Li et al. (2020). As shown in **Table 4**, we find that using kernel PCA for feature reduction, taking weighted fusion strategies instead of complex SKF, and finally generating clustering results from spectral clustering have comparable reliability and stability.

Clustering Analysis

The Rand index (RI) (Rand, 1971) is an indicator for evaluating clustering performance in statistics, by measuring the similarity between two data clusters. However, a problem with the index

is that the expected value of the RI of two random partitions cannot take a constant value (Steinley, 2004). The adjusted RI proposed by Yeung and Ruzzo (2001), which is the corrected-for-chance version of the RI, can effectively avoid RI's insufficient. We measure both the LUNG and KIDNEY datasets and compare the results obtained by Li et al. (2020). The results are shown in **Table 5**. From the indicators such as RI and ARI, our clustering is more stable, and the effect is much better than Li's in KIDNEY and LUNG. Especially on LUNG, although we get a higher *P*-value, we have a 28% advantage on the ARI.

Survival Analysis

Survival analysis is a branch of statistics that analyzes the expected duration until an event occurs, such as the death of a cancer patient. We find that patients with subtype 2 of liver cancer (LIHC), subtype 6 of colon cancer (COAD), and subtype 3 of stomach cancer (STAD) have higher mortality. More attention should be paid to these patients. We also see that the average survival time of breast cancer patients (BRCA) and lung cancer patients (LUNG) is longer than that of others. It indicates that these cluster results can be used to guide clinical treatment. The survival curves of all datasets are shown in **Figure 2**.

Gene Set Enrichment Analysis

GSEA (Mootha et al., 2003; Subramanian et al., 2005; Huang et al., 2020) is an analysis method for genome-wide expression profile chip data, which compares genes with a predefined set of genes. Synthesize the existing information base of gene location, nature, function, biological significance, etc., to build a molecular tag database (MSigDB), in which known genes are identified by chromosomal location, established gene set, and model sequence. Tumor-related gene set and GO gene set and other functional gene sets are grouped and classified. By analyzing the gene expression profile data, we can understand their expression status in a specific functional gene set and whether this expression status has some statistical significance. In this paper, we use Broad Institute's offline analysis software GSEA_4.0.2, and C4 collection (cancer gene neighborhoods and cancer modules), provided by Broad Institute in the Molecular Signatures Database (MSigDB), which is a computational gene set defined by mining large collections of cancer-oriented microarray data. We analyze the LUNG and LIHC datasets, and we collect the expression data of genes with higher scores on different subtypes. The heat maps drawn are shown in **Figure 3**.

Essential Gene Analysis

For each subtype on the datasets LUNG and LIHC, we select the essential gene that can highly distinguish the subtypes. According to the expression of each gene on its dataset, we obtain the box diagrams as shown in **Figures 4, 5**. We find that GMPS, EPHA10, C10orf54, and MAGEA6 are highly expressed in different subtypes on the dataset LUNG; and CMYA5, DEPDC6, FAU, VPS24, RCBTB2, LOC100133469, and SLC35B4 are significantly expressed in different subtypes on the dataset LIHC, respectively.

CONCLUSION

In this paper, we propose a model for accurately predicting cancer subtypes. First, we collect seven cancer datasets from Firehose website, which contained three kinds of expression data (gene expression, isoform expression, and methylation expression). Then we construct three similar kernels for three kinds of expression data, respectively, and we fuse the three kernels into the global one. Finally, the cancer subtypes are discovered by spectral clustering. We take *P*-value as the overall evaluation criterion, combining with survival curve analysis and GSEA.

In the future, we will also try other machine learning methods or deep learning methods (Kong and Yu, 2018; Ding et al., 2019a,b; Shen et al., 2019; Gao et al., 2020; Lee et al., 2020; Wang et al., 2020), to deal with the problem of small samples and large features of cancer data and predict cancer subtypes more accurately.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/**Supplementary Material**, further inquiries can be directed to the corresponding author/s.

AUTHOR CONTRIBUTIONS

JF and SL conceived and designed the experiments. JF and LJ performed the experiments and analyzed the data. LW and LJ wrote the manuscript. JT supervised the experiments and reviewed the manuscript. All authors contributed to the article and approved the submitted version.

FUNDING

This work was supported by a grant from the National Natural Science Foundation of China (NSFC 61902271 and 61972280) and the National Key RD Program of China (2020YFA0908401, 2020YFA0908400, 2018YFC0910405, and 2017YFC0908400).

ACKNOWLEDGMENTS

We would like to thank the Broad Institute GDAC Firehose for providing extensive amounts of biological data for the community.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/fgene.2021.647141/full#supplementary-material>

REFERENCES

- Alter, O., Brown, P. O., and Botstein, D. (2000). Singular value decomposition for genome-wide expression data processing and modeling. *Proc. Natl. Acad. Sci. U.S.A.* 97, 10101–10106. doi: 10.1073/pnas.97.18.10101
- Brunet, J.-P., Tamayo, P., Golub, T. R., and Mesirov, J. P. (2004). Metagenes and molecular pattern discovery using matrix factorization. *Proc. Natl. Acad. Sci. U.S.A.* 101, 4164–4169. doi: 10.1073/pnas.0308531101
- Center BITGDA (2016). *Analysis-Ready Standardized TCGA Data From Broad GDAC Firehose 2016_01_28 run: Dataset*. Cambridge, MA: Broad Institute of MIT and Harvard, doi: 10.7908/C11G0KM9
- de Kruijf, E. M., Engels, C. C., van de Water, W., Bastiaannet, E., Smit, V. T., van de Velde, C. J., et al. (2013). Tumor immune subtypes distinguish tumor subclasses with clinical implications in breast cancer patients. *Breast Cancer Res. Treat.* 142, 355–364. doi: 10.1007/s10549-013-2752-2
- Devi, H. S., Thounaojam, D. M., and Laishram, R. (2014). An approach to illumination and expression invariant multiple classifier face recognition. *Int. J. Comput. Appl.* 975:8887. doi: 10.5120/15959-5335
- Ding, Y., Tang, J., and Guo, F. (2019a). Identification of drug-side effect association via multiple information integration with centered kernel alignment. *Neurocomputing* 325, 211–224. doi: 10.1016/j.neucom.2018.10.028
- Ding, Y., Tang, J., and Guo, F. (2019b). Identification of drug-target interactions via fuzzy bipartite local model. *Neural Comput. Appl.* 32, 10303–10319. doi: 10.1007/s00521-019-04569-z
- Figuerola, M. E., Lugthart, S., Li, Y., Erpelinck-Verschueren, C., Deng, X., Christos, P. J., et al. (2010). DNA methylation signatures identify biologically distinct subtypes in acute myeloid leukemia. *Cancer cell* 17, 13–27. doi: 10.1016/j.ccr.2009.11.020
- Gao, J., Lyu, T., Xiong, F., Wang, J., Ke, W., and Li, Z. (2020). “MGNN: a multimodal graph neural network for predicting the survival of cancer patients,” in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, (New York, NY: United States Association for Computing), 1697–1700. doi: 10.1145/3397271.3401214
- Ge, S.-G., Xia, J., Sha, W., and Zheng, C.-H. (2016). Cancer subtype discovery based on integrative model of multigenomic data. *IEEE ACM Trans. Comput. Biol. Bioinform.* 14, 1115–1121. doi: 10.1109/TCBB.2016.2621769
- Holter, N. S., Mitra, M., Maritan, A., Cieplak, M., Banavar, J. R., and Fedoroff, N. V. (2000). Fundamental patterns underlying gene expression profiles: simplicity from complexity. *Proc. Natl. Acad. Sci.* 97, 8409–8414. doi: 10.1073/pnas.150242097
- Huang, Y., Yuan, K., Tang, M., Yue, J., Bao, L., Wu, S., et al. (2020). Melatonin inhibiting the survival of human gastric cancer cells under ER stress involving autophagy and Ras-Raf-MAPK signalling. *J. Cell. Mol. Med.* 25, 1480–1492. doi: 10.1111/jcmm.16237
- Jia, H., Ding, S., Xu, X., and Nie, R. (2014). The latest research progress on spectral clustering. *Neural Comput. Appl.* 24, 1477–1486. doi: 10.1007/s00521-013-1439-2
- Jiang, L., Wang, C., Tang, J., and Guo, F. (2019a). LightCpG: a multi-view CpG sites detection on single-cell whole genome sequence data. *BMC Genom.* 20:306. doi: 10.1186/s12864-019-5654-9
- Jiang, L., Xiao, Y., Ding, Y., Tang, J., and Guo, F. (2019b). Discovering cancer subtypes via an accurate fusion strategy on multiple profile data. *Front. Genet.* 10:20. doi: 10.3389/fgene.2019.00020
- Kong, Y., and Yu, T. (2018). A graph-embedded deep feedforward network for disease outcome classification and feature selection using gene expression data. *Bioinformatics* 34, 3727–3737. doi: 10.1093/bioinformatics/bty429
- Lapointe, J., Li, C., Higgins, J. P., Van De Rijn, M., Bair, E., Montgomery, K., et al. (2004). Gene expression profiling identifies clinically relevant subtypes of prostate cancer. *Proc. Natl. Acad. Sci. U.S.A.* 101, 811–816. doi: 10.1073/pnas.0304146101
- Lee, S., Lim, S., Lee, T., Sung, I., and Kim, S. (2020). Cancer subtype classification and modeling by pathway attention and propagation. *Bioinformatics* 36, 3818–3824. doi: 10.1093/bioinformatics/btaa203
- Li, S., Jiang, L., Tang, J., Gao, N., and Guo, F. (2020). Kernel fusion method for detecting cancer subtypes via selecting relevant expression data. *Front. Genet.* 11:979. doi: 10.3389/fgene.2020.00979
- Liu, K., and Yang, Y. (2018). Incorporating link information in feature selection for identifying tumor biomarkers by using miRNA-mRNA paired expression data. *Curr. Proteom.* 15, 165–171. doi: 10.2174/157016461466617031160232
- Mei, S., and Fei, W. (2010). Amino acid classification based spectrum kernel fusion for protein subnuclear localization. *BMC Bioinform.* 11:S17. doi: 10.1186/1471-2105-11-s17
- Mootha, V. K., Lindgren, C. M., Eriksson, K.-F., Subramanian, A., Sihag, S., Lehar, J., et al. (2003). PGC-1 α -responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. *Nat. Genet.* 34, 267–273. doi: 10.1038/ng1180
- Pan, X., Hu, X., Zhang, Y.-H., Chen, L., Zhu, L., Wan, S., et al. (2019). Identification of the copy number variant biomarkers for breast cancer subtypes. *Mol. Genet. Genom.* 294, 95–110. doi: 10.1007/s00438-018-1488-4
- Pölsterl, S., Gupta, P., Wang, L., Conjeti, S., Katouzian, A., and Navab, N. (2016a). Heterogeneous ensembles for predicting survival of metastatic, castrate-resistant prostate cancer patients. *Fl1000Research* 5:2676. doi: 10.12688/fl1000research.8231.3
- Pölsterl, S., Navab, N., and Katouzian, A. (2015). “Fast training of support vector machines for survival analysis,” in *Paper Presented at the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, eds A. Appice, P. Rodrigues, V. Santos Costa, J. Gama, A. Jorge, and C. Soares (Cham: Springer), 243–259. doi: 10.1007/978-3-319-23525-7_15
- Pölsterl, S., Navab, N., and Katouzian, A. (2016b). An efficient training algorithm for kernel survival support vector machines. *arXiv [Preprint]* arXiv:1611.07054.
- Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *J. Am. Statist. Assoc.* 66, 846–850. doi: 10.1080/01621459.1971.10482356
- Schölkopf, B., Smola, A., and Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural comput.* 10, 1299–1319. doi: 10.1162/089976698300017467
- Shen, R., Olshen, A. B., and Ladanyi, M. (2009). Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis. *Bioinformatics* 25, 2906–2912. doi: 10.1093/bioinformatics/btp543
- Shen, Y., Tang, J., and Guo, F. (2019). Identification of protein subcellular localization via integrating evolutionary and physicochemical information into Chou's general PseAAC. *J. Theor. Biol.* 462, 230–239. doi: 10.1016/j.jtbi.2018.11.012
- Steinley, D. (2004). Properties of the hubert-arable adjusted rand index. *Psychol. Methods* 9:386. doi: 10.1037/1082-989X.9.3.386
- Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., et al. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. U.S.A.* 102, 15545–15550. doi: 10.1073/pnas.0506580102
- Tomczak, K., Czerwińska, P., and Wiznerowicz, M. (2015). The cancer genome atlas (TCGA): an immeasurable source of knowledge. *Contemp. Oncol.* 19:A68. doi: 10.5114/wo.2014.47136
- Vert, J.-P., Tsuda, K., and Schölkopf, B. (2004). A primer on kernel methods. *Kernel Methods Comput. Biol.* 47, 35–70. doi: 10.7551/mitpress/4057.003.0004
- Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statist. Comput.* 17, 395–416. doi: 10.1007/s11222-007-9033-z
- Wang, B., Mezlini, A. M., Demir, F., Fiume, M., Tu, Z., Brudno, M., et al. (2014). Similarity network fusion for aggregating data types on a genomic scale. *Nat. Methods* 11:333. doi: 10.1038/nmeth.2810
- Wang, H., Ding, Y., Tang, J., and Guo, F. (2020). Identification of membrane protein types via multivariate information fusion with Hilbert-Schmidt independence criterion. *Neurocomputing* 383, 257–269. doi: 10.1016/j.neucom.2019.11.103
- Yang, Y., Huang, N., Hao, L., and Kong, W. (2017a). A clustering-based approach for efficient identification of microRNA combinatorial biomarkers. *BMC Genom.* 18:210. doi: 10.1186/s12864-017-3498-8
- Yang, Y., Xiao, Y., Cao, T., and Kong, W. (2017b). MiRFFS: a functional group-based feature selection method for the identification of microRNA biomarkers.

- Int. J. Data Mining Bioinform.* 18, 40–55. doi: 10.1504/IJDMB.2017.10007184
- Yeoh, E.-J., Ross, M. E., Shurtleff, S. A., Williams, W. K., Patel, D., Mahfouz, R., et al. (2002). Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. *Cancer Cell* 1, 133–143. doi: 10.1016/S1535-6108(02)00032-6
- Yeung, K. Y., and Ruzzo, W. L. (2001). Details of the adjusted rand index and clustering algorithms, supplement to the paper an empirical study on principal component analysis for clustering gene expression data. *Bioinformatics* 17, 763–774. doi: 10.1093/bioinformatics/17.9.763

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2021 Feng, Jiang, Li, Tang and Wen. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.



iDNA-MT: Identification DNA Modification Sites in Multiple Species by Using Multi-Task Learning Based a Neural Network Tool

Xiao Yang¹, Xiucui Ye², Xuehong Li^{3*} and Lesong Wei^{2*}

¹ School of Software, Shandong University, Jinan, China, ² Department of Computer Science, University of Tsukuba, Tsukuba, Japan, ³ Department of Rehabilitation, Heilongjiang Province Land Reclamation Headquarters General Hospital, Harbin, China

OPEN ACCESS

Edited by:

Xiangxiang Zeng,
Hunan University, China

Reviewed by:

Lei Chen,
Shanghai Maritime University, China
Renhai Chen,
Tianjin University, China

*Correspondence:

Xuehong Li
lixuehong1978@163.com
Lesong Wei
s2030143@s.tsukuba.ac.jp

Specialty section:

This article was submitted to
Computational Genomics,
a section of the journal
Frontiers in Genetics

Received: 03 February 2021

Accepted: 02 March 2021

Published: 31 March 2021

Citation:

Yang X, Ye X, Li X and Wei L
(2021) iDNA-MT: Identification DNA
Modification Sites in Multiple Species
by Using Multi-Task Learning Based
a Neural Network Tool.
Front. Genet. 12:663572.
doi: 10.3389/fgene.2021.663572

Motivation: DNA N4-methylcytosine (4mC) and N6-methyladenine (6mA) are two important DNA modifications and play crucial roles in a variety of biological processes. Accurate identification of the modifications is essential to better understand their biological functions and mechanisms. However, existing methods to identify 4mA or 6mC sites are all single tasks, which demonstrates that they can identify only a certain modification in one species. Therefore, it is desirable to develop a novel computational method to identify the modification sites in multiple species simultaneously.

Results: In this study, we proposed a computational method, called iDNA-MT, to identify 4mC sites and 6mA sites in multiple species, respectively. The proposed iDNA-MT mainly employed multi-task learning coupled with the bidirectional gated recurrent units (BGRU) to capture the sharing information among different species directly from DNA primary sequences. Experimental comparative results on two benchmark datasets, containing different species respectively, show that either for identifying 4mA or for 6mC site in multiple species, the proposed iDNA-MT outperforms other state-of-the-art single-task methods. The promising results have demonstrated that iDNA-MT has great potential to be a powerful and practically useful tool to accurately identify DNA modifications.

Keywords: multi-task learning, DNA modification, feature representation, deep learning, neural network

INTRODUCTION

DNA modifications have been identified in multiple species. DNA modification plays an irreplaceable role in many basic biological functions (Fu and He, 2012; Shen and Zou, 2020). It refers to add methyl or hydroxymethyl groups to the nucleotides of DNA molecules. In particular, it is essential in the normal development of organisms such as aging, carcinogenesis, and X chromosome inactivation. Due to its importance, DNA methylation is one of the most widely studied epigenetic modifications (Bergman and Cedar, 2013; Smith and Meissner, 2013). Currently,

four out of the DNA modifications, such as N4-methylcytosine (4mC), N6-methyladenine (6mA), 5-methylcytosine (5mC), and 5-hydroxymethylcytosine (5hmC), have been extensively studied (Cheng and Baldi, 2006; Guohua et al., 2017; He et al., 2019; Luo et al., 2020; Zuo et al., 2020c).

Schweizer (2008) proposed that 4mC has the effect of protecting the host DNA from degradation by restriction enzymes and belongs to restriction-modification (RM) systems. Timinskas et al. (1995) proposed 4mC can methylate the 4th amino group of cytosine in DNA under the catalysis of N-4 cytosine-specific DNA methyltransferase (DNMT). Iyer et al. (2011) proposed 4mC can distinguish the self and foreign DNA of prokaryotes and repair DNA replication errors. 5hmC arises from the oxidation of 5-methylcytosine (5mC) by Fe^{2+} and 2-oxoglutarate-dependent 10–11 translocation (TET) family proteins (Hu et al., 2019). Thomson and Meehan (2016) proposed 5hmC can be used as an identifier of cell type or disease state. It is an intermediate product produced during the 5mC demethylation process. Szulwach et al. (2011) proposed 5hmC is critical in neurodevelopment and diseases (Tang et al., 2018; Zhang Y. et al., 2019). 6mA is a non-canonical DNA base modification present at low levels and maybe a carrier of heritable epigenetic information in eukaryotes (Greer et al., 2015; Mondo et al., 2017) and is found in the genomes of certain protists and fungi and might exist in other eukaryotes (Wion and Casadesús, 2006). The role of 6mA is very extensive. For example, it protects against restriction enzymes in bacteria (Heyn and Esteller, 2015) and unravels the DNA double helix structure during the cell cycle (Fang et al., 2012), which is catalyzed by two classes of DNA adenine methyltransferases (Wion and Casadesús, 2006; Zhang L. et al., 2019).

Numerous studies have shown that 5hmC, 6mA, and 4mC, and others are widely present in the genome, and significant progress has been made (Wu et al., 2016; Ao et al., 2019; Hu et al., 2019; Zhu et al., 2019; Zou et al., 2019; Cai et al., 2020; Fu et al., 2020; Hong et al., 2020). However, methylation-related technologies—the short-read sequencing and long-read have major disadvantages. For example, short-read technology can convert unmethylated cytosine to uracil. However, it has intrinsic disadvantages, such as low positioning efficiency and low accuracy. Long-read sequencing can be used to identify DNA modifications. There is a problem that it does not have a high signal-to-noise ratio for DNA modification. In nature, 5hmC, 6mA, and 4mC content are low, and the requirements for detection technology are relatively high. Therefore, we perform predictive calculations in advance, which can improve the efficiency of the experiment, to reduce the cost of the experiment, and provide guidance information for subsequent implementations.

Recently, there have been many machine learning methods to predict DNA methylation sites (Basith et al., 2019; Chen and Zou, 2019; Dou et al., 2020; Lv et al., 2020b). For instance, Ni et al. (2019) proposed DeepSignal, a deep learning approach to detect DNA methylation states from Nanopore sequencing reads. Besides, Liu et al. (2016) designed a two-way neural network with long short-term memory, called DeepMod. It can also identify DNA methylation sites in *E. coli* and *Homo sapiens*.

Chen et al. (2019) developed a computational method called i6mA-Pred, to identify 6mA sites targeted to the rice genome, in which the optimal nucleotide chemical properties obtained by the using feature selection technique were used to encode the DNA sequences. Similarly, Yu and Dai (2019) created SNNRice6mA based on deep learning to identify 6mA in rice.

Kong and Zhang (2019) proposed a new machine learning-based method, namely i6mA-DNCP, which proved that there is also 6mA sites also in the rice genome. In i6mA-DNCP, dinucleotide composition and dinucleotide-based DNA properties were first employed to represent DNA sequences. Chen et al. (2017) developed iDNA4mC, the first webserver to identify 4mC sites, in which DNA sequences are encoded with both nucleotide chemical properties and nucleotide frequency. Later on, Wei et al. (2019b) developed a new predictor named 4mCPred-IFL to identify 4mC sites, in which they proposed an iterative feature representation algorithm that enables learning informative features from several sequential models in a supervised iterative mode. Basith et al. (2019) developed a novel computational predictor, called the Sequence-based DNA N6-methyladenine predictor (SDM6A), which is a two-layer ensemble approach for identifying 6mA sites in the rice genome. Manavalan et al. (2019a) designed the first method for identifying 4mC sites in the mouse genome, called 4mCPred-EL. Similarly, Hasan et al. (2020) invented a method to identify the 4mC sites, called i4mC-ROSE in the *Fragaria vesca* and *Rosa* genome. However, the training data of the above methods are all derived from specific species. And when extended to other species, it may produce a low true-positive rate with a high false-positive rate. Therefore, there is urgent to develop a generic DNA modification site predictor that can be used in different species. In other biological and medical fields, machine learning-based computational methods have been widely used, including microRNAs and cancer association prediction (Yuming et al., 2015; Jiang et al., 2018; Ding et al., 2020a; Wang et al., 2021), function prediction of proteins (Ding et al., 2019d, 2020b; Wang Y. et al., 2019; Wang H. et al., 2019; Tao et al., 2020; Zou et al., 2020b; Yang et al., 2021), drugs complex network analysis (Ding et al., 2017, 2019a,b,c, 2020c; Guo et al., 2020b) and dry weight assessment of hemodialysis patients (Guo et al., 2020a).

In this study, we developed a new deep learning-based multi-task method, called iDNA-MT, for identifying 4mC site and 6mA site in multiple species, respectively. This method combines both the bidirectional gated recurrent units (BGRU) and multi-task learning to learn sharing information hiding in different species for better characterizing a DNA sequence. Afterward, the sharing features are fed into the corresponding fully connected layers, specifically designed for a certain task, to identify the modification site. Several experiments were carried out to investigate the performance of the proposed iDNA-MT. Experimental results on two benchmark datasets showed that iDNA-MT achieved significantly better performance than state-of-the-art single-task methods for identifying 4mC site and 6mA site, respectively. In addition, our model can provide a powerful tool for identifying 4mC sites and 6mA sites in multiple species, respectively, and facilitate our knowledge of their biological functions.

MATERIALS AND METHODS

Dataset

For a fair comparison, we employed the same benchmark datasets derived from Lv et al. (2020a). Four species of 4mC site data and four species of 6mA site data were selected. The 4mC site data contains four species (*C. equisetifolia*, *F. vesca*, *S. cerevisiae*, and *Tolypocladium*) that were collected from the MDR database (Liu et al., 2016) and MethSMRT database (Pohao et al., 2017). The 6mA site data for four species (*Tolypocladium*, *C. elegans*, *C. equisetifolia*, and *R. chinensis*) were extracted from the MethSMRT database (Pohao et al., 2017), MethSMRT database (Pohao et al., 2017), and MDR database (Liu et al., 2016). The benchmark data is divided into two parts. One part is used as a training dataset, and the other one is a testing dataset. The function of the training dataset is to train and evaluate the predictive model, while the purpose of the testing dataset is to test the performance of the model. The number of positive and negative samples is the same in the training dataset and testing dataset. A summary of the different species datasets used for benchmarking is displayed in **Table 1**.

Neural Network Architecture of the Proposed iDNA-MT

In this section, we introduce the network architecture of our model iDNA-MT, as illustrated in **Figure 1**. This network architecture consists of three main components: (i) sequence processing module, (ii) sharing module, and (iii) task-specific output module. To make DNA sequences recognized easily by the neural network, the sequence processing module is designed to encode the original DNA sequences into matrices by one-hot encoding (Quang and Xie, 2016). Next, the encoded matrix is passed through a bidirectional GRU to extract different levels of dependency relationships between subsequences, and then a max-pooling layer is employed to automatically measure which feature plays a key role in NDA methylation site identification in each unit of the GRU. Finally, the features learned from the max-pooling layer are sent to the task-specific output module to identify 6mA sites in four species, respectively. The task-specific output module contains four parts and each part consists of fully connected layers that are designed in terms of the size of the training set of each species. The model is implemented using Pytorch. Below each module of our model is described in detail.

Sequence Processing Module

DNA modification identification is the task to separate the DNA sequences into related classes of DNA modifications, while text categorization is the problem of assigning text documents to predefined categories. To apply text categorization techniques to DNA sequences, we first employed n-gram nucleobases to define “words” in DNA sequences (Dong et al., 2006; Dao et al., 2020; Wang et al., 2020; Zhang et al., 2020). The n-grams are the set of all possible subsequences of nucleobases. Then, we split the DNA sequences into overlapping n-gram nucleobases. The number of possible it is 4^n , since there are four types of nucleobases (Yang et al., 2020). In this study, to avoid low-frequency words in the encoding, the n-gram number n is set to 2. For example, we split a DNA sequence into overlapping 2-gram nucleobase sequences as follows: *GTTGT...CTT* → “GT,” “TT,” “TG,” “GT,” ..., “CT,” “TT.”

For a given DNA sequence P of length L, it can be expressed as follows:

$$P = R_1, R_2, \dots, R_L \quad (1)$$

where R_i is the i -th word. These words are first randomly initialized embedded by one-hot embedding, which is referred to as “word embeddings.” Here, we defined the sequence of word embeddings as:

$$x_1, x_2, \dots, x_L \quad (2)$$

where $x_i \in \mathbb{R}^d$ is the d-dimensional embedding of the i -th word. In the proposed method, such a sequence is fed into the bidirectional GRU to extract dependency information.

Sharing Module

Bidirectional Gated Recurrent Units

GRU is one of the widely used deep learning techniques, which is designed to specifically address the problems of learning long-distance correlations in a sequence (Cho et al., 2014). Bidirectional GRU is the most important part of the sharing module, which is employed to automatically extract long-terms and short-term dependency relationships in DNA sequences. The structure of the basic unit of GRU is shown in **Figure 2**. The unit receives two input vectors: the embedding vector of the subsequence and the hidden state of the previous time step. The special thing about them is that they can be trained to keep information from long ago. Based on the two inputs, two gates, namely, reset gate and update gate, coordinate with each other to capture short-term and long-term dependencies in sequences. The reset gate is used to control how much of the previous information to forget. Likewise, the update gate helps the model to determine how much of the past information, from previous time steps, needs to be passed along to the future.

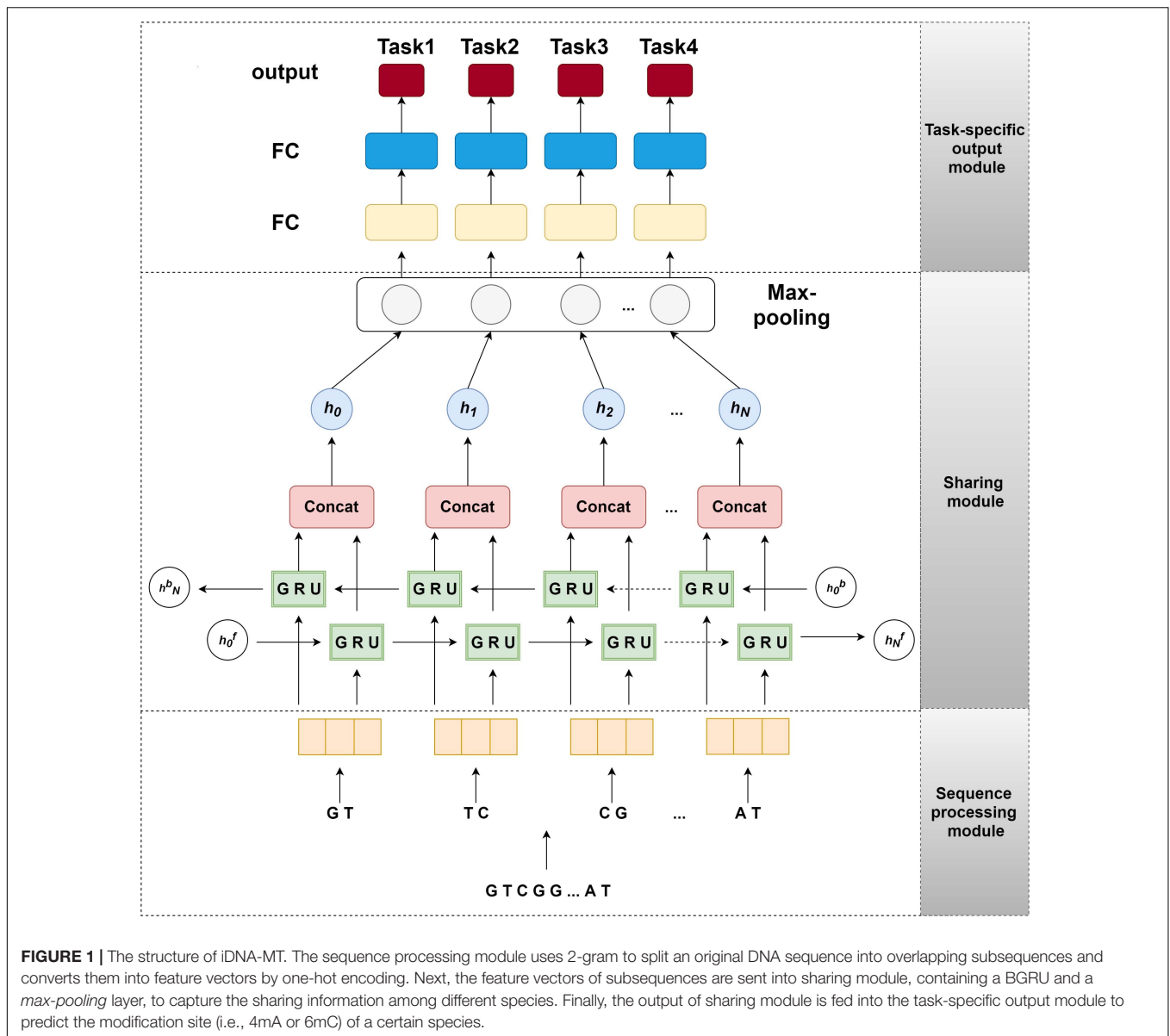
For a given time step t , there are four components composite the GRU-based recurrent neural network: a reset gate r_t with corresponding weight matrices W_r, U_r ; an update gate z_t with corresponding weight matrices W_z, U_z ; a candidate hidden state h'_t with corresponding weight matrices W, U ; and a new hidden state h_t . The equations of GRU are the following:

$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \quad (3)$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \quad (4)$$

TABLE 1 | Summary of benchmark datasets used in this study.

Modifications	Species	Testing dataset	Training dataset
4mC	<i>C. equisetifolia</i>	365	365
	<i>F. vesca</i>	15,795	15,797
	<i>S. cerevisiae</i>	1,977	1,979
	<i>Tolypocladium</i>	15,325	15,327
6mA	<i>Tolypocladium</i>	3,377	3,379
	<i>C. elegans</i>	7,959	7,961
	<i>C. equisetifolia</i>	6,065	6,065
	<i>R. chinensis</i>	597	599



$$h'_t = \tanh(Wx_t + r_t \odot Uh_{t-1}) \quad (5)$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot h'_t \quad (6)$$

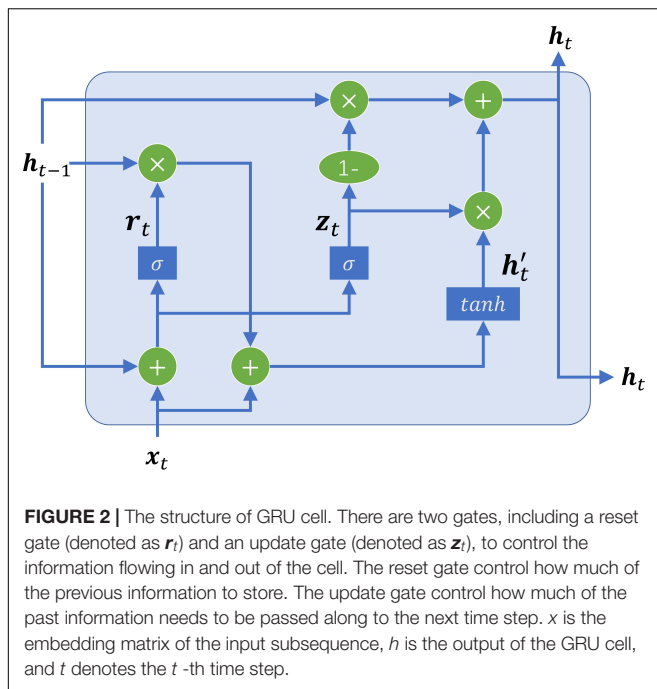
where x_t denotes the input of the current time step, σ denotes the logistic sigmoid function to transform input values to the interval $(0, 1)$, h_{t-1} denotes the output of the last time step, \odot denotes element-wise multiplication, and \tanh is a non-linear activation function to ensure the values in the candidate hidden state remain in the interval $(-1, 1)$. Hence, the new hidden state h_t holds information for the current step and previous steps and passes it down to the network.

However, a standard GRU network process a sequence in temporal order, resulting in that the outputs only contain the forward sequence information. To fully extract the information

of a sequence, it is significant to capture not only the forward information but also the backward information at each time step. Therefore, we attempt to add another GRU network that captures the backward sequence information by processing a DNA sequence in the opposite temporal order. Combine it with the standard GRU network to form a bidirectional GRU, which can exploit information both from the past and the future.

To better capture the dependency information of subsequences with large time step distances, in this study, we combined the forward and backward hidden vectors generated by bidirectional GRU in each step. Therefore, the i -th subsequence can be expressed as the following vector:

$$h_i = (h_i^f, h_i^b) \quad (7)$$



where h is the hidden vector, h_i^f and h_i^b denote the hidden vectors generated by the forward GRU and the backward GRU, respectively.

Max-pooling Layer

The feature vector h of each subsequence, generated by bidirectional GRU, is fed into a *max-pooling* layer to capture the most significant feature in identifying the DNA modification to represent this subsequence. Then, all the most significant features of subsequences are concatenated into a vector to represent a DNA sequence, which is shown in the following equation:

$$y = \max_{i=1}^n h_i \quad (8)$$

where i is the i -th subsequence, n is the number of subsequences in a DNA sequence, and the y is regarded as the feature vector of a target sequence. The max-pooling layer attempts to find the most important dependencies in subsequences.

Task-Specific Output Module

This module consists of four sets of fully connected layers corresponding to each task, respectively. In each fully connected layer with a *relu* activation function, its output is calculated by the following equation:

$$f_i^j = \text{relu}(W_i^j f_{i-1}^j + b_i^j) \quad (9)$$

where f_{i-1}^j is the output of the previous layer of j -th task, f_i^j is the current layer output of j -th task, W_i^j is the weight matrix, and b_i^j is the bias vector. In each layer, the “Batch Normalization” technique was used to improve generalization performance (Cheng and Baldi, 2006). Finally, a *softmax* layer is added on the top of final output f_i^j to perform the final prediction.

Note that the parameters of different set of the fully connected layer are designed differently in terms of the amount of data of the corresponding task.

Training

The task-specific features y , generated by the sharing module, are ultimately sent into one set of fully connected layers in terms of it belonging to which task. For classification tasks, we used binary cross-entropy loss function as the objective:

$$l = \frac{1}{N} \sum_i -[y_i \log(p_i) + (1 - y_i) \log(1 - p_i)] \quad (10)$$

where N denotes the number of training samples, y_i denotes the label (i.e., 1 or 0) of sample i , p_i denotes the probability that sample i is predicted to be positive. Our global loss function is the linear combination of loss function for all tasks:

$$l_{all} = \sum_{k=1}^k \alpha_k l_k \quad (11)$$

where α_k is the weight for task k .

It is worth noting that the samples for training each task can come from completely different datasets. Following the study (Liu et al., 2016), the training is carried out in a stochastic manner by looping over the tasks:

1. Select a task randomly.
2. Select a training sample from this task randomly.
3. Update the parameters for this task by taking a gradient step in terms of this sample.
4. Go to 1.

Evaluation Metrics

To evaluate the performance of our model, four commonly used metrics are employed to evaluate the performance of the model (Zou et al., 2016; Jin et al., 2019, 2020; Manayalan et al., 2019; Manavalan et al., 2019b; Hong et al., 2020; Lv et al., 2020b; Qiang et al., 2020; Su et al., 2020a,b,c, 2019a,b; Wei et al., 2020, 2014, 2019a, 2018a,b; Zhao et al., 2020; Zou et al., 2020a), including sensitivity (SN), specificity (SP), overall accuracy (ACC), and Matthew's correlation coefficient (MCC), respectively. They are formulated as:

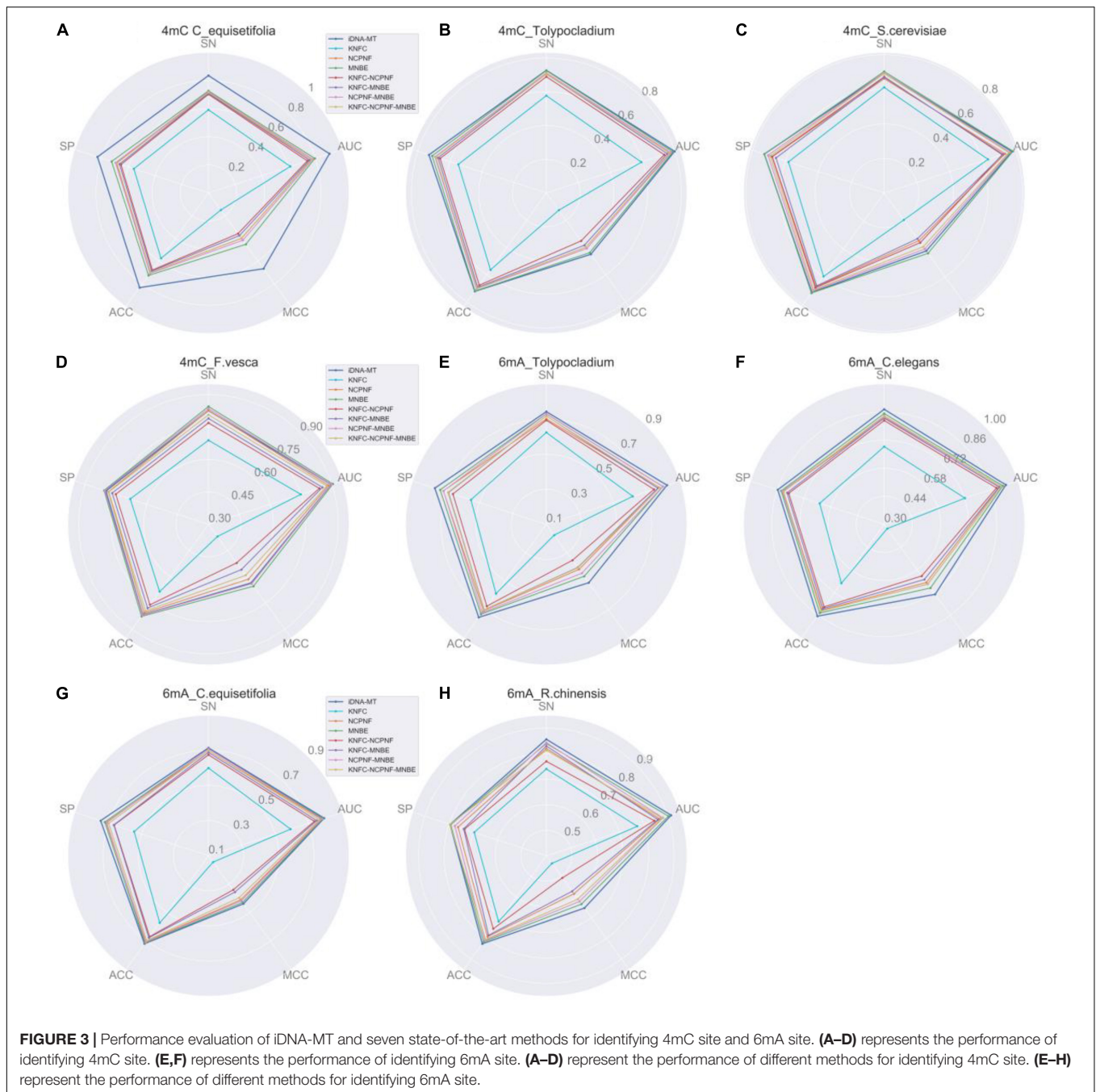
$$SN = \frac{TP}{TP + FN} \quad (12)$$

$$SP = \frac{TN}{TN + FP} \quad (13)$$

$$ACC = \frac{TP + TN}{TP + FN + TN + FP} \quad (14)$$

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FN) \times (TP + FP) \times (TN + FN) \times (TN + FP)}} \quad (15)$$

where TP, TN, FP, and FN represent the numbers of true positives, true negatives, false positives, and false negatives,



respectively. SN and SP are used to evaluate positive and negative predictive ability. MCC and ACC were used to evaluate the overall prediction performance. Besides, the ROC curve (receiver operating characteristic curve) can be used to visualize the performance of the classifier. In addition, we calculate the area under the ROC curve (AUC) to evaluate the prediction performance of the model. The range of AUC is 0.5–1. The higher the AUC score, the better the prediction performance of the model.

RESULTS AND DISCUSSION

Performance Comparison With the State-of-the-Art Methods

To evaluate the performance of our model iDNA-MT for identifying 4mC and 6mA site in multiple species, we compared it with seven state-of-the-art models based on random forest (RF), which were all single-task learning methods and used different feature descriptors to identify 4mC and 6mA site in each species, respectively, including K-tuple nucleotide frequency component

(KNFC), nucleotide chemical property and nucleotide frequency (NCPNF), and mono-nucleotide binary encoding (MNBE), and their four combinations (Lv et al., 2020a).

The experimental results of different methods are listed in **Figure 3**. From **Figure 3**, we can observe that for 4mC site identification, our proposed iDNA-MT significantly outperforms all the other competing methods in three species (*C. equisetifolia*, *Tolypocladium*, and *S. cerevisiae*) in terms of five metrics (SN, SP, ACC, MCC, and AUC), while the model using MNBE achieves the best performance amongst all methods. For 6mA site identification, iDNA-MT exhibits better performance than any other RF-based models in each species. These results indicate that using both BGRU and multi-task learning can extract more effective and discriminative features to represent DNA sequences for identifying 4mA site and 6mC site and be generalized well on different species. There are two main reasons for the outstanding performance of our model. First, compared with the RF-based methods that use handcrafted features to train models, which need prior knowledge, iDNA-MT can automatically capture effective features by data driving. Second, the proposed iDNA-MT employs the BGRU to learn long-distance dependency information of DNA subsequences, and then introduce the multi-task learning technique to capture the shared information hidden in data from different species to improve the performance of each task, to improve the accuracy for identifying 4mC and 6mA site in multiply species, respectively. Therefore, iDNA-MT can achieve better performance than other state-of-the-art single-task learning methods. Note that the detailed comparison results of iDNA-MT and seven state-of-the-art methods can be found in **Supplementary Table S1**.

Effectiveness of Multi-Task Learning

To evaluate whether or not introducing the multi-task learning technique can capture more discriminative features to improve the performance of DNA modification site prediction in multiple species, we compared the model considering the multi-task learning, namely iDNA-MT, with the model not considering the multi-task learning for prediction. The comparative results for 4mC site and 6mA site are illustrated in **Tables 2, 3**, respectively. In **Tables 2, 3**, we show better results in bold.

As shown in **Table 2** for 4mC site prediction, we can see that training with the multi-task learning, the model achieves higher performance in three species, including *C. equisetifolia*, *Tolypocladium*, and *S. cerevisiae*, with only one exception in *F. vesca*. Specifically, the model using the multi-task learning achieves an ACC of 83.33%, an MCC of 0.6667, and an AUC of 0.9049 for species *C. equisetifolia*, yielding a relative improvement of 2.3%, 5.7%, and 5.8%, respectively, achieves an ACC of 72.09%, an MCC of 0.4489 and an AUC of 0.7989 for species *Tolypocladium*, yielding a relative improvement of 1.1%, 3.0%, and 1.9%, respectively, and achieves an ACC of 71.09%, an MCC of 0.4139 and an AUC of 0.7765 for species *S. cerevisiae*, yielding a relative improvement of 2.2%, 5.5%, and 3.3%, respectively. For species *F. vesca*, the model using multi-task learning is slightly worse than the model not using multi-task learning, which achieves 82.67%, 79.86%, 81.79%, 0.6354, and 0.8966 in terms of SN, SP, ACC, MCC, and AUC. From **Table 3**, we

can see that for all four species (*Tolypocladium*, *C. elegans*, *C. equisetifolia*, and *R. chinensis*), the model using multi-task learning all significantly outperforms the model not using multi-task learning for identification 6mA site in terms of SN, SP, ACC, MCC, and AUC. The most significant improvement is observed in species *R. chinensis*, in which the model using multi-task learning improves the SN from 78.93% to 85.62%, the SP from 72.24% to 79.62%, the ACC from 75.85% to 82.61%, the MCC from 0.5129 to 0.6534 and the AUC from 0.8334 to 0.9134.

These results discussed above demonstrate that by introducing the multi-task learning, the model can achieve outstanding performance for 4mC site and 6mA site prediction in multiply species, respectively. The reason may be that multi-task learning aims to learn shared representations from multiple related tasks, which are used to share and supplement the information learned from different tasks to improve the performance of multiple related learning tasks. Therefore, there is not surprising that the model using multi-task learning significantly outperforms the model not using multi-task learning.

Performance of the Neural Network Architecture in Sharing Module

The sharing module of iDNA-MT mainly employed BGRU to exploit the potential information both from forward and backward and then used the *max-pooling* layer to extract the most significant features in subsequences, which play key roles in DNA modification identification. To evaluate the efficiency and superiority of the neural network architecture in sharing module,

TABLE 2 | Comparison results of the model using the multi-task learning and the model not using the multi-task learning for identifying 4mC site.

Modification type	Genome		SN (%)	SP (%)	ACC (%)	MCC	AUC
4mC	<i>C. equisetifolia</i>	Single	77.05	85.79	81.42	0.6308	0.8551
		Multi	83.61	83.06	83.33	0.6667	0.9049
	<i>Tolypocladium</i>	Single	69.94	72.61	71.28	0.4357	0.7837
		Multi	72.72	73.12	72.09	0.4489	0.7989
	<i>S. cerevisiae</i>	Single	66.23	72.91	69.57	0.3922	0.7520
		Multi	69.32	72.88	71.09	0.4139	0.7765
	<i>F. vesca</i>	Single	83.48	82.06	82.77	0.6544	0.9047
		Multi	82.67	79.86	81.79	0.6354	0.8966

TABLE 3 | Comparison results of the model using the multi-task learning and the model not using the multi-task learning for identifying 6mA site.

Modification type	Genome		SN (%)	SP (%)	ACC (%)	MCC	AUC
6mA	<i>Tolypocladium</i>	Single	73.96	74.25	74.91	0.5001	0.8170
		Multi	74.25	76.73	75.49	0.5110	0.8222
	<i>C. elegans</i>	Single	87.51	85.55	86.53	0.7308	0.9334
		Multi	87.39	85.73	86.56	0.7313	0.9374
	<i>C. equisetifolia</i>	Single	69.47	71.12	70.29	0.4059	0.7696
		Multi	71.45	74.55	72.01	0.4385	0.7923
	<i>R. chinensis</i>	Single	78.93	72.24	75.85	0.5129	0.8334
		Multi	85.62	79.62	82.61	0.6534	0.9134

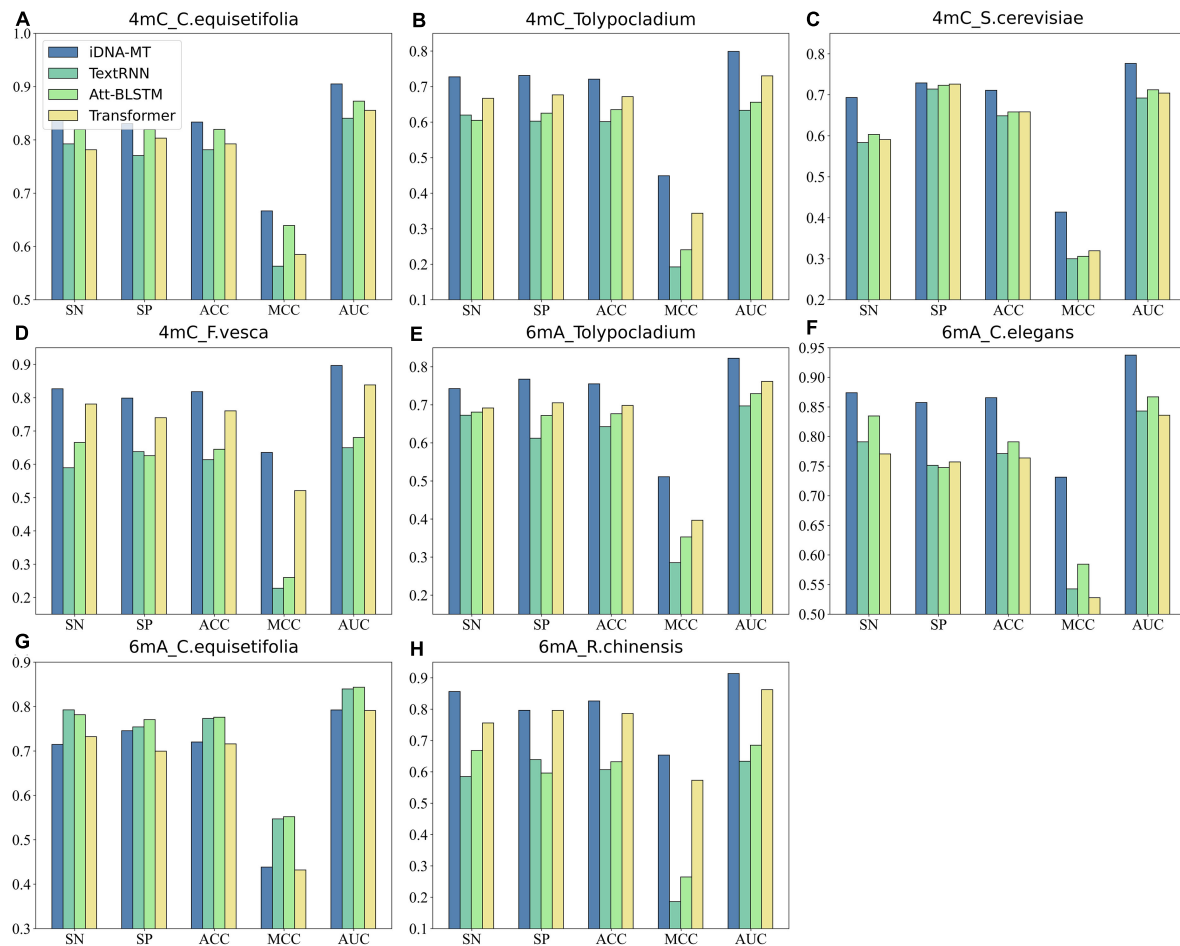


FIGURE 4 | Performance evaluation of iDNA-MT and other methods using different typical text classification methods in sharing module (A–D) represent the performance of iDNA-MT and other methods for identifying 4mC site. (E–H) represent the performance of iDNA-MT and other methods for identifying 6mA site.

we replaced it with other three typical text classification methods, respectively, including:

1. TextRNN (Liu et al., 2016): It uses the long short-term memory network (LSTM) to capture long-term semantic dependencies in a sentence.
2. Att-BLSTM (Zhou et al., 2016): It utilizes both neural attention mechanism and bidirectional long short-term memory networks (BLSTM) to capture the most important semantic information in a sentence.
3. Transformer (Vaswani et al., 2017): It is a novel neural network architecture based on a self-attention mechanism.

Figure 4 shows the comparison results of the proposed iDNA-MT and the other methods using different typical text classification methods in sharing modules on two different modification sites in terms of five metrics (SN, SP, ACC, MCC, and AUC). As shown in Figure 4, we can see that for 4mC site, the performance of iDNA-MT is significantly better than the other methods using different typical text classification methods in sharing module in every species. For 6mA site, although the performance of iDNA-MT is lower than other methods in species

C. equisetifolia, the performance of iDNA-MT significantly outperforms other methods in the rest species. Therefore, iDNA-MT is superior to other methods in identifying 4mC sites and 6mA sites in multiple species, respectively. The proposed iDNA-MT used BGRU to capture the dependency information of subsequences from the past and the future and added a *max-pooling* layer to extract the most important information hiding in every subsequence, which avoids irrelevant information from interfering with identifying results. Therefore, there is no surprise that iDNA-MT achieves the best performance when combining BGRU and a *max-pooling* layer.

CONCLUSION

Although 4mA and 6mC are two important genetic modifications and play crucial roles in regulating a series of biological processes, their biological functions are still unclear. Therefore, the accurate identification of them is pivotal to understand specific biological functions. In this study, we proposed a multi-task learning predictor namely iDNA-MT for identifying 4mA site and 6mC site in multiple species, respectively, which can automatically

extract the discriminative features for different tasks. To better represent the DNA sequences of different species, we constructed a sharing module, containing a BGRU and a *max-pooling* layer, to capture sharing information among different species. To evaluate the efficiency of our multi-task model, we compared it with the state-of-the-art single-task models on benchmark datasets of two different DNA modifications. Experimental results have shown that the proposed iDNA-MT achieved the top performance comparing with existing single-task models on two benchmark datasets, indicating that multi-task learning can improve the performance of multiple related tasks by leveraging useful information among them. In future work, we would like to investigate other sharing mechanisms to further improve the prediction of different DNA modifications in multiple species and apply it to other fields (Wei et al., 2017a,b,c, 2018c, 2019c,d; Zou et al., 2019).

DATA AVAILABILITY STATEMENT

All datasets generated for this study are included in the article/**Supplementary Material**, further inquiries can be directed to the corresponding author/s.

REFERENCES

- Ao, C., Jin, S., Lin, Y., and Zou, Q. (2019). Review of progress in predicting protein methylation sites. *Curr. Organ. Chem.* 23, 1663–1670. doi: 10.2174/1385272823666190723141347
- Basith, S., Manavalan, B., Shin, T. H., and Lee, G. (2019). SDM6A: a web-based integrative machine-learning framework for predicting 6mA sites in the rice genome. *Mol. Therapy - Nucleic Acids* 18, 131–141. doi: 10.1016/j.omtn.2019.08.011
- Bergman, Y., and Cedar, H. (2013). DNA methylation dynamics in health and disease. *Nat. Struct. Mol. Biol.* 20, 274–281.
- Cai, L., Ren, X., Fu, X., Peng, L., Gao, M., and Zeng, X. J. B. (2020). iEnhancer-XG: interpretable sequence-based enhancers and their strength predictor. *Bioinformatics* doi: 10.1093/bioinformatics/btaa914 Online ahead of print.
- Chen, W., Lv, H., Nie, F., and Lin, H. (2019). i6mA-Pred: identifying DNA N6-methyladenine sites in the rice genome. *Bioinformatics* 35, 2796–2800. doi: 10.1093/bioinformatics/btz015
- Chen, W., Yang, H., Feng, P., Ding, H., and Lin, H. (2017). iDNA4mC: identifying DNA N4-methylcytosine sites based on nucleotide chemical properties. *Bioinformatics* 33, 3518–3523.
- Chen, J. L. J., and Zou, Q. (2019). DeepM6ASeq-EL: prediction of human N6-methyladenosine (m6A) Sites with LSTM and ensemble learning. *Front. Comput. Sci.* doi: 10.1007/s11704-020-0180-0
- Cheng, J., and Baldi, P. (2006). A machine learning information retrieval approach to protein fold recognition. *Bioinformatics* 22, 1456–1463.
- Cho, K., Van Merriënboer, B., Bahdanau, D., and Bengio, Y. (2014). “On the properties of neural machine translation: encoder-decoder approaches,” in *Proceedings of the SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, (CityplaceDoha: Association for Computational Linguistics).
- Dao, F. Y., Lv, H., Zhang, D., Zhang, Z. M., Liu, L., and Lin, H. (2020). DeepYY1: a deep learning approach to identify YY1-mediated chromatin loops. *Brief. Bioinform.* doi: 10.1093/bib/bbaa356. Online ahead of print.
- Ding, Y., Jiang, L., Tang, J., and Guo, F. (2020a). Identification of human microRNA-disease association via hypergraph embedded bipartite local model. *Comput. Biol. Chem.* 89:107369. doi: 10.1016/j.compbiolchem.2020.107369
- Ding, Y., Tang, J., and Guo, F. (2020b). Human protein subcellular localization identification via fuzzy model on kernelized neighborhood representation. *Appl. Soft Comput.* 96:106596. doi: 10.1016/j.asoc.2020.106596
- Ding, Y., Tang, J., and Guo, F. (2020c). Identification of Drug-Target interactions via dual laplacian regularized least squares with multiple kernel fusion. *Knowledge-Based Systems* 204:106254. doi: 10.1016/j.knosys.2020.106254
- Ding, Y., Tang, J., and Guo, F. (2017). Identification of drug-target interactions via multiple information integration. *Inform. Sci.* 418, 546–560. doi: 10.1016/j.ins.2017.08.045
- Ding, Y., Tang, J., and Guo, F. (2019a). Identification of drug-side effect association via multiple information integration with centered kernel alignment. *Neurocomputing* 325, 211–224. doi: 10.1016/j.neucom.2018.10.028
- Ding, Y., Tang, J., and Guo, F. (2019b). Identification of drug-side effect association via semisupervised model and multiple kernel learning. *IEEE J. Biomed. Health Inform.* 23, 2619–2632. doi: 10.1109/jbhi.2018.2883834
- Ding, Y., Tang, J., and Guo, F. (2019c). Identification of drug-target interactions via fuzzy bipartite local model. *Neural Comp. Appl.* 32, 10303–10319. doi: 10.1007/s00521-019-04569-z
- Ding, Y., Tang, J., and Guo, F. (2019d). Protein crystallization identification via fuzzy model on linear neighborhood representation. *IEEE/ACM Trans. Comput. Biol. Bioinform.* Online ahead of print.
- Dong, Q.-W., Wang, X.-L., and Lin, L. (2006). Application of latent semantic analysis to protein remote homology detection. *Bioinformatics* 22, 285–290. doi: 10.1093/bioinformatics/bti801
- Dou, L. J., Li, X. L., Ding, H., Xu, L., and Xiang, H. K. (2020). Is there any sequence feature in the RNA pseudouridine modification prediction problem? *Mol. Ther.-Nucl. Acids* 19, 293–303. doi: 10.1016/j.omtn.2019.11.014
- Fang, G., Munera, D., Friedman, middlemameplaceD. middlemameI., Mandlik, A., Chao, M. C., Banerjee, O., et al. (2012). Genome-wide mapping of methylated adenine residues in pathogenic *Escherichia coli* using single-molecule real-time sequencing. *Nat. Biotechnol.* 30, 1232–1239. doi: 10.1038/nbt.2432
- Fu, X., Cai, L., Zeng, X., and Zou, Q. J. B. (2020). StackCPPred: a stacking and pairwise energy content-based prediction of cell-penetrating peptides and their uptake efficiency. *Bioinformatics* 36, 3028–3034. doi: 10.1093/bioinformatics/btaa131
- Fu, Y., and He, C. (2012). Nucleic acid modifications with epigenetic significance. *Curr. Opin. Chem. Biol.* 16, 516–524. doi: 10.1016/j.cbpa.2012.10.002

AUTHOR CONTRIBUTIONS

XY and LW surveyed the algorithms and implementations, preprocessed the datasets, and performed all the analyses. XCY and XL designed the benchmarking test. All the authors have written, read, and approved the manuscript.

FUNDING

This work was supported in part by the New Energy and Industrial Technology Development Organization 265 (NEDO) and the Japan Society for the Promotion of Science (JSPS), Grants-in-Aid for Scientific Research under Grant 18H03250, and the Natural Science Foundation of China.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/fgene.2021.663572/full#supplementary-material>

- Greer, E. L., Blanco, M. A., Gu, L., Sendinc, E., Liu, J., Aristizábal-Corralles, D., et al. (2015). DNA Methylation on N6-Adenine in *C. elegans*. *Cell* 161, 868–878. doi: 10.1016/j.cell.2015.04.005
- Guo, X. Y., Zhou, W., Shi, B., Wang, X. H., Du, A. Y., Ding, Y. J., et al. (2020a). An efficient multiple kernel support vector regression model for assessing dry weight of hemodialysis patients. *Curr. Bioinform.* 15, 466–469.
- Guo, X. Y., Zhou, W., Yu, Y., Ding, Y. J., Tang, J. J., and Guo, F. (2020b). A novel triple matrix factorization method for detecting drug-side effect association based on kernel target alignment. *BioMed Res. Int.* 2020, 1–11. doi: 10.1155/2020/4675395
- Guohua, W., Ximei, L., Jianan, W., Jun, W., Shuli, X., Heng, Z., et al. (2017). MeDReaders: a database for transcription factors that bind to methylated DNA. *Nucleic Acids Res.* 46, D146–D151.
- Hasan, M. M., Manavalan, B., Khatun, M. S., and Kurata, H. (2020). i4mC-ROSE, a bioinformatics tool for the identification of DNA N4-methylcytosine sites in the Rosaceae genome. *Int. J. Biol. Macromol.* 157, 752–758. doi: 10.1016/j.ijbiomac.2019.12.009
- He, W., Jia, C., and Zou, Q. (2019). 4mCPred: machine learning methods for DNA N4-methylcytosine sites prediction. *Bioinformatics* 35, 593–601. doi: 10.1093/bioinformatics/bty668
- Heyn, H., and Esteller, M. (2015). An adenine code for DNA: a second life for N6-methyladenine. *Cell* 161, 710–713. doi: 10.1016/j.cell.2015.04.021
- Hong, Z., Zeng, X., Wei, L., and Liu, X. J. B. (2020). Identifying enhancer-promoter interactions with neural network based on pre-trained DNA vectors and attention mechanism. *Bioinformatics* 36, 1037–1043.
- Hu, L., Liu, Y., Han, S., Yang, L., Cui, X., Gao, Y., et al. (2019). Jump-seq: genome-wide capture and amplification of 5-Hydroxymethylcytosine sites. *J. Am. Chem. Soc.* 141, 8694–8697. doi: 10.1021/jacs.9b02512
- Iyer, L. M., Abhiman, S., and Aravind, L. (2011). Chapter 2 - natural history of eukaryotic DNA methylation systems. *Prog. Mol. Biol. Transl. Sci.* 101, 25–104. doi: 10.1016/b978-0-12-387685-0.00002-0
- Jiang, L., Xiao, Y., Ding, Y., Tang, J., and Guo, F. (2018). FKL-Spa-LapRLS: an accurate method for identifying human microRNA-disease association. *BMC Genomics* 19:911. doi: 10.1186/s12864-018-5273-x
- Jin, Q., Meng, Z., Tuan, D. P., Chen, Q., Wei, L., and Su, R. (2019). DUNet: a deformable network for retinal vessel segmentation. *Knowledge-Based Systems* 178, 149–162. doi: 10.1016/j.knsys.2019.04.025
- Jin, S., Zeng, X., Xia, F., Huang, W., and Liu, X. (2020). Application of deep learning methods in biological networks. *Brief. Bioinform.* Online ahead of print.
- Kong, L., and Zhang, L. (2019). i6mA-DNCP: computational identification of DNA N6-Methyladenine sites in the rice genome using optimized dinucleotide-based features. *Genes* 10:828. doi: 10.3390/genes10100828
- Liu, P., Qiu, X., and Huang, X. (2016). "Recurrent neural network for text classification with multi-task learning," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)*, (CityShanghai: PlaceNamePlaceFudan PlaceTypeUniversity).
- Luo, X., Wang, F., Wang, G., and Zhao, Y. (2020). Identification of methylation states of DNA regions for Illumina methylation BeadChip. *BMC Genomics* 21:672. doi: 10.1186/s12864-019-6019-0
- Lv, H., Dao, F. Y., Zhang, D., Guan, Z. X., Yang, H., Su, W., et al. (2020a). iDNA-MS: an integrated computational tool for detecting DNA modification sites in multiple genomes. *iScience* 23:100991. doi: 10.1016/j.isci.2020.100991
- Lv, H., Dao, F.-Y., Guan, Z.-X., Yang, H., Li, Y.-W., and Lin, H. (2020b). Deep-Kcr: accurate detection of lysine crotonylation sites using deep learning method. *Brief. Bioinform.* bbaa255. doi: 10.1093/bib/bbaa356
- Manavalan, B., Basith, S., Shin, T. H., Lee, D. Y., Wei, L., and Lee, G. (2019a). 4mCPred-EL: an ensemble learning framework for identification of DNA N4-Methylcytosine sites in the mouse genome. *Cells* 8:1332. doi: 10.3390/cells8111332
- Manavalan, B., Basith, S., Shin, T. H., Wei, L., and Lee, G. (2019b). Meta-4mCPred: a sequence-based meta-predictor for accurate DNA 4mC site prediction using effective feature representation. *Mol. Therapy-Nucleic Acids* 16, 733–744. doi: 10.1016/j.omtn.2019.04.019
- Manayalan, B., Basith, S., Shin, T. H., Wei, L., and Lee, G. (2019). mAHTPred: a sequence-based meta-predictor for improving the prediction of anti-hypertensive peptides using effective feature representation. *Bioinformatics* 35, 2757–2765. doi: 10.1093/bioinformatics/bty1047
- Mondo, S. J., Dannebaum, R. O., Kuo, R. C., Louie, K. B., Bewick, A. J., LaButti, K., et al. (2017). Widespread adenine N6-methylation of active genes in fungi. *Nat. Genet.* 49, 964–968. doi: 10.1038/ng.3859
- Ni, P., Huang, N., Zhang, Z., Wang, D.-P., Liang, F., Miao, Y., et al. (2019). DeepSignal: detecting DNA methylation state from Nanopore sequencing reads using deep-learning. *Bioinformatics* 35, 4586–4595. doi: 10.1093/bioinformatics/btz276
- Pohao, Y., Yizhao, L., Kaining, C., Yizhi, L., Chuanle, X., and Zhi, X. J. (2017). MethSMRT: an integrative database for DNA N6-methyladenine and N4-methylcytosine generated by single-molecular real-time sequencing. *Nucleic Acids Res.* 45, D85–D89.
- Qiang, X., Zhou, C., Ye, X., Du, P.-F., Su, R., and Wei, L. (2020). CPPred-FL: a sequence-based predictor for large-scale identification of cell-penetrating peptides by feature representation learning. *Brief. Bioinform.* 21, 11–23.
- Quang, D., and Xie, X. (2016). DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences. *Nucleic Acids Res.* 44:e107. doi: 10.1093/nar/gkw226
- Schweizer, H. P. (2008). Bacterial genetics: past achievements, present state of the field, and future challenges. *Biotechniques* 44, 636–641.
- Shen, Z., and Zou, Q. (2020). Basic polar and hydrophobic properties are the main characteristics that affect the binding of transcription factors to methylation sites. *Bioinformatics* 36, 4263–4268. doi: 10.1093/bioinformatics/btaa492
- Smith, Z. D., and Meissner, A. (2013). DNA methylation: roles in mammalian development. *Nat. Rev. Genet.* 14, 204–220. doi: 10.1038/nrg3354
- Su, R., Hu, J., Zou, Q., Manavalan, B., and Wei, L. (2020a). Empirical comparison and analysis of web-based cell-penetrating peptide prediction tools. *Brief. Bioinform.* 21, 408–420. doi: 10.1093/bib/bby124
- Su, R., Liu, X., and Wei, L. (2020b). MinE-RFE: determine the optimal subset from RFE by minimizing the subset-accuracy-defined energy. *Brief. Bioinform.* 21, 687–698. doi: 10.1093/bib/bbz021
- Su, R., Liu, X., Xiao, G., and Wei, L. (2020c). Meta-GDBP: a high-level stacked regression model to improve anticancer drug response prediction. *Brief. Bioinform.* 21, 996–1005. doi: 10.1093/bib/bbz022
- Su, R., Liu, X., Wei, L., and Zou, Q. (2019a). Deep-Resp-Forest: a deep forest model to predict anti-cancer drug response. *Methods* 166, 91–102. doi: 10.1016/j.ymeth.2019.02.009
- Su, R., Wu, H., Xu, B., Liu, X., and Wei, L. (2019b). Developing a multi-dose computational model for drug-induced hepatotoxicity prediction based on toxicogenomics data. *IEEE-ACM Trans. Comput. Biol. Bioinform.* 16, 1231–1239. doi: 10.1109/tcbb.2018.2858756
- Szulwach, K. E., Li, X., Li, Y., Song, C. X., and Jin, P. (2011). 5-hmC-mediated epigenetic dynamics during postnatal neurodevelopment and aging. *Nat. Neurosci.* 14, 1607–1616. doi: 10.1038/nn.2959
- Tang, W., Wan, S., Yang, Z., Teschendorff, A. E., and Zou, Q. (2018). Tumor origin detection with tissue-specific miRNA and DNA methylation markers. *Bioinformatics* 34, 398–406. doi: 10.1093/bioinformatics/btx622
- Tao, Z., Li, Y., Teng, Z., and Zhao, Y. (2020). A method for identifying vesicle transport proteins based on LibSVM and MRMD. *Comput. Mathemat. Methods Med.* 2020:8926750.
- Thomson, J. P., and Meehan, R. R. (2016). The application of genome-wide 5-hydroxymethylcytosine studies in cancer research. *Epigenomics* 9, 77–91. doi: 10.2217/epi-2016-0122
- Timinskas, A., Butkus, V., and Janulaitis, A. (1995). Sequence motifs characteristic for DNA [cytosine-N4] and DNA [adenine-N6] methyltransferases. Classification of all DNA methyltransferases. *Gene* 157, 3–11. doi: 10.1016/0378-1119(94)00783-0
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. *arXiv [preprint]*.
- Wang, H., Ding, Y., Tang, J., and Guo, F. (2019). Identification of membrane protein types via multivariate information fusion with Hilbert-Schmidt Independence criterion. *Neurocomputing* 383, 257–269. doi: 10.1016/j.neucom.2019.11.103
- Wang, H., Tang, J., Ding, Y., and Guo, F. (2021). Exploring associations of non-coding RNAs in human diseases via three-matrix factorization with hypergraph-regular terms on center kernel alignment. *Brief. Bioinform.* Online ahead of print.

- Wang, J., Chen, S., Dong, L., and Wang, G. (2020). *CHTKC: a Robust and Efficient k-mer Counting Algorithm Based on a Lock-free Chaining Hash Table*. oxford: oxford university press.
- Wang, Y., Ding, Y., Tang, J., Dai, Y., and Guo, F. (2019). "CrystalM: a multi-view fusion approach for protein crystallization prediction," in *Proceedings of the IEEE/ACM Transactions on Computational Biology and Bioinformatics*, (CityplacePiscataway, StateNJ: IEEE).
- Wei, L., Chen, H., and Su, R. (2018a). M6APred-EL: a sequence-based predictor for identifying N6-methyladenosine sites using ensemble learning. *Mol. Therapy-Nucleic Acids* 12, 635–644. doi: 10.1016/j.omtn.2018.07.004
- Wei, L., Ding, Y., Su, R., Tang, J., and Zou, Q. (2018b). Prediction of human protein subcellular localization using deep learning. *J. Parallel Distrib. Comput.* 117, 212–217.
- Wei, L., Zhou, C., Chen, H., Song, J., and Su, R. (2018c). ACPred-FL: a sequence-based predictor using effective feature representation to improve the prediction of anti-cancer peptides. *Bioinformatics* 34, 4007–4016.
- Wei, L., Hu, J., Li, F., Song, J., Su, R., and Zou, Q. (2020). Comparative analysis and prediction of quorum-sensing peptides using feature representation learning and machine learning algorithms. *Brief. Bioinform.* 21, 106–119.
- Wei, L., Liao, M., Gao, Y., Ji, R., He, Z., and Zou, Q. (2014). Improved and promising identification of human micrornas by incorporating a high-quality negative set. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 11, 192–201. doi: 10.1109/tcbb.2013.146
- Wei, L., Luan, S., Nagai, L. A. E., Su, R., and Zou, Q. (2019a). Exploring sequence-based features for the improved prediction of DNA N4-methylcytosine sites in multiple species. *Bioinformatics* 35, 1326–1333. doi: 10.1093/bioinformatics/bty824
- Wei, L., Su, R., Luan, S., Liao, Z., Manavalan, B., Zou, Q., et al. (2019b). Iterative feature representations improve N4-methylcytosine site prediction. *Bioinformatics* 35, 4930–4937. doi: 10.1093/bioinformatics/btz408
- Wei, L., Su, R., Wang, B., Li, X., Zou, Q., and Gao, X. (2019c). Integration of deep feature representations and handcrafted features to improve the prediction of N-6-methyladenosine sites. *Neurocomputing* 324, 3–9. doi: 10.1016/j.neucom.2018.04.082
- Wei, L., Xing, P., Shi, G., Ji, Z., and Zou, Q. (2019d). Fast prediction of protein methylation sites using a sequence-based feature selection technique. *IEEE-ACM Trans. Comput. Biol. Bioinform.* 16, 1264–1273. doi: 10.1109/tcbb.2017.2670558
- Wei, L., Tang, J., and Zou, Q. (2017a). Local-DPP: an improved DNA-binding protein prediction method by exploring local evolutionary information. *Inform. Sci.* 384, 135–144. doi: 10.1016/j.ins.2016.06.026
- Wei, L., Wan, S., Guo, J., and Wong, K. K. L. (2017b). A novel hierarchical selective ensemble classifier with bioinformatics application. *Artif. Intell. Med.* 83, 82–90. doi: 10.1016/j.artmed.2017.02.005
- Wei, L., Xing, P., Zeng, J., Chen, J., Su, R., and Guo, F. (2017c). Improved prediction of protein-protein interactions using novel negative samples, features, and an ensemble classifier. *Artif. Intell. Med.* 83, 67–74. doi: 10.1016/j.artmed.2017.03.001
- Wion, D., and Casadesús, J. (2006). N6-methyl-adenine: an epigenetic signal for DNA-protein interactions. *Nat. Rev. Microbiol.* 4, 183–192. doi: 10.1038/nrmicro1350
- Wu, T. P., Wang, T., Seetin, M. G., Lai, Y., Zhu, S., Lin, K., et al. (2016). DNA methylation on N6-adenine in mammalian embryonic stem cells. *Nature* 532, 329–333. doi: 10.1038/nature17640
- Yang, H., Yang, W., Dao, F. Y., Lv, H., Ding, H., Chen, W., et al. (2020). A comparison and assessment of computational method for identifying recombination hotspots in *Saccharomyces cerevisiae*. *Brief. Bioinform.* 21, 1568–1580. doi: 10.1093/bib/bbz123
- Yang, C., Ding, Y., Meng, Q., Tang, J., and Guo, F. (2021). Granular multiple kernel learning for identifying RNA-binding protein residues via integrating sequence, and structure information. *Neural Comput. Appl.* 1–13. doi: 10.1007/s00521-020-05573-4
- Yu, H., and Dai, Z. (2019). SNNRice6mA: a deep learning method for predicting DNA N6-Methyladenine sites in rice genome. *Front. Genet.* 10:1071. doi: 10.3389/fgene.2019.01071
- Yuming, Z., Fang, W., and Liran, J. (2015). MicroRNA promoter identification in arabidopsis using multiple histone markers. *Biomed. Res. Int.* 2015:861402.
- Zhang, L., He, Y., Wang, H., Liu, H., Huang, Y., Wang, X., et al. (2019). Clustering count-based RNA methylation data using a nonparametric generative model. *Curr. Bioinform.* 14, 11–23. doi: 10.2174/1574893613666180601080008
- Zhang, Y., Kou, C., Wang, S., and Zhang, Y. (2019). Genome-wide differential-based analysis of the relationship between DNA methylation and gene expression in Cancer. *Curr. Bioinform.* 14, 783–792. doi: 10.2174/1574893614666190424160046
- Zhang, Z. Y., Yang, Y. H., Ding, H., Wang, D., Chen, W., and Lin, H. (2020). Design powerful predictor for mRNA subcellular location prediction in Homo sapiens. *Brief. Bioinform.* 22, 1–10.
- Zhao, X., Jiao, Q., Li, H., Wu, Y., and Wang, G. (2020). ECFS-DEA: an ensemble classifier-based feature selection for differential expression analysis on expression profiles. *BMC Bioinform.* 21:43. doi: 10.1186/s12859-020-3388-y
- Zhou, P., Shi, W., Tian, J., Qi, Z., and Xu, B. (2016). "Attention-based bidirectional long short-term memory networks for relation classification," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, (StateplaceBerlin: Association for Computational Linguistics).
- Zhu, T., Guan, J., Liu, H., and Zhou, S. (2019). RMDB: an integrated database of single-cytosine-resolution DNA methylation in *Oryza sativa*. *Curr. Bioinform.* 14, 524–531. doi: 10.2174/1574893614666190211161717
- Zou, Q., Li, J., Song, L., Zeng, X., and Wang, G. (2016). Similarity computation strategies in the microRNA-disease network: a survey. *Brief. Funct. Genom.* 15, 55–64.
- Zou, Q., Lin, G., Jiang, X., Liu, X., and Zeng, X. (2020a). Sequence clustering in bioinformatics: an empirical study. *Brief. Bioinform.* 21, 1–10.
- Zou, Q., Xing, P., Wei, L., and Liu, B. (2019). Gene2vec: gene subsequence embedding for prediction of mammalian N6-Methyladenosine sites from mRNA. *RNA* 25, 205–218. doi: 10.1261/rna.069112.118
- Zou, Y., Wu, H., Guo, X., Peng, L., Ding, Y., Tang, J., et al. (2020b). MK-FSVM-SVDD: a multiple kernel-based fuzzy SVM model for predicting DNA-binding proteins via support vector data description. *Curr. Bioinform.* Online ahead of print.
- Zuo, Y., Song, M., Li, H., Chen, X., Cao, P., Zheng, L., et al. (2020c). Analysis of the epigenetic signature of cell reprogramming by computational DNA methylation profiles. *Curr. Bioinform.* 15, 589–599. doi: 10.2174/1574893614666190919103752

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2021 Yang, Ye, Li and Wei. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.



GADTI: Graph Autoencoder Approach for DTI Prediction From Heterogeneous Network

Zhixian Liu^{1,2}, Qingfeng Chen^{3*}, Wei Lan³, Haiming Pan³, Xinkun Hao³ and Shirui Pan⁴

¹ School of Medical, Guangxi University, Nanning, China, ² School of Electronics and Information Engineering, Beibu Gulf University, Qinzhou, China, ³ School of Computer, Electronic and Information, Guangxi University, Nanning, China,

⁴ Department of Data Science and AI, Monash University, Melbourne, VIC, Australia

OPEN ACCESS

Edited by:

Xiangxiang Zeng,
Hunan University, China

Reviewed by:

Khanh N. Q. Le,
Taipei Medical University, Taiwan
Yusen Zhang,
Shandong University, China

*Correspondence:

Qingfeng Chen
qingfeng@gxu.edu.cn

Specialty section:

This article was submitted to
Computational Genomics,
a section of the journal
Frontiers in Genetics

Received: 08 January 2021

Accepted: 12 March 2021

Published: 09 April 2021

Citation:

Liu Z, Chen Q, Lan W, Pan H, Hao X
and Pan S (2021) GADTI: Graph
Autoencoder Approach for DTI
Prediction From Heterogeneous
Network. *Front. Genet.* 12:650821.
doi: 10.3389/fgene.2021.650821

Identifying drug–target interaction (DTI) is the basis for drug development. However, the method of using biochemical experiments to discover drug–target interactions has low coverage and high costs. Many computational methods have been developed to predict potential drug–target interactions based on known drug–target interactions, but the accuracy of these methods still needs to be improved. In this article, a graph autoencoder approach for DTI prediction (GADTI) was proposed to discover potential interactions between drugs and targets using a heterogeneous network, which integrates diverse drug-related and target-related datasets. Its encoder consists of two components: a graph convolutional network (GCN) and a random walk with restart (RWR). And the decoder is DistMult, a matrix factorization model, using embedding vectors from encoder to discover potential DTIs. The combination of GCN and RWR can provide nodes with more information through a larger neighborhood, and it can also avoid over-smoothing and computational complexity caused by multi-layer message passing. Based on the 10-fold cross-validation, we conduct three experiments in different scenarios. The results show that GADTI is superior to the baseline methods in both the area under the receiver operator characteristic curve and the area under the precision–recall curve. In addition, based on the latest Drugbank dataset (V5.1.8), the case study shows that 54.8% of new approved DTIs are predicted by GADTI.

Keywords: drug–target interaction prediction, network embedding, graph convolutional network, autoencoder, random walk, heterogeneous network

INTRODUCTION

The drug acts on the target protein, thereby affecting the expression of the target protein to achieve the therapeutic effect on the disease. Therefore, finding drug–target interactions is the basis of drug development. The research and development of innovative drugs often requires billions of dollars and more than a decade of work, and usually ends in failure. Hence, it is an important choice for pharmaceutical companies to discover potential drug–target interactions (DTIs) by using the known DTIs. The properties of existing drugs are familiar to people, and their safety is guaranteed. However, there are some limits in both coverage and throughput of biochemical experiments to identify new DTIs. Consequently, the prediction of DTIs using computational methods has attracted extensive attention.

Early computational methods were mainly based on drug–drug similarity and target–target similarity, or the features of drugs and targets. Some of the methods based on similarity first calculate the similarity between the drug pairs (e.g., chemical structure similarity) and the similarity between the target pairs (e.g., protein sequence similarity), and then use the known DTIs to score the unknown DTI (Cheng et al., 2012; Mei et al., 2013; Wang et al., 2014). Other similarity-based methods process a random walk on the network composed of multiple data sources, such as drug–drug interactions, target–target interactions, and DTIs to obtain the similarity between nodes to predict new DTIs (Chen et al., 2012; Seal et al., 2015). In the methods based on features, both the drugs and the targets are represented as fixed-length feature vectors, and the known drug–target pairs are divided into positive and negative categories. Then the DTI prediction is transformed into a binary classification problem. Machine learning methods such as support vector machines, random forests, and conditional random fields can be directly used for prediction (Nagamine et al., 2009; Lan et al., 2016; Olayan et al., 2018; Chen et al., 2019; Shi et al., 2019).

In recent years, network embedding methods (Perozzi et al., 2014) have shown excellent performance in network data analysis (Cai et al., 2017), and have been introduced into DTI prediction (Su et al., 2018; Bagherian et al., 2020; Liu et al., 2020). Network embedding is also known as graph embedding. In network embedding, nodes such as drugs and targets can all be converted into low-dimensional vectors that represent their features and can be directly used for DTI prediction. The main methods of network embedding include matrix factorization, random walk, and deep learning.

A multiple similarities collaborative matrix factorization model (Zheng et al., 2013) was proposed to predict DTI. It incorporates anatomical therapeutic chemical similarity and chemical structure similarity of drugs, as well as genomic sequence similarity, gene ontology (GO) similarity, and protein–protein interaction (PPI) network similarity of targets. A combination of these similarity matrices was used to approximate the drug feature matrix *D* and the target feature matrix *T*, and then the inner product between *D* and *T* was utilized to approximate the DTI matrix. TriModel (Mohamed et al., 2019) uses the drug-related knowledge graph to find potential DTIs. It learns the feature vectors of nodes in the knowledge graph through tensor decomposition. These vectors are used to determine whether the drug and the target interact. Meanwhile, DTINet (Luo et al., 2017) first uses the random walk to obtain the low-dimensional feature vector of each drug and protein, projects the drug vector and protein vector into the same space, and then discovers new interactions through matrix completion. Encouraged by the DeepWalk (Perozzi et al., 2014) model, some researchers have combined the random walk with shallow neural networks (Zong et al., 2017, 2019; Zhu et al., 2018). These methods first construct a heterogeneous network based on multiple data sources, and then apply DeepWalk, node2vec (Grover and Leskovec, 2016), and other algorithms to the network to obtain the embedding vectors of drug nodes and target nodes. NeoDTI (Wan et al., 2019) uses a deep learning method based on neighborhood information aggregation to

discover new DTIs. It aggregates neighbor information based on edge types in heterogeneous networks. Then, the feature vector of the node is used to reconstruct the original network. There are also several studies based on drug structure and protein sequence (Wen et al., 2017; Karimi et al., 2019; Öztürk et al., 2019). Starting from the chemical structure and protein sequence of compounds, deep learning methods are then employed to predict drug–target binding affinity.

Matrix factorization methods can capture the global structure of the network, but its space complexity increases rapidly as the network scale increases. Random walk methods are more efficient because they usually gather only local features. Deep learning methods are outstanding in DTI prediction because it can discover hidden features and associations from multi-source heterogeneous network, and it is easy to integrate externally associated data of drugs and targets (e.g., GO) to improve performance. However, deep learning is computationally expensive and time-consuming. Among the deep learning methods, the graph convolutional network (GCN)-based message passing (also known as neighborhood information aggregation) algorithms have recently attracted special attention due to their flexibility and good performance (Kearnes et al., 2016; Ying et al., 2018; Wan et al., 2019). The GCN algorithms usually only consider the neighborhood with a short distance (e.g., the first-order neighborhood) because large distances will lead to over-smoothing, which degrades performance and increases computational complexity. However, the short distance easily leads to insufficient information about the neighborhood of the node (Li et al., 2018; Xu et al., 2018).

In this article, we propose a graph autoencoder approach for DTI prediction using a heterogeneous network (GADTI), which combines a graph convolutional network, matrix factorization, and random walk. GADTI first constructs a heterogeneous network that integrates eight data sources related to drugs and targets. Then, it runs a graph autoencoder model on the network to discover new DTIs. The encoder of the graph autoencoder includes two components: a GCN and a random walk with restart (RWR). The GCN component aggregates the first-order neighborhood information of each node and uses it to subsequently update the feature vector of nodes. The RWR component propagates the influence of nodes over the heterogeneous network. Through this, we obtain the embedding vectors of nodes, which are sent to the decoder. We use the matrix factorization model DistMult (Yang et al., 2015) to reconstruct the original heterogeneous network from the embedding vectors of nodes. Through the combination of GCN and RWR, GADTI can provide nodes with more information through a larger neighborhood while avoiding the over-smoothing and computational complexity caused by multi-layer message passing. The experimental results demonstrate that our approach is effective and efficient to predict potential DTIs.

MATERIALS AND METHODS

Dataset

We adopted a dataset used in previous studies (Luo et al., 2017; Wan et al., 2019). It consists of eight networks, including

TABLE 1 | Sources of datasets and their statistical information.

(a) Statistical information of nodes		
Node Type	Count	
Drug	708	
Targets	1,512	
Disease	5,603	
Side effect	4,192	
(b) Statistical information and source of edges		
Edge	Count	Data Source
drug–target interaction	1,923	DrugBank v3.0 (Knox et al., 2011)
Drug–drug interaction	10,036	DrugBank v3.0 (Knox et al., 2011)
Protein–protein	7,363	HPRD Release 9 (Keshava Prasad et al., 2009)
Drug–disease	199,214	Comparative Toxicogenomics Database (Davis et al., 2013)
Drug side effect	80,164	SIDER Version 2 (Kuhn et al., 2010)
Protein–disease	1,596,745	Comparative Toxicogenomics Database (Davis et al., 2013)
Drug structure similarity	*	Based on Morgan fingerprints (Rogers and Hahn, 2010)
Protein sequence similarity	*	Based on Smith–Waterman scores (Smith and Waterman, 1981)
Total	1,895,445	

*This edge is not counted because all node pairs are connected.

four types of nodes (drugs, targets, diseases, and side effects) and eight types of edges (drug–drug interaction, DTI, drug–disease association, drug–side effects association, protein–protein interaction, protein–disease association, drug chemical structure similarity, and protein sequence similarity). These data come from public databases such as DrugBank, HPRD, and SIDER. The weights of edges in all networks are non-negative. Furthermore, only the drug chemical structure similarity and the protein sequence similarity are real-valued, and thus represent drug–drug chemical structure similarity scores and protein–protein sequence similarity scores. The others are binary values indicating whether there is an interaction or association between nodes. **Table 1** lists the sources and statistics of these data.

Spatial-Based Graph Convolutional Network

Most recent network embedding methods are based on the GCN, especially spatial-based GCN. These methods define convolution on graph as neighborhood information aggregation. They generate embeddings for nodes by aggregating the local neighborhood of the nodes instead of the entire network, which is regarded as a message passing mechanism.

A typical spatial-based GCN method includes two phases. In the initialization phase, it generates an initial vector based

on the features of each node. If all the nodes in the network have no features, a one-hot vector is assigned to each node and a neural network is used to generate the initial vector. In the second phase, the vectors of nodes are updated by a combination of aggregated neighborhood vectors and the previous vectors of the nodes. These updates can be done through neural networks or linear transformations. The embedding vector of a node is a function of its neighborhood (including the node itself). This process looks similar to the receptive field of the convolution kernel in image processing, so it is called GCN. After one aggregation, the embedding vector of the node contains the feature information of its first-order neighbors. If we repeat this aggregation process K times, the embedding vector of the node can capture the feature information of its K -order neighbors. In the spatial-based GCN, the information of a node is first passed to its first-order neighbors, and then propagated to higher-order neighbors through edges on the network. Therefore, these methods are also called message passing methods. The process of graph convolution operation is summarized as follows:

$$\begin{aligned}
 a_v^{(n)} &= \text{AGGREGATE}^{(n)} \left(\left\{ h_u^{(n-1)} : u \in \mathcal{N}(v) \right\} \right), h_v^{(n)} \\
 &= \text{UPDATE}^{(n)} \left(\left\{ h_v^{(n-1)}, a_v^{(n)} \right\} \right)
 \end{aligned} \quad (1)$$

where $\text{AGGREGATE}()$ and $\text{UPDATE}()$ are functions to aggregate neighborhood information and update node vectors, respectively; u, v are nodes; $a_v^{(n)}$ is the aggregated feature information of v at the n -th iteration; $\mathcal{N}(v)$ indicates the neighborhood of v ; and $h_v^{(n)}$ is the embedding vector of v at the n -th iteration. After the iteration, we obtain $h_v^{(K)}$, which represents the features of v and can be directly used for node-level tasks such as node similarity calculations, node classification, and link prediction.

Graph Autoencoder

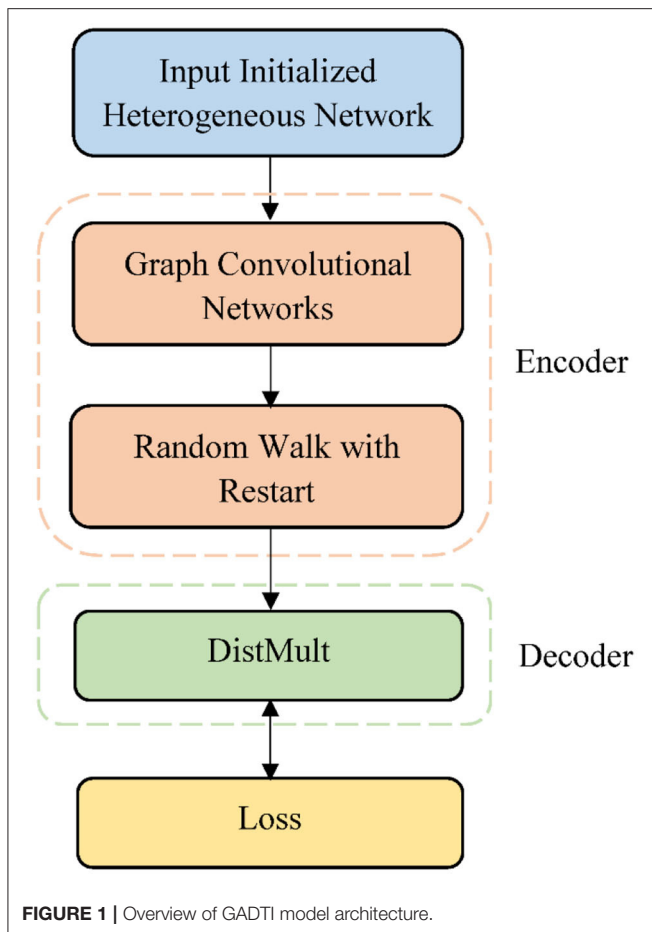
The graph autoencoder takes the network and the feature vectors of the node as input to generate a low-dimensional embedding vector of the node or the entire network.

Unlike traditional autoencoders, the encoder of a graph autoencoder is usually a GCN and its variants, and the decoder can be an inner product (Kipf and Welling, 2016; Pan et al., 2018) or matrix factorization (Zitnik et al., 2018; Lan et al., 2020). Generative adversarial networks (GANs) (Goodfellow et al., 2014) and attention mechanisms have also been applied to graph autoencoders (Ma et al., 2018; Pan et al., 2018; Jin et al., 2019). For heterogeneous graphs containing multiple edge types, the encoder aggregates neighbor features one by one according to the edge type, and then merges them to obtain the embedding vectors of the nodes (Gligorić et al., 2018; Ma et al., 2018; Zitnik et al., 2018).

GADTI

The data related to drugs and targets are represented in the form of a network, and the DTI prediction is then transformed into a link prediction of the network.

Definition 1 Network $G = (V, R)$, where $v \in V$ and $r \in R$ are nodes and edges, respectively.



Given a network G , v_d and v_t are the drug node and target node, respectively. Our goal is to determine whether the unknown edge $r_{dt} = (v_d, v_t)$ exists, or how likely it is to exist. To this end, we developed an end-to-end framework GADTI based on the graph autoencoder to discover new DTIs. This approach combines a graph convolutional network, matrix factorization, and random walk. GADTI first integrates multiple data sources to build a heterogeneous network, and then conducts prediction through a graph autoencoder model. As shown in **Figure 1**, GADTI has two main components:

- An encoder: a GCN followed by an RWR, which produces embeddings for nodes in G ;
- A decoder: a matrix factorization model using these embeddings to predict DTIs.

Encoder

The encoder consists of a GCN and an RWR. The GCN is used to aggregate first-order neighbor information to update node representation. Then, an RWR on the entire heterogeneous network allows the influence of nodes to spread far away so that we can obtain the final embedding vector. This approach can provide more information to nodes

through a larger neighborhood while avoiding the over-smoothing and computational complexity caused by multi-layer convolutional networks.

Aggregation by GCN

In this stage, only the first-order neighborhood of the node is considered. For each node, we first group its first-order neighbors according to the type of edge. Then, for each neighbor group, a neighborhood aggregation operation is performed to aggregate information. Finally, the neighbor information of different groups is accumulated and concatenated with the previous embedding vector of the node, and then sent to the neural network to generate a new embedding vector. The process of aggregating and updating are defined as follows:

$$\begin{aligned} a_v^r &= \sum_{u \in \mathcal{N}_r(v)} \frac{1}{c_r^v} \sigma(W_r^0 h_u^0 + b_r), h_v^* \\ &= \text{MEAN} \left(\{h_v^0\} \cup \{a_v^r : r \in R\} \right) \end{aligned} \quad (2)$$

where a_v^r refers to the aggregated neighborhood information of v related to edge type r , $h_v^0 \in \mathbb{R}^d$ refers to the initial embedding vector of v , d denotes the dimension of vector, R indicates the set of edge types, $\mathcal{N}_r(v)$ are the neighbors of v related to edge type r , σ is a non-linear activation function, and $W_r^0 \in \mathbb{R}^{d \times d}$ and $b_r \in \mathbb{R}^d$ are edge-type specific parameter matrix and bias terms used to aggregate neighborhood information, respectively. c_r^v is a normalization constant that we choose to be $c_r^v = |\mathcal{N}_r(v)|$. $\text{MEAN}()$ is an element-wise mean operator, h_v^* is the updated embedding vector of v .

Figure 2 shows a small example of the network. Drug node $D1$ is associated with two diseases and one side effect, as well as targets two proteins, and interacts with three other drugs. The bold dotted line indicates the similarity between drugs.

The process of the encoder is provided in **Figure 3**. Multiple different single-layer neural networks (SLNs) are used in the encoder according to edge types. We take the drug node $D1$ in **Figure 2** as an example. Since there are five types of edges connected to $D1$, there will be five SLNs to aggregate neighbor information of corresponding edge types. The mean operator is chosen as the aggregation function, to perform an element-wise mean of the vectors in $\{h_v^0\} \cup \{a_v^r : r \in R\}$. It results in the new node embedding vector h^* . $\text{Relu}(x) = \max(0, x)$ is selected as the element-wise activation function. A projection with learnable parameters is employed to initialize h^0 .

Propagation by RWR

The multi-hop neighborhood information aggregation implemented by multi-layer convolution often leads to over-smoothing. The aforementioned GCN only considers the one-hop graph structure, which causes the multi-hop information of the node to be underutilized. In order to solve this problem, we introduce an RWR, which spreads the influence of nodes to other nodes that are not directly adjacent through a walk on the heterogeneous network. The introduction of multi-hop information extends the range of information aggregation from the first-order neighborhood to the high-order neighborhood, which is equivalent to increasing

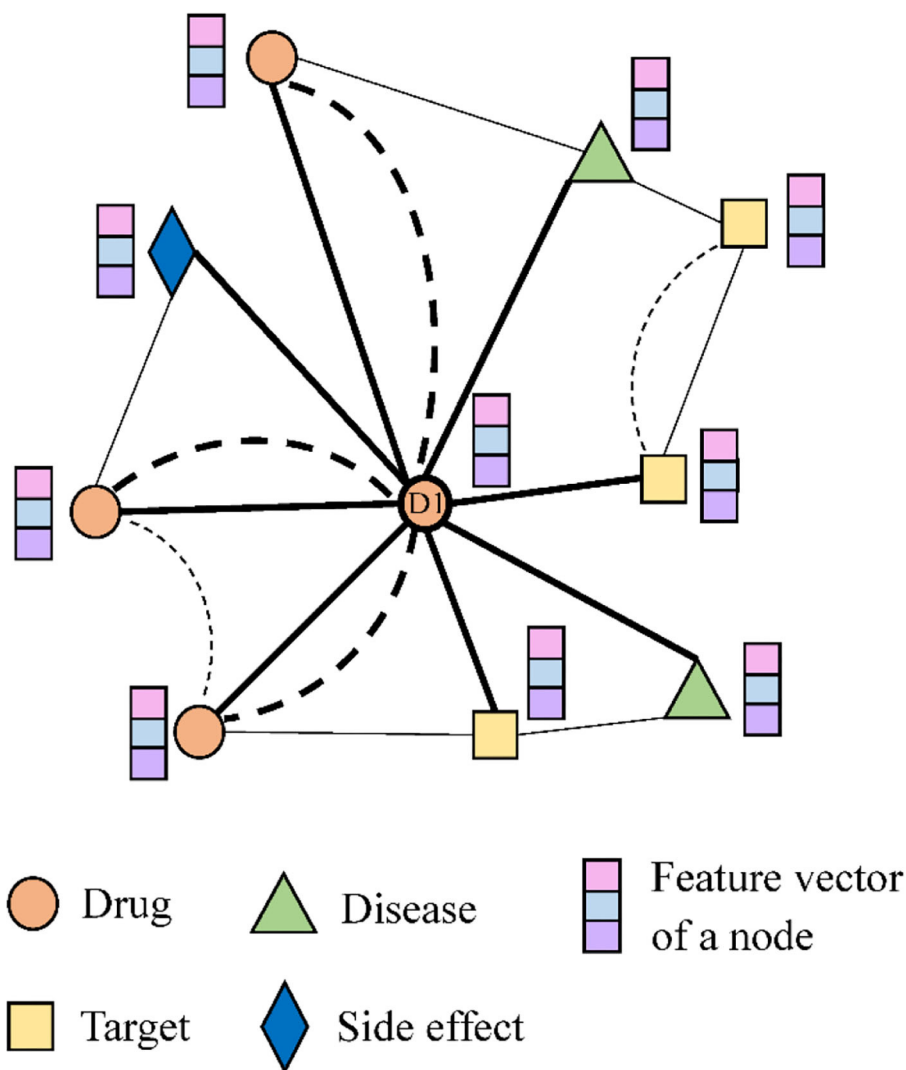


FIGURE 2 | An small example of the heterogeneous network.

the receptive field of convolution, thereby realizing long-range message passing.

Assuming that the transition matrix of the heterogeneous network is A and the restart probability is α , the RWR is defined as follows (Tong et al., 2008):

$$A_{ppr} = \alpha(I - (1 - \alpha)A)^{-1} \quad (3)$$

where I is the identity matrix, and $A_{ppr}(u, v)$ indicates the influence of node u on node v .

According to Equation (3), we can spread node information over long distances to get the final node embedding vector:

$$H^1 = \alpha(I - (1 - \alpha)A)^{-1}H^* \quad (4)$$

where H^* is the node embedding vector matrix obtained by the aforementioned convolution operation.

Since the time complexity of Equation (4) is $\mathcal{O}(n^2)$, when the network scale is large, it may be expensive. Therefore, we introduce the iterative form of Equation (4):

$$Z^0 = H^*, Z^{k+1} = (1 - \alpha)AZ^k + \alpha Z^0 \quad (5)$$

It is easy to prove that $\lim_{K \rightarrow \infty} Z^K = \alpha(I - (1 - \alpha)A)^{-1}H^*$.

Because all drug node pairs have edges of chemical structure similarity, there may be two edges between some drug node pairs. The same is true for target node pairs, and will bring inconvenience to the random walk. To simplify the problem, we delete the edges representing the similarity of drug structure and protein sequence from the heterogeneous network. That is, the graph convolution operates on a complete heterogeneous network whereas the random walk is only performed on a sub-network of the complete network.

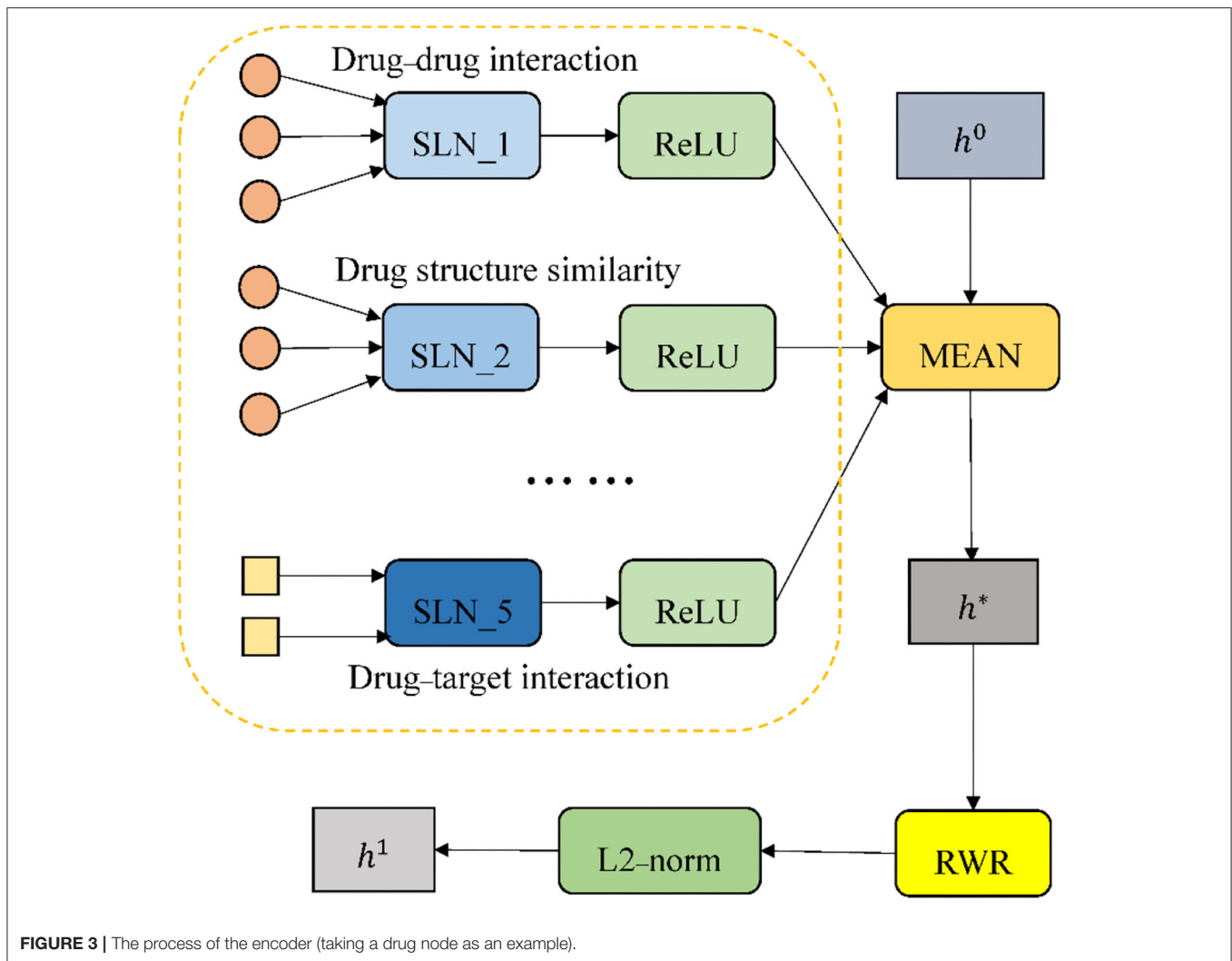


FIGURE 3 | The process of the encoder (taking a drug node as an example).

Decoder

While encoder maps each node in the heterogeneous network to a real-valued embedding vector, the decoder reconstructs the original network from the embedding vectors. The decoder is essentially a scoring function $s(u, r, v) : \mathbb{R}^d \times \mathcal{R} \times \mathbb{R}^d \rightarrow \mathbb{R}$, used to score the triplets (u, r, v) so that we can evaluate the probability of edge r existing between u and v , where u and v are nodes, and r is a certain type of edge.

In our experiments, we use DistMult (Yang et al., 2015) as the decoder, which is known to perform well on standard link prediction benchmarks. The scoring function is:

$$s(u, r, v) = e_u^T M_r e_v \quad (6)$$

where e_u and e_v are the embedding vectors of u and v , respectively. e_u^T is the transpose of e_u , and $M_r \in \mathbb{R}^{d \times d}$ is an edge-type specific diagonal matrix.

In terms of Equation (6), we can reconstruct the original networks. Take the reconstruction of a DTI network as

an example:

$$DTI_{re} = V_{drug}^T M_{DTI} V_{protein} \quad (7)$$

where V_{drug} and $V_{protein}$ are the matrices of drug embedding vectors and target embedding vectors, respectively, and M_{DTI} is the diagonal matrix used to reconstruct the DTI network.

Training

The loss of network reconstruction is as follows:

$$\begin{aligned} \mathcal{L}_{re} &= \sum_{r \in \mathcal{R}} \sum_{i,j} (Q_{ij})^2, Q \\ &= P(\text{Network}_{original}^r - \text{Network}_{reconstruction}^r) \end{aligned} \quad (8)$$

where $\text{Network}_{original}^r$ and $\text{Network}_{reconstruction}^r$ are the original network with edge type r and the corresponding reconstructed network, respectively. P is a mask matrix where $P_{ij} = 1$ indicates that the element in the i -th row and j -th column of $\text{Network}_{original}^r$ appears in the training set, otherwise it does not occur. Q is a matrix that stores the difference between the predicted

value and the ground truth in the training set. We further add the regularization term of the weight coefficient to obtain the objective function:

$$\begin{aligned}\mathcal{L}_{re} &= \sum_{r \in R} \sum_{i,j} (Q_{ij})^2 + \lambda \sum_w w^2, Q \\ &= P(\text{Network}_{original}^r - \text{Network}_{reconstruction}^r)\end{aligned}\quad (9)$$

Our optimization goal is to minimize Equation (9), where $\sum_w w^2$ is the sum of the squares of all the weights, and λ is an adjustment coefficient. In GADTI, there are three trainable parameters: (1) four matrices for initializing node vectors, i.e., W^{drug} , $W^{disease}$, $W^{protein}$ and $W^{sideeffect}$; (2) 12 edge-type-specific neural network weight matrices W_r^0 for aggregating neighborhood information; and (3) 8 edge-type-specific diagonal matrices M_r used to reconstruct the networks.

We adopted the same sampling strategy and dataset division strategy as Wan et al. (2019). For the DTI network, the sample pair with an edge connection is regarded as the positive sample, and the sample pair without a connection is the negative sample. We randomly collect 10 negative samples for each positive sample to form the DTI dataset used by the model. Ten-fold cross-validation (Le et al., 2019) was used for performance evaluation. In each fold, the DTI dataset is randomly divided into three independent parts: training set, validation set and test set, with ratios of 0.855, 0.045, and 0.1 respectively. The training set of GADTI is composed of the training set of DTIs and other seven datasets. In each iteration, we update the model parameters on the training set, and then evaluate the model on the validation set. If the new model parameters show better performance on the validation set than before, the test set will be used to test the generalization ability of the model.

In addition to L2 regularization, early stopping is introduced to alleviate over-fitting. If the performance of the model on the validation set does not increase for n iterations, it can be considered that overfitting has occurred, so the training will stop early. Adaptive moment estimation algorithm (Adam) (Kingma and Ba, 2015) is selected to minimize the objective function. The dimension of embedding vector and the learning rate are set to 1,000 and 0.001, respectively, according to independent experiments. Our code runs on PyTorch V1.7 and DGL V0.5.

RESULTS

Performance Evaluation

We used 10-fold cross-validation to test the performance of our algorithm, and stratified sampling to ensure that the proportion of samples in each category in the training set and test set were the same as in the original dataset. The area under the receiver operator characteristic curve (AUROC) (Le, 2019) and the area under the precision-recall curve (AUPRC) were chosen to evaluate the performance of our approach and baseline methods.

The receiver operator characteristics (ROC) curve is suitable for evaluating the overall performance of the classifier because it takes both positive and negative samples into consideration (Le et al., 2020). However, class imbalance often occurs in actual datasets. For example, in a DTI network, the number of negative

samples is much larger than that of positive samples. In this case, the ROC curve presents an overly optimistic estimate of the effect. Conversely, both indicators of the precision-recall (PR) curve focus on positive samples. In the class imbalance cases, people are mainly concerned with positive samples, and thus the PR curve is widely considered to be better than the ROC curve. We use both AUROC and AUPRC. The larger the value of AUROC and AUPRC, the better the performance of the method.

Comparison With Baseline Methods

To evaluate the performance of GADTI, we compared it with four popular computational methods: MSCMF (Zheng et al., 2013), TL_HGBI (Wang et al., 2014), DTINet (Luo et al., 2017), and NeoDTI (Wan et al., 2019). These methods all predict DTIs from a heterogeneous network composed of multiple datasets. MSCMF uses matrix factorization methods and linear combinations of matrices to achieve prediction. TL_HGBI first establishes a three-layer heterogeneous network consisting of disease, drug, and protein data, and then uses an iterative strategy for drug repositioning. Meanwhile, DTINet focuses on learning low-dimensional vector representations of features that can accurately interpret the topological characteristics of each node in a heterogeneous network, and then makes predictions based on these representations through a vector space projection scheme. NeoDTI is close to the non-random walk version of GADTI. It first aggregates neighborhood information, and then reconstructs the network through two bilinear transformations. We run all five methods on the same dataset and implement three rounds of 10-fold cross-validation to compare their performance. The hyperparameters used in the baseline methods are the same as those in NeoDTI.

When the ratio of positive sample to negative sample is 1:10, the results of GADTI and the baseline methods are shown in **Figures 4, 5**. We observe that GADTI has an AUROC value of 0.9582, which is higher than those of NeoDTI (0.9509), DTINet (0.9208), TL_HGBI (0.8914), and MSCMF (0.8355). Meanwhile, in terms of AUPRC, which is more suitable for the current class imbalance case, GADTI is also better than all the baseline methods. Our approach slightly outperforms the second-best method (0.73% in terms of AUROC and 0.79% in terms of AUPRC).

Some DTI prediction methods based on machine learning include all unknown DTIs (treated as negative examples) in the training. To have a better comparison, we did additional test in this scenario. Experiment shows that GADTI still achieve the best performance, with an AUROC of 0.9369 and an AUPRC of 0.6205, and it stays ahead by a bigger margin. We notice that the AUROC values of all methods range from 0.8504 to 0.9369, but the AUPRC values range from 0.0312 to 0.6205, which is a large gap. **Figure 6** shows the experimental results of the dataset including all unknown DTIs.

The dataset in section Dataset contains homologous proteins or structurally similar drugs, which reduces the difficulty of predicting their interactions. In other words, the good performance of the DTI prediction method may come from a simple algorithm rather than a well-designed algorithm. Therefore, we carried out an additional experiment which is

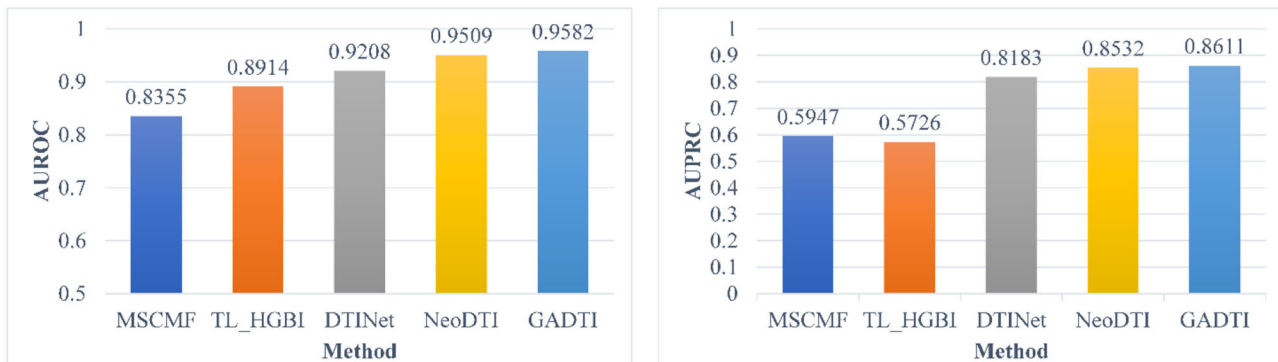


FIGURE 4 | Comparison between MSCMF, TL_HGBI, DTINet, NeoDTI, and GADTI in terms of AUROC and AUPRC based on 10-fold cross-validation (#positive: #negative = 1:10).

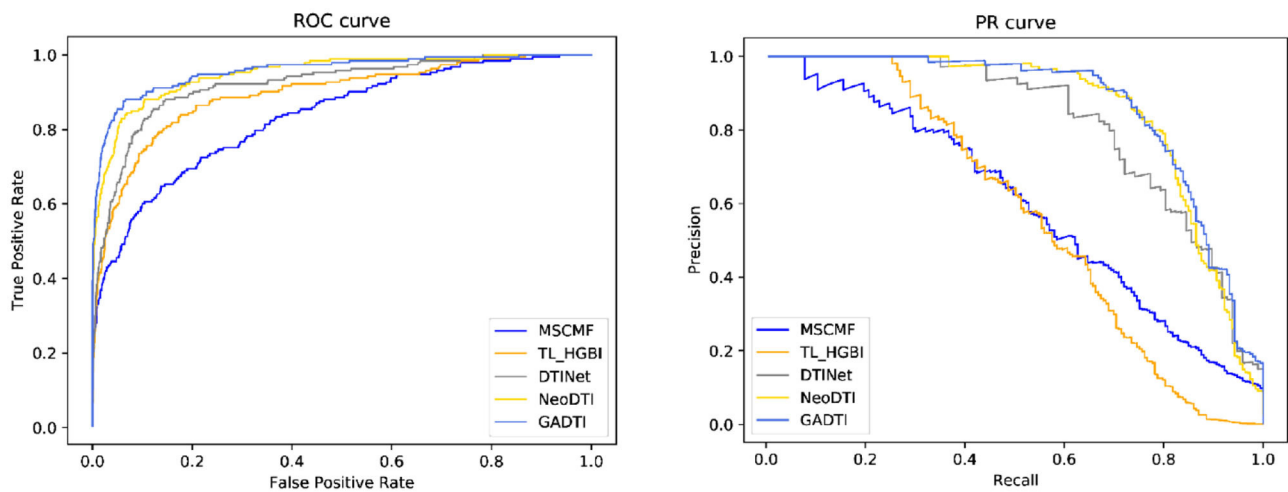


FIGURE 5 | The ROC curves and PR curves of different methods (#positive: #negative = 1:10).

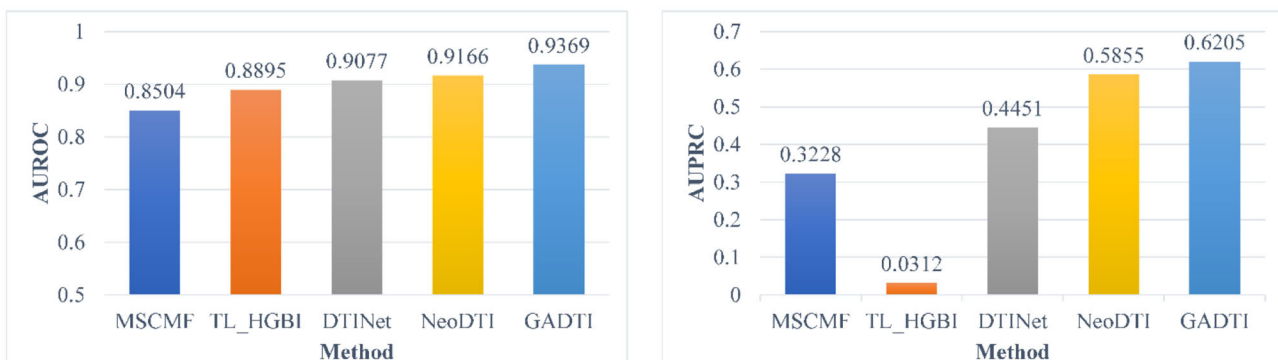


FIGURE 6 | Comparison between different methods in terms of AUROC and AUPRC based on 10-fold cross-validation (all unknown pairs were treated as negative examples).

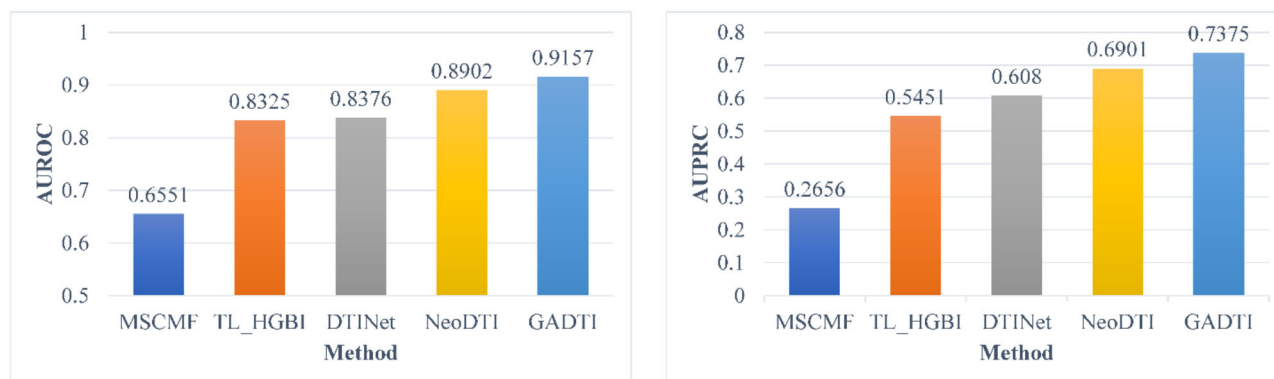


FIGURE 7 | Comparison between different methods in terms of AUROC and AUPRC based on 10-fold cross-validation (#positive: #negative = 1:10, DTIs with similar drugs or targets were removed).

TABLE 2 | Hit numbers of GADTI in different configurations.

	<i>m</i> = 10	<i>m</i> = 20	<i>m</i> = 30	<i>m</i> = 40
Configuration A: #positive: #negative = 1:10	211	406	508	570
Configuration B: all unknown pairs were treated as negative examples	291	402	475	523
Configuration C: #positive: #negative = 1:10, and DTIs with similar drugs or targets were removed	149	265	351	422

the same as in Wan et al. (2019): the DTIs with homologous proteins (similarities > 0.4) or similar drugs (similarities > 0.6) were removed. **Figure 7** shows the experimental results, where the ratio of positive samples to negative samples is 1:10. GADTI greatly outperforms the second-best method (2.55% in terms of AUROC and 4.74% in terms of AUPRC).

Case Study

To evaluate the prediction performance, we downloaded the latest approved DTI dataset (V5.1.8, 2021-01-03) from DrugBank to verify the DTIs predicted by GADTI. We generated a set *DTI_newly*, which contained 1,040 new DTIs related to our original dataset from the latest DTI dataset. For each fold, top 40 potential DTIs were selected for each drug based on their predicted scores. Because the experiment used three rounds of 10-fold cross-validation, we obtained 30 tables with 708 rows and 40 columns. Each row represented the potential DTIs of a drug. The DTIs of each drug were then sorted in descending order by the number of occurrences. A predicted DTIs set *DTI_pre* was generated by selecting the top *m* DTIs for each drug. Finally, the number of DTIs (hit number) in the intersection of *DTI_newly* and *DTI_pre* was calculated to verify the reliability of the prediction. The results are shown in **Table 2**.

We observe that 54.8% of all new DTIs are predicted by GADTI in case of *m* = 40.

DISCUSSION

Finding novel DTI pairs is of great significance for drug development. However, biochemical experiments are very costly and time-consuming. Therefore, computational methods have attracted much attention recently because they can quickly and cheaply evaluate potential DTIs. Early DTI prediction studies are mainly divided into two categories: (a) inferring based on drug similarity and target similarity (Chen et al., 2012; Cheng et al., 2012; Mei et al., 2013; Wang et al., 2014; Seal et al., 2015); and (b) binary prediction based on drug feature and target feature (Nagamine et al., 2009; Lan et al., 2016; Olayan et al., 2018; Chen et al., 2019; Shi et al., 2019). The GADTI approach proposed in this paper also utilizes similarity data and the features of drugs and targets, which are represented in vectors. However, unlike previous studies, the network embedding method and the graph autoencoder framework are introduced to learn the embedding feature vectors of drugs and targets from multi-source heterogeneous networks for predicting unknown DTIs.

We use AUROC and AUPRC to evaluate the performance of GADTI and the baseline methods. The results show that GADTI greatly outperforms the other methods in three different scenarios. Only NeoDTI achieves comparable results under the situation where the ratio of positive sample to negative sample is 1:10 (**Figure 4**). This may be because NeoDTI also adopts GCN for aggregating and updating. In case study, GADTI accurately predicted 54.8% of the new DTIs (**Table 2**). We observe that the hit numbers of configuration B are less than those of configuration A, in case of *m* = 20, 30, and 40. However, the gap decreases with the decrease of *m*. We can see that in case of *m* = 10, the result is just reversed: the hit number of configuration B is much greater than that of configuration A. A reasonable inference is that configuration B, all unknown pairs are treated as negative examples, can make the ranking of potential DTIs more accurate. As a result of our experiments we conclude that, compared with baseline methods, GADTI is more reliable and effective in discovering potential DTIs. Hence, it can be used to identify new targets for existing drugs.

The reason why GADTI performs well is that it aggregates multi-hop neighborhood information and avoids over-smoothing. First of all, GADTI uses a GCN to aggregate first-order neighbor information from heterogeneous networks to update node representation. Then, an RWR is carried out on the whole network to spread the influence of nodes. The combination of the GCN and RWR introduces multi-hop information for node feature updating. It extends the scope of information aggregation from the first-order neighborhood to the higher-order neighborhood, which is equivalent to increasing the receptive field of convolution, thereby realizing long-range message passing.

Although GADTI has made outstanding achievements in DTI prediction, it still has room for improvement. For new nodes of drugs or targets that did not appear during training, GADTI cannot directly predict their interaction with known nodes, that is, it needs to restart training to make predictions. In addition, GADTI cannot predict isolated new nodes that are not associated with known drugs or target nodes. In future research, we will introduce node features and improve the model structure to try to solve these two problems.

REFERENCES

- Bagherian, M., Sabeti, E., Wang, K., Sartor, M. A., Nikolovska-Coleska, Z., and Najarian, K. (2020). Machine learning approaches and databases for prediction of drug-target interaction: a survey paper. *Brief. Bioinform.* 22, 247–269. doi: 10.1093/bib/bbz157
- Cai, H., Zheng, V. W., and Chang, C. C. (2017). A comprehensive survey of graph embedding: problems, techniques and applications. *IEEE Trans. Knowl. Data Eng.* 30, 1616–1637. doi: 10.1109/TKDE.2018.2807452
- Chen, Q., Lai, D., Lan, W., Wu, X., Chen, B., Chen, Y. P., et al. (2019). ILDMSE: inferring associations between long non-coding RNA and disease based on multi-similarity fusion. *IEEE/ACM Trans. Comput. Biol. Bioinform.* doi: 10.1109/TCBB.2019.2936476. [Epub ahead of print].
- Chen, X., Liu, M. X., and Yan, G. Y. (2012). Drug-target interaction prediction by random walk on the heterogeneous network. *Mol. Biosyst.* 8, 1970–1978. doi: 10.1039/c2mb00002d
- Cheng, F., Liu, C., Jiang, J., Lu, W., Li, W., Liu, G., et al. (2012). Prediction of drug-target interactions and drug repositioning via network-based inference. *PLoS Comput. Biol.* 8:e1002503. doi: 10.1371/journal.pcbi.1002503
- Davis, A. P., Murphy, C. G., Johnson, R., Lay, J. M., Lennson-Hopkins, K., Saraceni-Richards, C., et al. (2013). The comparative toxicogenomics database: update 2013. *Nucleic Acids Res.* 41, D1104–D1114. doi: 10.1093/nar/gks994
- Gligorijevic, V., Barot, M., and Bonneau, R. (2018). deepNF: Deep network fusion for protein function prediction. *Bioinformatics* 33, 3873–3881. doi: 10.1093/bioinformatics/bty440
- Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). “Generative adversarial nets,” in *Proceedings of the 27th International Conference on Neural Information Processing Systems*, Vol. 2 (Montreal, QC: MIT Press).
- Grover, A., and Leskovec, J. (2016). “node2vec: Scalable Feature Learning for Networks,” in *Conference on Knowledge Discovery and Data Mining*, eds. B. Krishnapuram, M. Shah, A. J. Smola, C. C. Aggarwal, D. Shen, and R. Rastogi (San Francisco, CA: ACM), 855–864.
- Jin, W., Yang, K. K., Barzilay, R., and Jaakkola, T. S. (2019). “Learning multimodal graph-to-graph translation for molecular optimization,” in *The 7th International Conference on Learning Representations: OpenReview.net* (New Orleans, LA).
- Karimi, M., Wu, D., Wang, Z., and Shen, Y. (2019). DeepAffinity: interpretable deep learning of compound-protein affinity through unified recurrent and convolutional neural networks. *Bioinformatics* 35, 3329–3338. doi: 10.1093/bioinformatics/btz111
- Kearnes, S., McCloskey, K., Berndl, M., Pande, V., and Riley, P. (2016). Molecular graph convolutions: moving beyond fingerprints. *J. Comput. Aided Mol. Des.* 30, 1–14. doi: 10.1007/s10822-016-9938-8
- Keshava Prasad, T. S., Goel, R., Kandasamy, K., Keerthikumar, S., Kumar, S., Mathivanan, S., et al. (2009). Human protein reference database 2009 update. *Nucleic Acids Res.* 37, D767. doi: 10.1093/nar/gkn892
- Kingma, D. P., and Ba, J. (2015). Adam: a method for stochastic optimization. *arXiv [Preprint]*. Available online at: <https://arxiv.org/abs/1412.6980v9> (accessed May 15, 2020).
- Kipf, T. N., and Welling, M. (2016). Variational graph auto-encoders. *arXiv [Preprint]*. Available online at: <https://arxiv.org/abs/1611.07308> (accessed September 9, 2019).
- Knox, C., Law, V., Jewison, T., Liu, P., Ly, S., Frolkis, A., et al. (2011). DrugBank 3.0: a comprehensive resource for ‘omics’ research on drugs. *Nucleic Acids Res.* 39, D1035–1041. doi: 10.1093/nar/gkq1126
- Kuhn, M., Campillos, M., Letunic, I., Jensen, L. J., and Bork, P. (2010). A side effect resource to capture phenotypic effects of drugs. *Mol. Syst. Biol.* 6:343. doi: 10.1038/msb.2009.98
- Lan, W., Lai, D., Chen, Q., Wu, X., Chen, B., Liu, J., et al. (2020). LDICDL: LncRNA-disease association identification based on Collaborative Deep Learning. *IEEE/ACM Trans. Comput. Biol. Bioinform.* doi: 10.1109/TCBB.2020.3034910. [Epub ahead of print].
- Lan, W., Wang, J., Li, M., Liu, J., Li, Y., Wu, F.-X., et al. (2016). Predicting drug-target interaction using positive-unlabeled learning. *Neurocomputing* 206, 50–57. doi: 10.1016/j.neucom.2016.03.080
- Le, N. Q. K. (2019). Fertility-GRU: identifying fertility-related proteins by incorporating deep-gated recurrent units and original position-specific scoring matrix profiles. *J. Proteome Res.* 18, 3503–3511. doi: 10.1021/acs.jproteome.9b00411
- Le, N. Q. K., Do, D. T., Chiu, F. Y., Yapp, E. K. Y., Yeh, H. Y., and Chen, C. Y. (2020). XGBoost improves classification of MGMT promoter methylation status in IDH1 wildtype glioblastoma. *J. Pers. Med.* 10:128. doi: 10.3390/jpm10030128
- Le, N. Q. K., Yapp, E. K. Y., Nagasundaram, N., and Yeh, H. Y. (2019). Classifying promoters by interpreting the hidden information of DNA sequences via deep learning and combination of continuous fasttext N-grams. *Front. Bioeng. Biotechnol.* 7:305. doi: 10.3389/fbioe.2019.00305

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found at: <https://github.com/shulijiuba/GADTI>.

AUTHOR CONTRIBUTIONS

ZL and QC conceived the project, developed the prediction approach. ZL and WL designed and implemented the experiments. ZL, HP, XH, and SP analyzed the result. ZL wrote the paper. All authors read and approved the final manuscript.

FUNDING

This work was supported by the National Natural Science Foundation of China (Nos. 61963004, 61702122 and 62072124), the Natural Science Foundation of Guangxi (Nos. 2017GXNSFDA198033 and 2018GXNSFBA281193), the Key Research and Development Plan of Guangxi (No. AB17195055), and the Science and Technology Base and talent Special project of Guangxi (No. AD20159044).

- Li, Q., Han, Z., and Wu, X. M. (2018). Deeper insights into graph convolutional networks for semi-supervised learning. *arXiv [Preprint]*. Available online at: <https://arxiv.org/abs/1801.07606v1> (accessed April 20, 2020).
- Liu, Z., Chen, Q., Lan, W., Liang, J., Chen, Y.-P. P., and Chen, B. (2020). A survey of network embedding for drug analysis and prediction. *Curr. Protein Peptide Sci.* 21:1. doi: 10.2174/1389203721666200702145701
- Luo, Y., Zhao, X., Zhou, J., Yang, J., Zhang, Y., Kuang, W., et al. (2017). A network integration approach for drug-target interaction prediction and computational drug repositioning from heterogeneous information. *Nat. Commun.* 8:573. doi: 10.1038/s41467-017-00680-8
- Ma, T., Xiao, C., Zhou, J., and Wang, F. (2018). "Drug similarity integration through attentive multi-view graph auto-encoders," in *The 27th International Joint Conference on Artificial Intelligence*, ed. J. Lang (California: International Joint Conferences on Artificial Intelligence), 3477–3483.
- Mei, J., Kwok, C. K., Yang, P., Li, X., and Zheng, J. (2013). Drug-target interaction prediction by learning from local information and neighbors. *Bioinformatics* 29, 238–245. doi: 10.1093/bioinformatics/bts670
- Mohamed, S. K., Nováček, V., and Nounu, A. (2019). Discovering protein drug targets using knowledge graph embeddings. *Bioinformatics* 36, 603–610. doi: 10.1093/bioinformatics/btz600
- Nagamine, N., Shirakawa, T., Minato, Y., Torii, K., Kobayashi, H., Imoto, M., et al. (2009). Integrating statistical predictions and experimental verifications for enhancing protein-chemical interaction predictions in virtual screening. *PLoS Comput. Biol.* 5:e1000397. doi: 10.1371/journal.pcbi.1000397
- Olayan, R. S., Ashoor, H., and Bajic, V. B. (2018). DDR: Efficient computational method to predict drug-target interactions using graph mining and machine learning approaches. *Bioinformatics* 34, 1164–1173. doi: 10.1093/bioinformatics/btx731
- Öztürk, H., Ozkirimli, E., and Özgür, A. (2019). WideDTA: prediction of drug-target binding affinity. *arXiv [Preprint]*. Available online at: <https://arxiv.org/abs/1902.04166v1> (accessed April 8, 2020).
- Pan, S., Hu, R., Long, G., Jiang, J., Yao, L., and Zhang, C. (2018). "Adversarially regularized graph autoencoder for graph embedding," in: *The 27th International Joint Conference on Artificial Intelligence*, ed. J. Lang (California: International Joint Conferences on Artificial Intelligence), 2609–2615.
- Perozzi, B., Alrfou, R., and Skiena, S. (2014). "DeepWalk: online learning of social representations," in: *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, eds. S.A. Macskassy, C. Perlich, J. Leskovec, W. Wang and R. Ghani (New York, NY: ACM), 701–710.
- Rogers, D., and Hahn, M. (2010). Extended-connectivity fingerprints. *J Chem Inf Model* 50, 742–754. doi: 10.1021/ci100050t
- Seal, A., Ahn, Y.-Y., and Wild, D. J. (2015). Optimizing drug-target interaction prediction based on random walk on heterogeneous networks. *J. Cheminform.* 7:40. doi: 10.1186/s13321-015-0089-z
- Shi, H., Liu, S., Chen, J., Li, X., Ma, Q., and Yu, B. (2019). Predicting drug-target interactions using Lasso with random forest based on evolutionary information and chemical structure. *Genomics* 111, 1839–1852. doi: 10.1016/j.ygeno.2018.12.007
- Smith, T. F., and Waterman, M. S. (1981). Identification of common molecular subsequences. *J. Mol. Biol.* 147, 195–197. doi: 10.1016/0022-2836(81)90087-5
- Su, C., Tong, J., Zhu, Y., Cui, P., and Wang, F. (2018). Network embedding in biomedical data science. *Brief. Bioinform.* 21, 182–197. doi: 10.1093/bib/bby117
- Tong, H., Faloutsos, C., and Pan, J. Y. (2008). Random walk with restart: fast solutions and applications. *Knowl. Inform. Syst.* 14, 327–346. doi: 10.1007/s10115-007-0094-2
- Wan, F., Hong, L., Xiao, A., Jiang, T., and Zeng, J. (2019). NeoDTI: neural integration of neighbor information from a heterogeneous network for discovering new drug-target interactions. *Bioinformatics* 35, 104–111. doi: 10.1093/bioinformatics/bty543
- Wang, W., Yang, S., Zhang, X., and Li, J. (2014). Drug repositioning by integrating target information through a heterogeneous network model. *Bioinformatics* 30, 2923–2930. doi: 10.1093/bioinformatics/btu403
- Wen, M., Zhang, Z., Niu, S., Sha, H., Yang, R., Yun, Y., et al. (2017). Deep-learning-based drug-target interaction prediction. *J. Proteome Res.* 16, 1401–1409. doi: 10.1021/acs.jproteome.6b00618
- Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K., and Jegelka, S. (2018). "Representation learning on graphs with jumping knowledge networks," in: *The 35th International Conference on Machine Learning*, eds. J.G. Dy and A. Krause (UK: PMLR), 5449–5458.
- Yang, B., Yih, W., He, X., Gao, J., and Deng, L. (2015). "Embedding Entities and Relations for Learning and Inference in Knowledge Bases," in: *The 3rd International Conference on Learning Representations*, eds. Y. Bengio and Y. LeCun (ICLR). Available online at: www.iclr.cc
- Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., and Leskovec, J. (2018). "Graph convolutional neural networks for web-scale recommender systems," in: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (London: Association for Computing Machinery), 974–983.
- Zheng, X., Ding, H., Mamitsuka, H., and Zhu, S. (2013). "Collaborative matrix factorization with multiple similarities for predicting drug-target interactions," in: *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, eds. I.S. Dhillon, Y. Koren, R. Ghani, T.E. Senator and P. Bradley (New York, NY: ACM), 1025–1033.
- Zhu, S., Bing, J., Lin, C., Zeng, X., and Min, X. (2018). Prediction of drug-gene interaction by Using Metapath2vec. *Front. Genet.* 9:248. doi: 10.3389/fgene.2018.00248
- Zitnik, M., Agrawal, M., and Leskovec, J. (2018). Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics* 34, 457–466. doi: 10.1093/bioinformatics/bty294
- Zong, N., Kim, H., Ngo, V., and Harismendy, O. (2017). Deep mining heterogeneous networks of biomedical linked data to predict novel drug-target associations. *Bioinformatics* 33, 2337–2344. doi: 10.1093/bioinformatics/btx160
- Zong, N., Wong, R. S. N., Ngo, V., Yu, Y., and Li, N. (2019). Scalable and accurate drug-target prediction based on heterogeneous bio-linked network mining. *bioRxiv [Preprint]*. doi: 10.1101/539643. Available online at: <https://www.biorxiv.org/content/10.1101/539643v1> (accessed April 10, 2020).

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2021 Liu, Chen, Lan, Pan, Hao and Pan. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.



CLGBO: An Algorithm for Constructing Highly Robust Coding Sets for DNA Storage

Yanfen Zheng, Jieqiong Wu and Bin Wang*

The Key Laboratory of Advanced Design and Intelligent Computing, Ministry of Education, School of Software Engineering, Dalian University, Dalian, China

OPEN ACCESS

Edited by:

Pan Zheng,
University of Canterbury, New Zealand

Reviewed by:

Xiangtao Li,
Jilin University, China
Wu Sheng,
Beijing Research Center
for Information Technology
in Agriculture, China

*Correspondence:

Bin Wang
wangbin@dlu.edu.cn

Specialty section:

This article was submitted to
Computational Genomics,
a section of the journal
Frontiers in Genetics

Received: 22 December 2020

Accepted: 08 April 2021

Published: 04 May 2021

Citation:

Zheng Y, Wu J and Wang B
(2021) CLGBO: An Algorithm
for Constructing Highly Robust
Coding Sets for DNA Storage.
Front. Genet. 12:644945.
doi: 10.3389/fgene.2021.644945

In the era of big data, new storage media are urgently needed because the storage capacity for global data cannot meet the exponential growth of information. Deoxyribonucleic acid (DNA) storage, where primer and address sequences play a crucial role, is one of the most promising storage media because of its high density, large capacity and durability. In this study, we describe an enhanced gradient-based optimizer that includes the Cauchy and Levy mutation strategy (CLGBO) to construct DNA coding sets, which are used as primer and address libraries. Our experimental results show that the lower bounds of DNA storage coding sets obtained using the CLGBO algorithm are increased by 4.3–13.5% compared with previous work. The non-adjacent subsequence constraint was introduced to reduce the error rate in the storage process. This helps to resolve the problem that arises when consecutive repetitive subsequences in the sequence cause errors in DNA storage. We made use of the CLGBO algorithm and the non-adjacent subsequence constraint to construct larger and more highly robust coding sets.

Keywords: DNA storage, primer and address sequences, CLGBO, non-adjacent subsequence constraint, DNA coding sets

INTRODUCTION

The amount of global data has exponentially increased with the advent of the Internet age, and is expected to grow from 45 ZB in 2019 to 175 ZB in 2025 (Reinsel et al., 2018). The problems of existing storage media, which include difficulty in achieving large capacity storage, the existence of high maintenance costs, limited service life and easy data loss, mean that the storage industry faces unprecedented challenges and opportunities (Zhirnov et al., 2016). It is therefore urgent to find a new storage medium to meet the demand of data storage. In 1953, Watson and Crick (1953) published a paper on the molecular structure of nucleic acid. Their research opened the door to the study of biogenetics, and also promoted people to begin exploring the form of life from a new perspective at the molecular biological level. Later, deoxyribonucleic acid (DNA) molecular replication, DNA molecular recombination, genetic code, genetic information transmission and other genetic molecular mechanisms make people have a more comprehensive and profound understanding of DNA gene theory. Information about organisms has been stored in DNA molecules composed of four bases called adenine (A), thymine (T), guanine (G), and cytosine

(C) for three billion years since life first came into existence on the earth. Pair pairs between A and T, C, and G can form stable double-stranded structures, and both single-stranded DNA and double-stranded DNA can be used to store information in the form of binary code. **Figure 1** shows the structural models of single-stranded and double-stranded DNA. DNA storage has the advantages of high storage density, low maintenance costs and long storage life compared with the traditional storage media, and it is a widely studied area for researchers (Ping et al., 2019).

In 2012, Church et al. (2012) successfully stored a 650 KB sized book in oligonucleotides (shorter DNA sequences) and retrieved them by sequencing. Shortly thereafter, Goldman et al. (2013) stored 739 KB of information in DNA and recovered the original file with 100% accuracy. In 2015, Grass et al. (2015) demonstrated that digital information could be stored in DNA and that the original information could be recovered error-free over long periods of time using error-correcting codes. Later, in the same year, Yazdi et al. (2015) proposed that DNA storage could provide ultra-high data storage capacity. They described a DNA storage architecture that allowed random access and rewriting of information blocks. In 2017, Erlich and Zielinski (2017) stored a complete computer-operating system, movies and other files in a DNA sequence with a total size of 2.14×10^6 bytes. This level of storage was several orders of magnitude higher than previously reported work that used a storage strategy called DNA Fountain. In 2018, Organick et al. (2018) stored 35 different files (over 200 MB of data) and demonstrated that each file could be recovered accurately using a random-access method. A year later, Lopez et al. (2019) demonstrated the successful decoding of 1.67 MB of information stored in DNA sequences using portable nanopore sequencing. In 2020, Meiser et al. (2020) proposed a protocol that focused on providing an ideal starting point for small experiments and reducing the corresponding error rate by changing the parameters of the encoder/decoder to achieve a higher amount of data storage and random access to the data. Chen Y. J. et al. (2020) studied the heterogeneity of oligonucleotide replication and showed that the two main sources of bias were the synthesis and amplification processes. They built statistical models for each molecule and the entire process based on these findings. Lin et al. (2020) proposed a simple architecture consisting of a T7 promoter and a single-strand protruding domain (SS-dsDNA) that can be used for dynamic DNA information storage. In another study (Chen H. et al., 2020), Chen et al. proposed a DNA hard drive as a rewritable molecular storage system. Data could only be read after the correct key was provided, which ensured the security of the data

storage. In 2021, Cao et al. (2021) proposed a thermodynamic minimum free energy constraint and applied to the construction of DNA storage coding sets. The introduction of this constraint improves the quality of DNA coding and reduces the error rate in the storage process.

The process of DNA storage involves the following steps: DNA coding (mapping binary data to nucleotide sequences), DNA synthesis (synthesizing specific sequences of DNA to complete the writing of the code), DNA processing and storage, polymerase chain reaction (PCR) amplification to enable random access to the data, followed by sequencing (reading) with a sequencing instrument, and DNA decoding (mapping nucleotide sequences to binary data).

Three important processes in the DNA storage are described here in detail. The results of DNA coding directly affect the performance of DNA storage. The entire DNA coding consists mainly of the process of data compression, introduction of error correction and conversion to DNA sequence:

- (1) **Compression:** Compression makes greater use of DNA storage space and removes redundancy before storing information in DNA. Common compression methods include Hoffman coding and Fountain coding but there are many examples. In 2013, Goldman et al. (2013) used Hoffman coding in DNA storage for the first time, which increased the coding efficiency to 1.58 bit/nt. This coding method can reduce but does not avoid the appearance of homopolymers; it does control the GC content well. In 2017, Erlich and Zielinski (2017) used DNA Fountain in DNA storage for the first time and used a quadratic conversion model with 00, 01, 10, and 11 mapped to A, C, G, and T, respectively. This encoding method filters sequences containing homopolymers and GC content anomalies and improves the encoding efficiency to 1.98 bit/nt.
- (2) **Introduction of error correction:** In each process of DNA storage, errors may occur that result in the loss of the original digital information. The introduction of an error-correction mechanism is necessary to obtain accurate information. The introduction of an error-correction mechanism at the coding stage is the most effective way to ensure accuracy and cost saving. Error correction improves accuracy by removing redundancy. It is, however, critical to strike a balance between redundancy and accuracy. At present, Reed-Solomon codes (RS codes) are the main error-correction method. In 2015, Grass et al. (2015) applied RS coding to DNA storage for error-free storage. RS coding has the advantage of

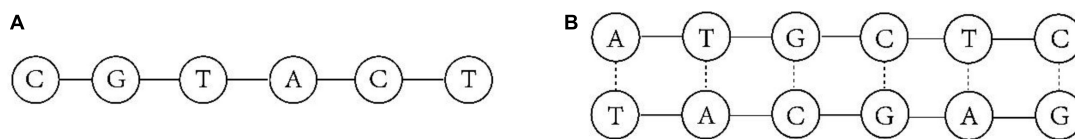


FIGURE 1 | DNA model. (A) single-stranded DNA. (B) double-stranded DNA.

recovering more information about the original data with less redundancy.

- (3) Conversion model: The conversion of digital information to DNA information is required for the conversion model to work. The coding model can be divided into three forms depending on the conversion method: binary, ternary, and quaternary. The binary model was used by Church et al. (2012) in 2012. The conversion of the model was achieved by encoding the binary digits into specific DNA sequences where A or G was coded as 0 and T or C was coded as 1. The binary model effectively avoids the effects of unbalanced GC content or homopolymers. In 2013, Goldman et al. (2013) used the ternary model to convert information into DNA sequence. The entire base sequence had three states: 0, 1, and 2. The ternary model mainly determines the last base by the first base. However, it does not establish a specific mapping relationship between bases and data like the binary model. The ternary model can store more information than the binary model. However, the ternary model does not take full advantage of the storage power of DNA. In 2017, Erlich and Zielinski (2017) used the quaternary coding model to map A, T, G, and C to 00, 01, 10, and 11, respectively. The quaternary coding model has the strongest storage capacity compared with other models, but it is prone to excessive GC content and high homopolymers, which impacts DNA storage.

Deoxyribonucleic acid synthesis chemically joins one nucleotide to another and forms a single-stranded DNA sequence (Kosuri and Church, 2014). In the synthesis process, the coupling efficiency is 99% at each step but the small error still results in an exponential decrease in product yield with increasing length. Therefore the length of the synthesized DNA sequence should be kept to about 200 nucleotides (Bornholt et al., 2016). DNA sequencing technology is used to determine the DNA sequence. The current DNA sequencing technology is divided into three main generations. The first generation of sequencing technologies mainly include the double deoxygenated strand end-termination sequencing method proposed by Sanger et al. (1977) and the chemical degradation method invented by Maxam and Gilbert (1977). The first generation DNA sequencing technologies can sequence up to 1,000 bp in length, but have the disadvantages of slow speed (the automatic Sanger sequencer can only read 1,000 bases in 24 h) and high cost (about \$1 to sequence 600–700 bp). The second generation of sequencing technologies arose due to advances in science and technology and the efforts of researchers to specifically improve sequencing technologies. It is also known as next generation sequencing (NGS) or high throughput sequencing, which allows rapid sequencing of millions of molecules simultaneously at one time. The second generation of sequencing technologies also has its limitations. Most NGS requires primers for *in vitro* template amplification and sequencing of the resulting template library. Replication errors and loss of information can occur during this process (e.g., the errors mentioned earlier are most likely to occur in sequences with high and low GC content and the presence of

homopolymers; Church et al., 2012). The second generation of sequencing technologies solved the problem of high throughput. Today researchers are more inclined to study the characteristics of single molecules of DNA and the third generation of DNA sequencing technology was created for this purpose. The third generation sequencing refers to single-molecule sequencing technology. It is capable of analyzing long sequences and produces only random errors although it has a relatively high error rate (about 10%; Yazdi et al., 2017). It is an inevitable that DNA storage technology will be widely used in the next few years due to the maturity and success of DNA synthesis and sequencing technologies (Carmean et al., 2018). However, non-specific hybridization, mutation, insertion, deletion and other errors are common during DNA storage and that can lead to data-reading errors and deletions.

Therefore, it is vital to study the sources of errors that impact DNA storage and coding. Earlier studies (Myers, 2007 Tandem Repeats and Morphological Variation | Learn Science at Scitable; Kovacevic and Tan, 2018; Schwarz et al., 2020) revealed that the error rate in the storage process increases if there are consecutive repetitive subsequences in the sequence. Hence, we propose a novel constraint (non-adjacent subsequence constraint) to avoid the occurrence of this sequence. The design of coding sets under multiple constraints is difficult and belongs to the NP problem. However, the heuristic algorithms that have emerged in recent years are well suited to this problem by virtue of their low complexity and high accuracy. Hence, an improved optimization algorithm is proposed, which uses two mutation strategies to enhance the gradient-based optimizer (GBO). Specifically, this algorithm takes advantage of the Cauchy mutation operator for random perturbation to increase the diversity of the population and improve the ability of the algorithm to explore the optimal value globally. At the same time, the Levy mutation operator is used to enhance the local search ability of the algorithm, and this helps to avoid falling into local optima. In this study, the combination of Cauchy and Levy mutation strategy (CLGBO) and specific constraints (Hamming distance, GC content, No-runlength constraint, and non-adjacent subsequence constraint) not only ensures the quality of the coding sets but also improves its lower bounds.

The article is structured as follows: section “Coding constraints in DNA storage” describes in detail the four constraints of the coding sets in DNA storage. Section “Algorithm description” describes the CLGBO algorithm and the test results and analysis of the improved algorithm. Section “Designing of lower bounds of coding sets” describes the design of coding sets and comparison of lower bounds of DNA coding sets. Section “Conclusion” summarizes this study and presents an overall outlook.

CODING CONSTRAINTS IN DNA STORAGE

The most important and difficult aspect of DNA storage is the synthesis and sequencing of DNA strands. The two processes are most prone to substitution, deletion and insertion errors.

According to statistics, the error probability of each nucleotide in the sequencing process is 1% (Press et al., 2020), and some special cases (For example, there are homopolymers, consecutive repetitive subsequences, and the content of G and C bases is too high or too low in the DNA sequence.) will produce higher error rate. During storage, DNA molecules are prone to non-specific hybridization reactions. If non-specific hybridization occurs between DNA molecules, it may prevent the DNA molecules carrying information from being sequenced normally, and will also cause data reading failure. By restricting the DNA sequence to comply with the following constraints, the incidence of non-specific hybridization and the rate of read and write errors can be reduced:

Non-adjacent Subsequence

Deoxyribonucleic acid sequences containing consecutive repetitive subsequences are more likely to be misaligned during sequencing and this results in data-reading errors (Myers, 2007 Tandem Repeats and Morphological Variation | Learn Science at Scitable). Sequences containing consecutive repetitive subsequences easily produce polymerase slippage at the synthesis phase (Schwarz et al., 2020). Two DNA sequences can easily become dislocated in the repetitive region. For example, an ATG subsequence on one sequence could base-pair with the first TAC in the other sequence, or the second, or the third. In this study, we mainly focus on the case where the length of subsequence is 2 and 3. For example, there is a subsequence AG in the GTAGAGAGCTA sequence, and there is a subsequence TGA in the AGTGATGACG sequence. Sequences containing these two types are not added to the DNA coding sets. For the coding set A, any sequence S ($S = s_1s_2...s_n$) exists as follows:

$$\begin{aligned} \text{when } K = 2 \\ s_i s_{i+1} \neq s_{i+2} s_{i+3}, 0 < i \leq n - (2k - 1) \\ \text{when } K = 3 \\ s_i s_{i+1} s_{i+2} \neq s_{i+3} s_{i+4} s_{i+5}, 0 < i \leq n - (2k - 1) \end{aligned} \quad (1)$$

Hamming Distance

For any two sequences v ($v = v_1v_2...v_n$) and u ($u = u_1u_2...u_n$) of length n in the DNA coding sets, the Hamming distance is expressed as the number of different elements at the same position between the two sequences v and u (Aboluion et al., 2012). $H(v, u)$ is required to be $H(v, u) \geq d$, where d is the defined threshold, and $H(v, u)$ calculates the Hamming distance by the following formula:

$$H(v, u) = \sum_{i=1}^n h(v_i, u_i), h(v_i, u_i) = \begin{cases} 0, & v_i = u_i \\ 1 & v_i \neq u_i \end{cases} \quad (2)$$

The Hamming distance can be used to measure the similarity between different sequences. The larger the value of d , the greater the differences between the sequences and the less similar they are. The smaller the value of d , the smaller the differences between the sequences, the greater the similarity and the more likely it is for non-specific hybridization to occur between sequences. This will result in storage errors. In addition, the Hamming distance has an error-correction function with relational

data elasticity, which can also effectively decrease the error rate in the process.

GC Content

GC content is the percentage of bases G and C in a DNA sequence (Wang et al., 2020). An appropriate GC content is crucial to maintain the chemical stability of DNA sequences because the base pair G-C contains three hydrogen bonds, while the base pair A-T contains two hydrogen bonds. Previous work has shown that 50% GC content is optimal (Chee and Ling, 2008; Aboluion et al., 2012; Tulpan et al., 2014) and the formula is as follows:

$$GC(s) = \frac{|G| + |C|}{|s|} \times 100\% \quad (3)$$

where $GC(s)$ denotes the GC content of sequences, $|G|$ and $|C|$ denote the number of bases G and C, respectively, in sequence s , and $|s|$ denotes the number of bases in the entire sequence.

No-Runlength

The presence of homopolymers in sequences is one of the major sources of errors during DNA storage. Overly long homopolymers can lead to insertion, substitution and deletion errors (Church et al., 2012). For example, in TAAAGC, the presence of A base repeats can easily be misinterpreted as TAAGC or TAAAAGC during sequencing. Therefore, it is required that each DNA sequence should not contain consecutive repetitive bases (Erlich and Zielinski, 2017). The presence of consecutively repetitive bases during sequencing will read them as a single signal and may result in data loss. It is therefore strictly forbidden to have the same bases adjacent to each sequence and this is mathematically modeled as follows:

$$S_i \neq S_{i+1}, i \in [1, n - 1] \quad (4)$$

ALGORITHM DESCRIPTION

Gradient-Based Optimizer

Generally speaking, optimization methods can be divided into two categories: one is gradient-based optimization methods, such as gradient descent method (Keshavan and Sewoong, 2009), newton method (Agarwal et al., 2006), and quasi-newton method (Broyden, 1970); the other is non-gradient-based optimization method, namely metaheuristic algorithm. Algorithms of this type can be divided into two categories: one is single objective algorithm such as animal migration optimization algorithm (Li et al., 2014), simulated annealing algorithm (Rutenbar, 1989), cuckoo search algorithm (Li and Yin, 2015), the gray wolf optimization algorithm (Mirjalili et al., 2014), differential evolution algorithm (Li and Yin, 2011a), henry gas optimization algorithm (Hashim et al., 2019), multi-search differential evolution algorithm (Li et al., 2017b), hybrid differential evolution with biogeography-based optimization (Li and Yin, 2011b), the other is multi-objective algorithm such as: the NSGA-II (Huang et al., 2010), multi-objective biogeography based optimization algorithm (Li and Yin, 2013), new multi-objective optimization algorithm combined

with opposition-based learning (Ewees et al., 2021), and multi-objective ranking binary artificial bee colony algorithm (Li et al., 2017a).

Ahmadianfar et al. (2020), inspired by the gradient-based Newtonian approach, developed the GBO, a powerful and efficient algorithm that combines gradient and metaheuristic algorithm. Gradient-based methods are widely used to solve optimization problems. The optimal solution using a gradient-based optimization algorithm is found by determining an extreme point at which the gradient is equal to zero. In the gradient-based optimization method, a search direction is selected and the search process moves along this direction toward the optimal solution (Shahidi et al., 2005). In the metaheuristic algorithm, the initial solution (i.e., the initial population) is randomly generated and the search direction is determined from the results of previous searches. The search direction will not stop updating until the convergence condition is satisfied. This kind of method is very effective in finding the global optimal. Therefore, it is worthwhile to develop a population-based optimization algorithm that uses the gradient method to skip infeasible points and move toward feasible regions. GBO is mainly composed of a gradient search rule (GSR) and a local escape operator (LEO). The GSR uses a gradient-based approach to enhance the exploration capability of the algorithm and speed up convergence to obtain a better position in the entire search space. The LEO are mainly used to improve the efficiency of GBO for solving complex problems and to escape from local optima. All detailed mathematical models of GBO can be found in the literature (Ahmadianfar et al., 2020).

Improved Algorithm

The GBO algorithm has low computational complexity and a simple structure. However, this algorithm also has some disadvantages. The main function of the LEO phase of the algorithm is to avoid the occurrence of local optimal stagnation, but only when the random number is less than 0.5, will it enter the LEO phase (Ahmadianfar et al., 2020). In addition to being easy to fall into the local optimum, the GBO algorithm has the shortcomings of premature convergence, imbalance between exploitation and exploration, and slow convergence speed. A method called mutation strategy is introduced in this work to solve these shortcomings. The basic GBO algorithm is embedded with two innovations, the Cauchy mutation strategy and the Levy mutation strategy, to improve the overall optimization performance of the algorithm. Mutation strategy is a commonly used method for evolutionary algorithms to produce new individuals, which can effectively enrich the population. However, it is difficult for a single mutation operator to effectively balance the exploration and exploitation capabilities of the algorithm. Therefore, an algorithm combining two mutation operators is proposed to alleviate the lack of population diversity, the imbalance between exploitation and exploration, and the premature and slow convergence of the GBO algorithm. The fitness of the mutated individual is compared with the fitness of the parent. The parent is replaced by the mutated individual to improve the overall quality if the

fitness of the mutated individual is better than the parent. The experimental results show that the CLGBO algorithm is significantly improved in terms of convergence speed, stability and seeking accuracy.

Cauchy Mutation Strategy

The Cauchy mutation operator is an effective strategy to improve the algorithm (Wang et al., 2007; Hu et al., 2009; Ali and Pant, 2011; Sapre and Mini, 2019). The theoretical basis of the Cauchy mutation operator is derived from the standard Cauchy distribution density function, which is defined by eq. (5). The Cauchy distribution density function has a smaller peak at the origin but a longer distribution at both ends (**Figure 2**). This allows individuals to have a higher probability of jumping to a better position, which means that the Cauchy mutation operator has strong global control. The Cauchy distribution function has a relatively small peak value and individuals spend less time searching adjacent intervals in the iterative process. More energy is put into searching for the global optimal value around the best individual, which means that the improved algorithm has a good ability to adjust and to optimize its searching capabilities. The use of the Cauchy mutation operator for random perturbation has several benefits. It helps to increase the diversity of the population and makes the exploration range of the previous iteration broader and more inclined to be a promising area. And the important point is that it can effectively reduce the search blind spots and improve the exploration ability of the algorithm. In addition, the characteristics of the Cauchy distribution enable it to generate random numbers that are far away from the origin. This means that individuals after the Cauchy mutation have the ability to quickly escape from the local optimal value. Eqs (5, 6) are given by

$$f_{\text{cauchy}}(r) = \frac{1}{\pi(1 + \gamma^2)}, \quad (5)$$

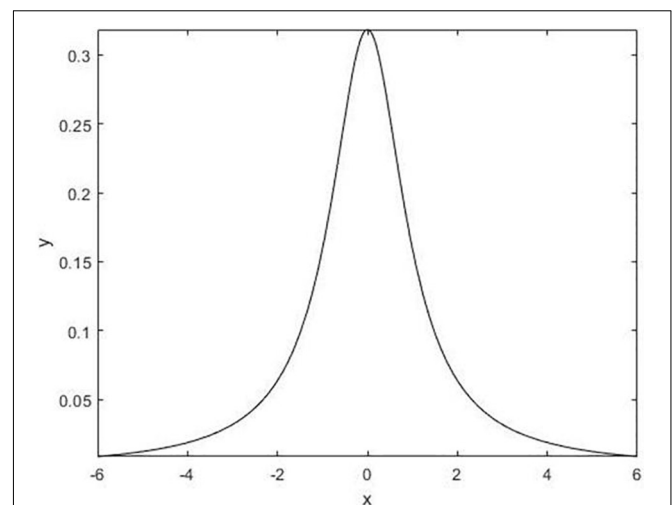


FIGURE 2 | Standard Cauchy distribution density function.

$$y = \frac{1}{2} + \frac{1}{\pi} \arctan(\gamma). \quad (6)$$

Equation (6) is the mathematical model of the standard Cauchy distribution function and y is a random number uniformly distributed on the interval of (0,1) $\gamma = \tan(\pi(y-1/2))$. The Cauchy mutation operator $C(\gamma)$ is obtained according to Eqs (5, 6) and is used to update the position. The formula is as follows:

$$X_{\text{new}}(i) = X(i) + X(i) \times C(\gamma) \quad i \in \{1, \dots, N\}. \quad (7)$$

Levy Mutation Strategy

Many organisms in nature use the Levy flight strategy when foraging for food (Faramarzi et al., 2020). Moreover, many heuristic algorithms have been improved based on this strategy and achieved good results (Zhu et al., 2013; Aydogdu et al., 2016; Li et al., 2019; Iacca et al., 2021). Levy flight has a strong disturbance capability and is a motion mode of alternate exploration through high-frequency short distance exploration and low-frequency long distance exploration. This not only expands the search range but also enhances the local search capability in a specific region. Moreover, this approach can avoid falling into the local optimal when seeking the optimal solution in a large range. Another important point is that the introduction of Levy flight can effectively avoid the excessive dependence of position changes on the position information of the previous generation, thus ensuring the diversity of the species. A simple version of the Levy distribution is mathematically defined as

$$\text{levy}(\beta) : \mu = t^{-1-\beta}, 0 < \beta \leq 2. \quad (8)$$

The expressions of Levy random numbers are as follows:

$$\text{levy}(\beta) : \frac{\varphi \times \mu}{|\nu|^{1/\beta}}, \quad (9)$$

$$\varphi = \left[\frac{\Gamma(1+\beta) \times \sin(\pi \times \beta/2)}{\Gamma((\frac{1+\beta}{2}) \times \beta \times 2^{\frac{\beta-1}{2}})} \right]^{1/\beta}, \quad (10)$$

where μ and ν are all standard normal distribution, β is typically 1.5, Γ is the standard gamma function.

The Levy mutation operator is applied to the GBO algorithm to update the position and the formula is as follows:

$$X_{\text{new}}(i) = X(i) + X(i) \times L(\beta) \quad i \in \{1, \dots, N\}, \quad (11)$$

where $L(\beta)$ is a randomly distributed number obtained from the Levy distribution. The Levy flight strategy can search in the space far enough away from the current optimal solution to ensure that individuals can jump out of the local optimal solution.

Pseudo-Code of CLGBO

The pseudo-code for CLGBO is as follows.

```

Assign values for parameters  $pr$ ,  $\epsilon$  and  $M$ 
Generate an initial population
Evaluate the objective function value
Specify the best and worst solutions  $x_{\text{best}}$  and  $x_{\text{worst}}$ 
While( $m < M$ )
    for  $i = 1 : N$ 
        for  $n = 1 : D$ 
            Select randomly  $r1 \neq r2 \neq r3 \neq r4 \neq n$  in the range of  $[1, N]$ 
            Calculate the position  $x_{i,n}^{m+1}$ 
        end for
        Local escaping operator
        if  $\text{rand} < pr$ 
            Calculate the position  $x_{LEO}^m$ 
             $x_i^{m+1} = x_{LEO}^m$ 
        end
        Create the new position  $x_{\text{new}}$  using Eqn.(7)
        if  $x_{\text{new}}$  better than  $x(i)$ 
             $x(i) = x_{\text{new}}$ 
        end if
        Create the new position  $x_{\text{new}}$  using Eqn.(11)
        if  $x_{\text{new}}$  better than  $x(i)$ 
             $x(i) = x_{\text{new}}$ 
        end if
        Update the positions  $x_{\text{best}}$  and  $x_{\text{worst}}$ 
    end for
     $m = m + 1$ 
end

```

Experimental Environment

All experimental tests were conducted in a unified environment and the detailed parameters are shown in Table 1.

Benchmark Functions and Experimental Setup

The 14 benchmark functions of the famous CEC-2017 are used to comprehensively evaluate the overall performance of the CLGBO algorithm. These 14 test functions have been widely used in previous studies (Hashim et al., 2019; Faramarzi et al., 2020). The 14 test functions are divided into two

TABLE 1 | Operating environment.

Name	Value
Hardware:	
CPU	Core i5
Frequency	2.30 GHz
RAM	8 GB
Software:	
Operating system	Windows 10
Language	MATLAB R2018b

categories as a benchmark to test the performance of the algorithm: one is a unimodal function (F_1 – F_6) and the other is a multimodal function (F_7 – F_{14}). The mathematical model, dimension, search space, and theoretical optimal values of all functions are listed in **Tables 1, 2** in note 1 of the **Supplementary Material**. The CLGBO algorithm was compared with six well-known metaheuristic optimization algorithms to benchmark its performance: GBO, GWO, CS, ABC, WOA, and ISA. All of the data for the performance of these algorithms were taken from the literature (Ahmadianfar et al., 2020). In addition, all tests were conducted under same conditions. The size of the population and the maximum number of iterations were set at 50 and 500, respectively. At the same time, each test function was independently executed 30 times to reduce the randomness of the results, the best, average and standard deviation values were calculated. When solving the minimum problem, the smaller average value is, the better the algorithm performance, and a smaller standard deviation value indicates a more stable algorithm. Therefore, we use average and standard deviation values to evaluate the performance and stability of the algorithm. The specific results are shown in **Tables 2, 3** and bold font indicates the best results. In the following subsections, the exploitation, exploratory capability and speed of convergence of the CLGBO algorithm are analyzed. A non-parametric statistical Wilcoxon rank sum test is also conducted to further evaluate the algorithm.

Experimental Results

Evaluation of the Exploitation Ability

Unimodal functions (F_1 – F_6) are usually used to evaluate the exploitation ability of the optimization algorithm. These test functions have only one global optimal solution and no local optimal solution. They can therefore be used to evaluate the exploitation capability of the CLGBO algorithm. The results

of the CGBO (LGBO) algorithm obtained by adding only the Cauchy (Levy) mutation operator are shown in **Table 2**. The CGBO and LGBO algorithms have improved in all three test metrics (best value, average, and standard deviation), but are inferior to the CLGBO algorithm, which proves that the combination of the two mutation operators is more effective than using only one of them. For example, the mean value of function F_1 is reduced by more than 100 orders of magnitude by only using one of the mutation strategies, but neither of them converges to the global minimum value 0. When the two mutation strategies are combined, the average value converges to the global optimal value 0. The best value (Best), average (AVG), and standard deviation (SD) of the test functions F_1 – F_3 and F_5 – F_6 for the CLGBO algorithm have reached the global optimal value. The F_4 function does not reach the global optimal value. However, its optimal value and average value are improved compared with the original GBO algorithm. The average value of 5 of the 6 unimodal test functions is 0, which proves that the algorithm converges to the global optimum in different mathematical models, and their variances are also 0, which proves that the data has strong stability. Compared with the other six optimization algorithms, the CLGBO algorithm has obvious advantages in the exploitation stage.

Evaluation of the Exploration Ability

The exploration ability of the CLGBO algorithm is evaluated by multimodal functions (F_7 – F_{14}). These functions have a global optimal solution and a large number of local optimal solutions. The number of local optimal solutions increases exponentially as the dimensions of the problems increase. Therefore, the multimodal functions can reflect well the exploration ability of the algorithm. The results of the CGBO (LGBO) algorithm obtained by simply adding the Cauchy (Levy) mutation operator

TABLE 2 | Results of the unimodal test functions.

ID	Metric	CLGBO	CGBO	LGBO	GBO	GWO	CS	ABC	WOA	ISA
F_1	Best	0.00E + 00	1.15E-309	0.00E + 00	1.26E-135	4.33E-29	4.44E-05	6.25E-10	9.43E-89	2.94E + 00
	AVG	0.00E + 00	6.85E-309	3.61E-316	1.46E-125	3.87E-27	2.52E-02	1.77E-02	6.75E-80	9.87E + 01
	SD	0.00E + 00	0.00E + 00	0.00E + 00	7.96E-125	7.73E-27	1.17E-01	6.49E-02	2.45E-79	1.92E + 02
F_2	Best	0.00E + 00	0.00E + 00	0.00E + 00	2.33E-206	2.79E-108	1.46E-06	1.90E-76	9.17E-141	6.76E-09
	AVG	0.00E + 00	0.00E + 00	0.00E + 00	3.29E-193	4.17E-97	1.81E + 01	3.76E-54	1.56E-110	1.61E-01
	SD	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	1.87E-96	8.44E + 01	2.06E-53	7.86E-110	6.00E-01
F_3	Best	0.00E + 00	5.59E-311	0.00E + 00	1.50E-138	2.25E-31	5.38E-03	2.11E-08	2.88E + 01	4.16E-04
	AVG	0.00E + 00	1.10E-297	9.88E-324	2.40E-128	5.78E-29	9.00E-01	1.32E + 00	5.52E + 03	1.54E-02
	SD	0.00E + 00	0.00E + 00	0.00E + 00	1.21E-127	1.48E-28	1.70E + 00	2.68E + 00	3.85E + 03	2.88E-02
F_4	Best	1.87E + 01	1.79E + 01	1.83E + 01	1.98E + 01	2.52E + 01	2.96E + 01	3.97E + 01	2.69E + 01	2.35E + 01
	AVG	2.14E + 01	2.10E + 01	2.16E + 01	2.16E + 01	2.68E + 01	1.39E + 02	6.93E + 01	2.75E + 01	7.56E + 01
	SD	1.41E + 00	1.32E + 00	1.86E + 00	8.03E-01	7.53E-01	2.37E + 02	5.50E + 01	4.12E-01	5.18E + 01
F_5	Best	0.00E + 00	2.25E-309	0.00E + 00	3.92E-140	1.61E-34	6.67E-06	4.06E-16	2.63E-94	2.54E-05
	AVG	0.00E + 00	3.79E-298	1.98E-323	8.86E-131	5.60E-33	5.16E-04	1.56E-08	2.86E-84	1.50E-01
	SD	0.00E + 00	0.00E + 00	0.00E + 00	4.07E-130	5.84E-33	7.63E-04	7.60E-08	1.11E-83	7.42E-01
F_6	Best	0.00E + 00	3.06E-308	0.00E + 00	1.35E-136	1.12E-31	1.22E-02	1.48E-10	2.90E-89	2.80E-02
	AVG	0.00E + 00	1.10E-293	1.34E-321	9.61E-129	5.14E-30	1.88E-01	6.39E + 00	1.30E-81	6.14E + 01
	SD	0.00E + 00	0.00E + 00	0.00E + 00	4.92E-128	8.14E-30	3.04E-01	3.50E + 01	5.59E-81	1.65E + 02

TABLE 3 | Results of the multimodal test functions.

ID	Metric	CLGBO	CGBO	LGBO	GBO	GWO	CS	ABC	WOA	ISA
F ₇	Best	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	2.11E + 00	7.74E + 00	8.69E + 00	0.00E + 00	9.06E + 00
	AVG	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	5.91E + 00	9.86E + 00	1.05E + 01	3.00E + 00	1.09E + 01
	SD	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	2.20E + 00	8.36E-01	9.07E-01	4.43E + 00	8.96E-01
F ₈	Best	5.70E-10	4.68E-09	1.25E-09	4.60E-09	6.36E-01	6.28E-01	4.49E-01	6.99E-02	4.72E + 00
	AVG	1.05E-07	1.39E-07	7.07E-08	2.96E-07	1.01E + 00	2.41E + 00	3.84E + 00	5.12E-01	3.42E + 01
	SD	3.68E-07	3.07E-07	9.31E-08	8.45E-07	1.59E-01	2.27E + 00	3.98E + 00	3.58E-01	2.25E + 01
F ₉	Best	3.82E-04	3.82E-04	3.82E-04	3.82E-04	3.82E-04	3.82E-04	3.82E-04	3.82E-04	3.83E-04
	AVG	3.82E-04	3.82E-04	3.82E-04	3.82E-04	3.82E-04	4.12E-04	9.84E + 01	3.82E-04	1.87E-03
	SD	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	8.72E-13	4.54E-05	1.66E + 02	5.55E-13	5.08E-03
F ₁₀	Best	8.88E-16	8.88E-16	8.88E-16	8.88E-16	3.64E-14	4.69E-04	2.22E + 00	8.88E-16	1.33E-03
	AVG	8.88E-16	8.88E-16	8.88E-16	8.88E-16	4.46E-14	3.73E-03	4.90E + 00	3.73E-15	9.27E-01
	SD	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	4.19E-15	3.44E-03	1.51E + 00	2.70E-15	8.13E-01
F ₁₁	Best	1.42E-14	7.11E-15	2.84E-14	1.35E-13	2.27E + 01	8.53E-14	3.79E + 00	7.11E-15	3.46E + 01
	AVG	6.92E-14	6.70E-14	8.08E-14	1.97E-13	2.91E + 01	6.23E-02	9.29E + 00	1.92E-14	3.89E + 01
	SD	2.50E-14	2.51E-14	2.95E-14	3.62E-14	3.34E + 00	9.52E-02	3.89E + 00	6.62E-14	1.71E + 00
F ₁₂	Best	2.82E-01	1.66E-01	2.97E-01	3.46E-01	4.41E-01	4.42E-01	2.64E-01	2.60E-01	3.31E-01
	AVG	4.93E-01	4.88E-01	4.95E-01	5.31E-01	6.39E-01	5.93E-01	5.19E-01	5.24E-01	4.63E-01
	SD	1.22E-01	1.23E-01	1.14E-01	1.72E-01	9.60E-02	8.40E-02	1.84E-01	1.88E-01	9.80E-02
F ₁₃	Best	4.91E-01	4.97E-01	5.00E-01	4.06E-01	3.43E-01	3.18E-01	2.25E-01	1.21E-01	2.54E-01
	AVG	4.96E-01	5.00E-01	4.90E-01	4.24E-01	4.65E-01	4.54E-01	5.78E-01	3.84E-01	6.42E-01
	SD	1.93E + 00	6.29E-04	1.89E + 00	1.17E-02	7.42E-02	1.47E-01	2.71E-01	9.68E-02	2.96E-01
F ₁₄	Best	2.03E-243	1.18E-158	9.98E-172	2.95E-73	1.64E-19	2.60E-05	4.20E-18	0.00E + 00	1.27E-04
	AVG	3.92E-236	1.21E-153	1.40E-164	6.45E-69	4.27E-04	2.64E-02	1.74E-03	0.00E + 00	6.15E-01
	SD	0.00E + 00	4.83E-153	0.00E + 00	1.99E-68	5.72E-04	2.69E-02	9.45E-03	0.00E + 00	1.22E + 00

are listed in **Table 3**. The three test indexes (the best value, average, and standard deviation) of the CGBO and LGBO algorithms are improved but they are not as good as the CLGBO algorithm. This proves once again that the combination of mutation operators is more effective than using only one of these operators alone. The function F₇ also reaches the global optimal value in CLGBO. The average of the function F₈ is closer to the global optimal value than the other six algorithms. Its standard deviation is smaller than the other algorithms, which indicates that the results of the CLGBO algorithm are more stable. The functions F₉, F₁₀, and F₁₃ in CLGBO are almost identical to the results in GBO. F₁₁ and F₁₂ are not the best results for these seven algorithms. However, they are significantly better than the previous GBO results. The results show that CLGBO has strong exploration ability.

Evaluation of Convergence Efficiency

The convergence curve is an important indicator for the performance of the algorithm, through which we can see the convergence speed and the ability of the algorithm to jump out of the local optimum. For further illustration, the convergence curves of the CLGBO and other 5 algorithms are plotted in **Figures 3, 4**. **Figures 3, 4** contain three-dimensional representations and convergence curve of unimodal functions (F₃, F₆) and multimodal functions (F₉, F₁₄). The remaining three-dimensional representation of unimodal and multimodal functions and convergence curves can be found in note 2 of the **Supplementary Material**. All optimization algorithms

hope to achieve global optimization quickly and accurately. Convergence curves are often used to evaluate the convergence efficiency of an algorithm. The changes of convergence curves of the GBO, EO, WOA, GWO, and PSO algorithms are also depicted in **Figures 3, 4**. The convergence speed of the CLGBO algorithm is faster than the speed of the other five algorithms, which is clear from the convergence curves in **Figures 3, 4**. This is true for both the unimodal and multimodal functions and indicates that the CLGBO algorithm can achieve an appropriate balance between exploration and exploitation. More importantly, the convergence curves can reach the global optimal value accurately in the optimization process of the CLGBO algorithm.

Wilcoxon Rank Sum Test

The Wilcoxon rank sum test (Kim and Kim, 1996) was used to evaluate the significant difference between the two positions of the CLGBO algorithm. The test randomly selects two sets of samples and the *P*-values obtained can be used as an indicator for evaluating the algorithm. Specifically, the corresponding algorithm is considered to have a statistically significant advantage when the *P*-values are greater than 0.05.

We ran each algorithm 30 times and calculated its average value to avoid the randomness of the results. The *P*-values obtained by the 14 test functions from this statistical test are shown in **Tables 4, 5**. The *P*-values of the CLGBO algorithm are greater than 0.05, which indicates that this algorithm provides very competitive results.

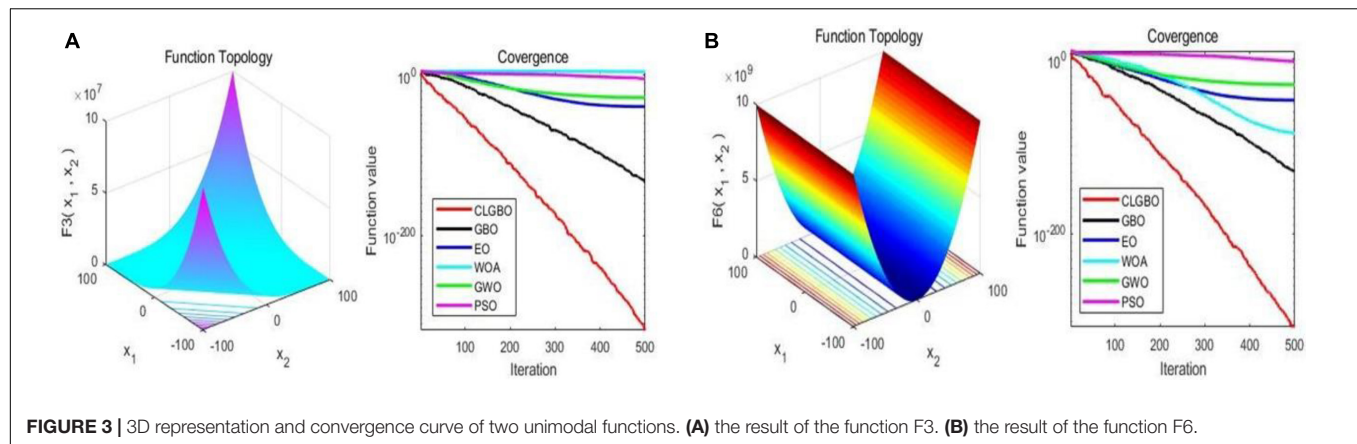


FIGURE 3 | 3D representation and convergence curve of two unimodal functions. **(A)** the result of the function F3. **(B)** the result of the function F6.

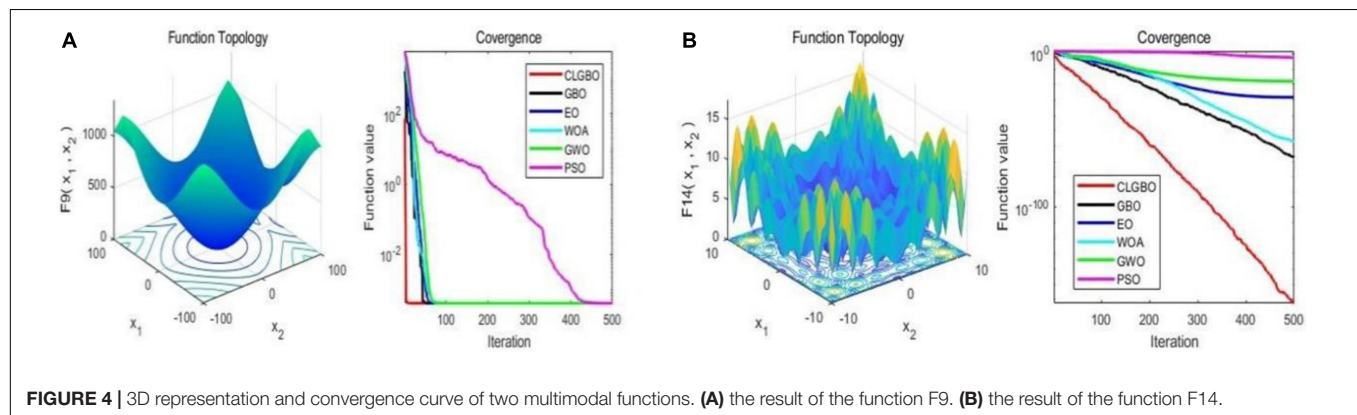


FIGURE 4 | 3D representation and convergence curve of two multimodal functions. **(A)** the result of the function F9. **(B)** the result of the function F14.

TABLE 4 | *P*-values of Wilcoxon rank sum test.

	F ₁	F ₂	F ₃	F ₄	F ₅	F ₆
GBO	0.47379	0.48579	0.52822	0.56719	0.51674	0.48343
CLGBO	0.51016	0.50425	0.57822	0.60641	0.59674	0.50238

DESIGNING OF LOWER BOUNDS OF CODING SETS

The construction of DNA storage coding sets that satisfy constraints can be used as primer (address) libraries. These constructed coding sets are essential to enable random storage. It has been shown in the literature (Organick et al., 2018) that each file can be recovered individually without error using a random-access method. Restricted by the existing DNA synthesis technology, the encoded base sequence will be divided into short sequences of the same length, and the length of a single sequence is generally no more than 200 bp. Each sequence

that needs to be synthesized includes primers, data, address bits, and error-correcting codes, etc., among which address bits are used for quick positioning, stitching and searching of each sequence. Primers are specially designed and added to both ends of the sequence prior to synthesis to extract the desired DNA sequence. We can obtain the content of this file by adding primers to the DNA pool using PCR technology for amplification, and subsequently sequencing and decoding. With the development of random DNA storage, primers, and address bits play important roles. Therefore, it is very essential to construct more and highly robust DNA coding sets as primer (address) libraries.

The Comparison of Lower Bounds

In this study, we apply the CLGBO algorithm to practical problems to improve the lower bounds of coding sets. $A^{GC,NL}$ (n , d , and w) represents the sets of DNA sequences that satisfy the GC content constraint, the No-runlength constraint and the Hamming distance constraint, where n represents the length of

TABLE 5 | *P*-values of Wilcoxon rank sum test.

	F ₇	F ₈	F ₉	F ₁₀	F ₁₁	F ₁₂	F ₁₃	F ₁₄
GBO	0.4922	0.44528	0.46159	0.6046	0.51303	0.53191	0.57046	0.57868
CLGBO	0.48824	0.47581	0.48725	0.48092	N/A	0.56072	0.70443	0.55438

the sequence, d represents the size of the Hamming distance and w represents the GC content, which is usually $n/2$. Meanwhile, we compared results for CLGBO algorithm with the best results recently obtained using the altruistic algorithm proposed by Limbachiya et al. (2018) and the NOL-HHO algorithm used by Yin et al. (2020). Altruistic algorithm is an intelligent algorithm which uses greedy algorithm to iteratively delete potential code words. It removes the “worst” candidate code word in each iteration. NOL-HHO algorithm is an algorithm to improve the Harris Hawks optimization algorithm by using a new nonlinear control parameter strategy and a random opposition-based learning strategy. **Tables 6, 7** show the lower bounds of coding sets of $4 \leq n \leq 10$, $3 \leq d \leq n$ obtained using the altruistic algorithm and NOL-HHO algorithm, respectively. **Table 8** shows the lower bounds of the coding sets using the CLGBO algorithm. The black bold font indicates the optimal result and the numbers in parentheses represent the best lower bounds of the coding sets acquired by the altruistic algorithm and the NOL-HHO algorithm. The superscripts are identified in **Table 9**.

The lower bounds of the coding sets acquired using the CLGBO algorithm are higher than the other two algorithms (**Table 8**). The multiple coding sets reported in the table are in the same state as previous work, for example, $n = 4, 5$, $d = 3$; $n = 7$, $d = 6$. This is mainly the case since we have reached the limit of the number of sequences that satisfy the constraint, which is the theoretically optimal value. However, the lower bound acquired using the CLGBO algorithm improves significantly further for the same value of d as n increases. For example, when $d = 3$, $n = 6, 7, 8, 9$, and 10 , the lower bounds of the coding sets obtained by CLGBO algorithm are 8.6–29.5% higher than the altruistic algorithm. When $d = 4$, $n = 6, 7, 8, 9$, and 10 , the lower bounds

obtained using the CLGBO algorithm are 4.3–7.4% higher than the NOL-HHO algorithm. In conclusion, the CLGBO algorithm can greatly increase the number of DNA coding sets and create conditions for storing large files. In addition, the increase of the lower bounds of the coding sets directly leads to improvements of the coding rate. The coding rate is defined as $R = \log_4^M/n$ (Cao et al., 2020), where n is the length of coded DNA and M is the number of the DNA coding set. For example, the values used in previous work (Limbachiya et al., 2018) are $n = 9$ and $d = 4$, $R = \log_4^{199}/9 \approx 0.42$. When $n = 8$, $d = 4$, the encoding rate also reaches 0.42 using our algorithm. Short sequences can therefore achieve the same storage performance as long sequences at the same coding rate.

Introduction of the Non-adjacent Subsequence Constraint

The sequence that contains consecutive repetitive subsequences is more prone to errors in the sequencing process, we propose an original constraint (non-adjacent subsequence constraint) for this, so that the constructed DNA coding sets can be more robust. The higher the robustness of the DNA coding sets, the lower the probability of errors in the DNA storage process. Therefore the non-adjacent subsequence constraint is added to the three basic constraints to build more stable and robust coding sets. The results are shown in **Table 10**. $A^{GC,NL,NS}(n, d, \text{ and } w)$ denotes DNA coding sets that satisfy the GC content, No-runlength, Hamming distance and non-adjacent subsequence constraints. In addition, in note 3 of the **Supplementary Material**, 66 sequences constructed using CLGBO when $n = 9$ and $d = 5$ are presented as experimental samples for detection.

The validity of the non-adjacent subsequence constraint was tested by calculating the variance of the melting temperature of the DNA coding sets. In a DNA set, the melting temperature (T_m) of the DNA sequence is the point when 50% of the DNA double-stranded molecules become single-stranded structures due to the process of heating and deformation (Sager and Stefanovic, 2005). The T_m will affect the rate of reactions between DNA molecules in PCR amplification. Non-specific hybridization is related to the structure of oligonucleotides and their thermodynamic properties. Significantly, each oligonucleotide in the library must have a similar T_m to reduce the possibility of non-specific hybridization of the oligonucleotide library (Chee and Ling, 2008). Therefore, each sequence must have a similar T_m when designing DNA coding sets. The variance is used to value the quality of sequences: the smaller the variance, the more stable the T_m of the whole coding set.

In this study, the concentration of the DNA molecule was set at 10 nM and the concentration of salt was set at 1 M. Coding sets with and without the new constraint obtained by the CLGBO algorithm were analyzed for their T_m values. As can be seen from the values in **Table 11**, 93% of the values show that the variance with the constraint is smaller than that without the constraint. In addition, the T_m variance of coding sets obtained by adding the new constraint were reduced by 10–66% compared with the values obtained without adding this constraint (**Table 11**). The T_m values of the sequences in

TABLE 6 | Lower bounds of the altruistic algorithm for $A^{GC,NL}(n, d, \text{ and } w)$; Limbachiya et al., 2018).

$n \backslash d$	3	4	5	6	7	8	9
4	11						
5	17	7					
6	44	16	6				
7	110	36	11	4			
8	289	86	29	9	4		
9	662	199	59	15	8	4	
10	1810	525	141	43	7	5	4

TABLE 7 | Lower bounds of the NOL-HHO algorithm for $A^{GC,NL}(n, d, \text{ and } w)$; Yin et al., 2020).

$n \backslash d$	3	4	5	6	7	8	9
4	12						
5	20	8					
6	55	23	8				
7	121	42	14	7			
8	339	108	35	13	5		
9	705	216	69	22	11	4	
10	1796	546	148	51	20	9	4

TABLE 8 | Lower bounds of the CLGBO algorithm for $A^{GC,NL}$ (n , d , and w).

$n \backslash d$	3	4	5	6	7	8	9
4	12 ^c (12 ⁿ)						
5	20 ^c (20 ⁿ)	8 ^c (8 ⁿ)					
6	58 ^c (55 ⁿ)	24 ^c (23 ⁿ)	8 ^c (8 ⁿ)				
7	131 ^c (121 ⁿ)	45 ^c (42 ⁿ)	17 ^c (14 ⁿ)	7 ^c (7 ⁿ)			
8	349 ^c (339 ⁿ)	113 ^c (108 ⁿ)	38 ^c (35 ⁿ)	15 ^c (13 ⁿ)	5 ^c (5 ⁿ)		
9	743 ^c (705 ⁿ)	234 ^c (216 ⁿ)	71 ^c (69 ⁿ)	27 ^c (22 ⁿ)	11 ^c (11 ⁿ)	5 ^c (4 ^{a,n})	
10	2030 ^c (1810 ^a)	580 ^c (546 ⁿ)	168 ^c (148 ⁿ)	56 ^c (51 ⁿ)	23 ^c (20 ⁿ)	9 ^c (9 ⁿ)	5 ^c (4 ^{a,n})

TABLE 9 | Meaning of superscript.

Superscript	Meaning
a	Altruistic algorithm (Limbachiya et al., 2018)
n	NOL-HHO algorithm (Yin et al., 2020)
c	CLGBO algorithm

TABLE 10 | Lower bounds for $A^{GC,NL,NS}$ (n , d , and w).

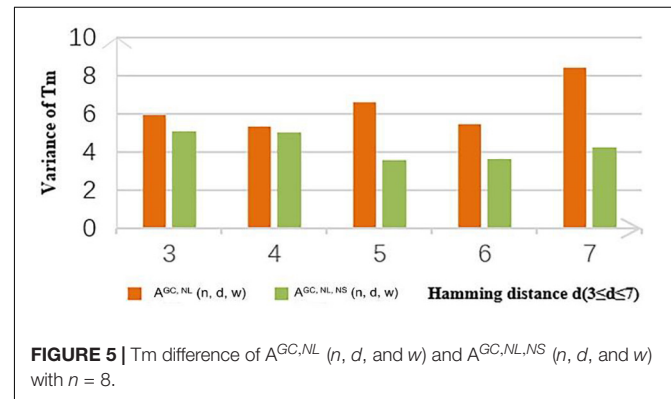
$n \backslash d$	3	4	5	6	7	8	9	10
6	51	22	8					
7	113	42	15	6				
8	319	105	35	15	5			
9	635	206	66	25	10	5		
10	1634	518	157	56	21	10	5	
11	2974	922	282	104	38	16	8	4
12	6184	1736	577	182	68	30	14	7
13	13590	3923	1050	386	130	50	24	10

TABLE 11 | Comparison of the variance of Tm.

$n \backslash d$		3	4	5	6	7
8	$A^{GC,NL}$	5.9351	5.2979	6.6136	5.4233	8.3799
	$A^{GC,NL,NS}$	5.0461	4.9945	3.5621	3.5888	4.2264
9	$A^{GC,NL}$	4.7033	4.8000	4.7655	3.6546	4.8876
	$A^{GC,NL,NS}$	4.5800	4.4041	3.9916	2.8110	1.4578
10	$A^{GC,NL}$	4.4705	4.5131	4.8233	5.0288	3.3062
	$A^{GC,NL,NS}$	4.0754	4.0554	4.2452	4.2037	3.9066

The bold values indicates the smaller the variance of Tm.

a coding set are closer if the Tm variance was smaller. To highlight our results, a comparison between the variances of Tm obtained for $A^{GC,NL}$ (n , d , and w) and $A^{GC,NL,NS}$ (n , d , and w) when $n = 8$ are shown in **Figure 5**. The variance of Tm for the coding sets with this constraint is smaller than without this constraint. And when $n = 8$, $d = 7$, the variance of the coding set with or without this constraint differs by 4.1535. When the variance of the coding set TM value is small, the possibility of non-specific hybridization is reduced and the PCR reaction is more stable. At the same time, the results confirm the applicability and necessity of the non-adjacent subsequence constraint.

**FIGURE 5** | Tm difference of $A^{GC,NL}$ (n , d , and w) and $A^{GC,NL,NS}$ (n , d , and w) with $n = 8$.

CONCLUSION

In this study, the CLGBO algorithm and non-adjacent subsequence constraint were combined to construct more stable primer and address libraries for DNA storage. First, the GBO algorithm was improved by employing the Cauchy mutation operator and Levy strategy. Cauchy mutation operator not only expands the diversity of the population, but also can effectively reduce the search blind spots, improve the exploration ability and convergence speed of the algorithm. Levy flight strategy can effectively avoid the over-dependence of position update on the previous position, and search for the optimal solution in a large range, so as to avoid falling into local optimum and premature convergence. The combination of the two strategies not only controlled the global ability well but also enhanced the local exploration ability, and makes the algorithm achieve a good balance in the exploitation and exploration stages. Next, the classical CEC-2017 test function and the Wilcoxon rank sum test were adopted to evaluate comprehensively the CLGBO algorithm in the exploitation phase, exploration phase and statistically. The test results and convergence curves showed that the CLGBO algorithm has stronger competitiveness, convergence ability and optimization ability compared with other algorithms. Second, CLGBO algorithm was applied to construct DNA storage coding sets. The lower bounds of DNA coding sets constructed by the CLGBO algorithm under the same constraint were significantly increased by 4.3–13.5% compared with previous work, and there was also

an improvement of the coding rate. When storing large files, it is possible to use shorter DNA primers and address sequences due to the improved lower bounds of the coding sets. Shorter DNA sequences mean lower error rates for DNA synthesis and sequencing. Finally, sequences containing consecutive repetitive subsequences are prone to cause errors during DNA storage. We therefore introduced the non-adjacent subsequence constraint to avoid mistakes and improve the stability of the coding sets. A comparison of the variance of T_m with and without this constraint showed that the variance of T_m with this constraint was reduced by 10–66%. The smaller T_m variance indicated that the T_m values of sequences in a DNA coding set were relatively similar. This can reduce the incidence of non-specific hybridization in the storage process and ensure that the DNA sequence is untied at similar temperatures during the PCR process to successfully amplify the DNA sequence.

In future work, we will further improve the lower bounds of the primer and address libraries while ensuring high robustness of the DNA coding sets. However, the quality of coding sets is inversely proportional to the quantity. It therefore remains a challenge to find the right balance between quality and quantity in future work. We will also continue to explore DNA storage and hope to come up with an original way of encoding for the payload and non-payload that will reduce redundancy and ensure accurate information recovery. In addition, the constructed coding sets can also be applied to other fields, including DNA image encryption (Zhou et al., 2020), DNA-binding proteins (Zhao et al., 2012), DNA computing (Li et al., 2020), and information security (Zhang et al., 2020).

REFERENCES

- Abolun, N., Smith, D. H., and Perkins, S. (2012). Linear and nonlinear constructions of DNA codes with Hamming distance d , constant GC-content and a reverse-complement constraint. *Discrete. Math.* 312, 1062–1075. doi: 10.1016/j.disc.2011.11.021
- Agarwal, A., Hazan, E., Kale, S., and Schapire, R. E. (2006). “Algorithms for portfolio management based on the Newton method,” in *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, 9–16. doi: 10.1145/1143844.1143846
- Ahmadianfar, I., Bozorg-Haddad, O., and Chu, X. (2020). Gradient-based optimizer: a new metaheuristic optimization algorithm. *Inform. Sci.* 540, 131–159. doi: 10.1016/j.ins.2020.06.037
- Ali, M., and Pant, M. (2011). Improving the performance of differential evolution algorithm using Cauchy mutation. *Soft. Comput.* 15, 991–1007. doi: 10.1007/s00500-010-0655-2
- Aydogdu, I., Akin, A., and Saka, M. P. (2016). Design optimization of real world steel space frames using artificial bee colony algorithm with Levy flight distribution. *Adv. Eng. Softw.* 92, 1–14. doi: 10.1016/j.advengsoft.2015.10.013
- Bornholt, J., Lopez, R., Carmean, D. M., Ceze, L., Seelig, G., and Strauss, K. (2016). “A DNA-Based archival storage system,” in *Proceedings of the Twenty-First International Conference on Architectural Support for Programming Languages and Operating Systems - ASPLOS '16*, Atlanta, GA: ACM Press, 637–649. doi: 10.1145/2872362.2872397
- Broyden, C. G. (1970). Quasi-newton methods. *Math. Comput.* 21, 368–381. doi: 10.1090/S0025-5718-1970-0279993-0
- Cao, B., Li, X., Zhang, X., Wang, B., Zhang, Q., and Wei, X. (2020). “Designing uncorrelated address constrain for DNA storage by DMVO algorithm,”

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

YZ: conceptualization, resources, and writing—original draft preparation. JW: investigation. BW: writing—review, editing, and funding acquisition. All authors have read and agreed to the published version of the manuscript.

FUNDING

This work is supported by the National Key Technology R&D Program of China (No. 2018YFC0910500), the National Natural Science Foundation of China (Nos. 61425002, 61751203, 61772100, 61972266, 61802040, and 61672121), the High-level Talent Innovation Support Program of Dalian City (Nos. 2017RQ060 and 2018RQ75), and the Innovation and Entrepreneurship Team of Dalian University (No. XQN202008).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/fgene.2021.644945/full#supplementary-material>

- in *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, Piscataway, NJ: IEEE, 1. doi: 10.1109/TCBB.2020.3011582
- Cao, B., Zhang, X., Wu, J., Wang, B., Zhang, Q., and Wei, X. (2021). Minimum free energy coding for DNA storage. *IEEE Trans. Nanobiosci.* 20, 212–222. doi: 10.1109/TNB.2021.3056351
- Carmean, D., Ceze, L., Seelig, G., Stewart, K., Strauss, K., and Willsey, M. (2018). DNA data storage and hybrid molecular–electronic computing. *Proc. IEEE* 107, 63–72. doi: 10.1109/JPROC.2018.2875386
- Chee, Y. M., and Ling, S. (2008). Improved lower bounds for constant GC-Content DNA codes. *IEEE. Trans. Inform. Theory.* 54, 391–394. doi: 10.1109/TIT.2007.911167
- Chen, H., Heidari, A. A., Chen, H., Wang, M., Pan, Z., and Gandomi, A. H. (2020). Multi-population differential evolution-assisted Harris hawks optimization: framework and case studies. *Future. Gener. Comp. Syst.* 111, 175–198. doi: 10.1016/j.future.2020.04.008
- Chen, Y. J., Takahashi, C. N., Organick, L., Bee, C., Ang, S. D., Weiss, P., et al. (2020). Quantifying molecular bias in DNA data storage. *Nat. Commun.* 11:3264. doi: 10.1038/s41467-020-16958-3
- Church, G. M., Gao, Y., and Kosuri, S. (2012). Next-generation digital information storage in DNA. *Science* 337, 1628–1628. doi: 10.1126/science.1226355
- Erlich, Y., and Zielinski, D. (2017). DNA Fountain enables a robust and efficient storage architecture. *Science* 355, 950–954. doi: 10.1126/science.aaj2038
- Ewees, A. A., Abd Elaziz, M., and Oliva, D. (2021). A new multi-objective optimization algorithm combined with opposition-based learning. *Expert. Syst. Appl.* 165:113844. doi: 10.1016/j.eswa.2020.113844
- Faramarzi, A., Heidarinejad, M., Mirjalili, S., and Gandomi, A. H. (2020). Marine Predators algorithm: a nature-inspired metaheuristic. *Expert. Syst. Appl.* 152:113377. doi: 10.1016/j.eswa.2020.113377

- Goldman, N., Bertone, P., Chen, S., Dessimoz, C., LeProust, E. M., Sipos, B., et al. (2013). Towards practical, high-capacity, low-maintenance information storage in synthesized DNA. *Nature* 494, 77–80. doi: 10.1038/nature11875
- Grass, R. N., Heckel, R., Puddu, M., Paunescu, D., and Stark, W. J. (2015). Robust chemical preservation of digital information on DNA in silica with error-correcting codes. *Angew. Chem. Int. Ed.* 54, 2552–2555. doi: 10.1002/anie.201411378
- Hashim, F. A., Houssein, E. H., Mabrouk, M. S., Al-Atabany, W., and Mirjalili, S. (2019). Henry gas solubility optimization: a novel physics-based algorithm. *Future. Gener. Comp. Syst.* 101, 646–667. doi: 10.1016/j.future.2019.07.015
- Hu, C., Wu, X., Wang, Y., and Xie, F. (2009). “Multi-swarm particle swarm optimizer with cauchy mutation for dynamic optimization problems,” in *Advances in Computation and Intelligence Lecture Notes in Computer Science*, eds Z. Cai, Z. Li, Z. Kang, and Y. Liu (Berlin: Springer), 443–453. doi: 10.1007/978-3-642-04843-2_47
- Huang, B., Buckley, B., and Kechadi, T.-M. (2010). Multi-objective feature selection by using NSGA-II for customer churn prediction in telecommunications. *Expert. Syst. Appl.* 37, 3638–3646. doi: 10.1016/j.eswa.2009.10.027
- Iacca, G., dos Santos Junior, V. C., and de Melo, V. V. (2021). An improved Jaya optimization algorithm with Levy flight. *Expert Syst. Appl.* 165:113902. doi: 10.1016/j.eswa.2020.113902
- Keshavan, R. H., and Sewoong, O. (2009). A gradient descent algorithm on the grassman manifold for matrix completion. *arXiv [Preprint]*. doi: 10.1016/j.trc.2012.12.007
- Kim, D., and Kim, Y. (1996). Wilcoxon signed rank test using ranked-set sample. *J. Comput. Appl. Math.* 3, 235–243. doi: 10.1007/BF03008904
- Kosuri, S., and Church, G. M. (2014). Large-scale de novo DNA synthesis: technologies and applications. *Nat. Methods* 11, 499–507. doi: 10.1038/nmeth.2918
- Kovacevic, M., and Tan, V. Y. F. (2018). Asymptotically optimal codes correcting fixed-length duplication errors in DNA storage systems. *IEEE. Commun. Lett.* 22, 2194–2197. doi: 10.1109/LCOMM.2018.2868666
- Li, X., Li, M., and Yin, M. (2017a). “Multiobjective ranking binary artificial bee colony for gene selection problems using microarray datasets,” in *IEEE/CAA Journal of Automatica Sinica*, Piscataway, NJ: IEEE, 1–16. doi: 10.1109/JAS.2016.7510034
- Li, X., Ma, S., and Hu, J. (2017b). Multi-search differential evolution algorithm. *Appl. Intell.* 47, 231–256. doi: 10.1007/s10489-016-0885-9
- Li, X., Wang, B., Lv, H., Yin, Q., Zhang, Q., and Wei, X. (2020). Constraining DNA sequences with a triplet-bases unpaired. *IEEE. Trans. Nanobiosci.* 19, 299–307. doi: 10.1109/TNB.2020.2971644
- Li, X., and Yin, M. (2011a). Design of a reconfigurable antenna array with discrete phase shifters using differential evolution algorithm. *Prog. Electromagn. Res.* 31, 29–43. doi: 10.2528/PIERB11032902
- Li, X., and Yin, M. (2011b). Hybrid differential evolution with biogeography-based optimization for design of a reconfigurable antenna array with discrete phase shifters. *Int. J. Antenn. Propag.* 2011:685629. doi: 10.1155/2011/685629
- Li, X., and Yin, M. (2013). Multiobjective binary biogeography based optimization for feature selection using gene expression data. *IEEE. Trans. NanoBiosci.* 12, 343–353. doi: 10.1109/TNB.2013.2294716
- Li, X., and Yin, M. (2015). Modified cuckoo search algorithm with self adaptive parameter method. *Inform. Sci.* 298, 80–97. doi: 10.1016/j.ins.2014.11.042
- Li, X., Zhang, J., and Yin, M. (2014). Animal migration optimization: an optimization algorithm inspired by animal migration behavior. *Neural. Comput. Appl.* 24, 1867–1877. doi: 10.1007/s00521-013-1433-8
- Li, Y., Li, X., Liu, J., and Ruan, X. (2019). An improved bat algorithm based on lévy flights and adjustment factors. *Symmetry* 11:925. doi: 10.3390/sym11070925
- Limbachiya, D., Gupta, M. K., and Aggarwal, V. (2018). Family of constrained codes for archival DNA data storage. *IEEE. Commun. Lett.* 22, 1972–1975. doi: 10.1109/LCOMM.2018.2861867
- Lin, K. N., Volk, K., Tuck, J. M., and Keung, A. J. (2020). Dynamic and scalable DNA-based information storage. *Nat. Commun.* 11:2981. doi: 10.1038/s41467-020-16797-2
- Lopez, R., Chen, Y. J., Dumas Ang, S., Yekhanin, S., Makarychev, K., Rac, M. Z., et al. (2019). DNA assembly for nanopore data storage readout. *Nat. Commun.* 10:2933. doi: 10.1038/s41467-019-10978-4
- Maxam, A. M., and Gilbert, W. (1977). A new method for sequencing DNA. *Proc. Natl. Acad. Sci. U.S.A.* 74, 560–564. doi: 10.1073/pnas.74.2.560
- Meiser, L. C., Antkowiak, P. L., Koch, J., Chen, W. D., Kohll, A. X., Stark, W. J., et al. (2020). Reading and writing digital data in DNA. *Nat. Protoc.* 15, 86–101. doi: 10.1038/s41596-019-0244-5
- Mirjalili, S., Mirjalili, S. M., and Lewis, A. (2014). Grey wolf optimizer. *Adv. Eng. Softw.* 69, 49–61. doi: 10.1016/j.advengsoft.2013.12.007
- Myers, P. (2007). Tandem repeats and morphological variation. *Nat. Educ.* 1:1.
- Organick, L., Ang, S. D., Chen, Y. J., Lopez, R., Yekhanin, S., Makarychev, K., et al. (2018). Random access in large-scale DNA data storage. *Nat. Biotechnol.* 36, 242–248. doi: 10.1038/nbt.4079
- Ping, Z., Ma, D., Huang, X., Chen, S., Liu, L., Guo, F., et al. (2019). Carbon-based archiving: current progress and future prospects of DNA-based data storage. *Gigascience* 8:giz075. doi: 10.1093/gigascience/giz075
- Press, W. H., Hawkins, J. A., Jones, S. K., Schaub, J. M., and Finkelstein, I. J. (2020). HEDGES error-correcting code for DNA storage corrects indels and allows sequence constraints. *Proc. Natl. Acad. Sci. U.S.A.* 117, 18489–18496. doi: 10.1073/pnas.2004821117
- Reinsel, D., Gantz, J., and Rydning, J. (2018). *The Digitization of the World From Edge to Core*. Available at: <https://www.seagate.com/cn/zh/our-story/data-age-2025/> (accessed September 20, 2020).
- Rutenbar, R. A. (1989). Simulated annealing algorithms: an overview. *IEEE. Circuits. Devices. Mag.* 5, 19–26. doi: 10.1109/101.17235
- Sager, J., and Stefanovic, D. (2005). “Designing nucleotide sequences for computation: a survey of constraints,” in *International Workshop on DNA-Based Computers*, eds A. Carbone and N. A. Pierce (Berlin: Springer), 275–289.
- Sanger, F., Nicklen, S., and Coulson, A. R. (1977). DNA sequencing with chain-terminating inhibitors. *P. Natl. Acad. Sci. U.S.A.* 74, 5463–5467. doi: 10.1073/pnas.74.12.5463
- Sapre, S., and Mini, S. (2019). Opposition-based moth flame optimization with Cauchy mutation and evolutionary boundary constraint handling for global optimization. *Soft. Comput.* 23, 6023–6041. doi: 10.1007/s00500-018-3586-y
- Schwarz, M., Welzel, M., Kabdullayeva, T., Becker, A., Freisleben, B., and Heider, D. (2020). MESA: automated assessment of synthetic DNA fragments and simulation of DNA synthesis, storage, sequencing and PCR errors. *Bioinformatics* 36, 3322–3326. doi: 10.1093/bioinformatics/btaa140
- Shahidi, N., Esmaeilzadeh, H., Abdollahi, M., Ebrahimi, E., and Lucas, C. (2005). “Self-adaptive memetic algorithm: an adaptive conjugate gradient approach,” in *IEEE Conference on Cybernetics and Intelligent Systems*, Piscataway, NJ: IEEE, doi: 10.1109/ICCIS.2004.1460378
- Tulpan, D., Smith, D. H., and Montemanni, R. (2014). Thermodynamic Post-processing versus GC-Content pre-processing for DNA codes satisfying the hamming distance and reverse-complement constraints. *IEEE. ACM. Trans. Comput. Biol.* 11, 441–452. doi: 10.1109/TCBB.2014.2299815
- Wang, B., Zhang, Q., and Wei, X. (2020). tabu variable neighborhood search for designing DNA barcodes. *IEEE. Trans. NanoBiosci.* 19, 127–131. doi: 10.1109/TNB.2019.2942036
- Wang, H., Li, H., Liu, Y., Li, C., and Zeng, S. Y. (2007). “Opposition-based Particle Swarm Algorithm with Cauchy mutation,” in *Proceedings of the IEEE Congress on Evolutionary Computation*, Singapore, doi: 10.1109/CEC.2007.4425095
- Watson, J. D., and Crick, F. H. C. (1953). Molecular structure of nucleic acids: a structure for deoxyribose nucleic acid. *Nature* 171, 737–738. doi: 10.1038/171737a0
- Yazdi, S. H. T., Kiah, H. M., Garcia-Ruiz, E., Ma, J., Zhao, H., and Milenkovic, O. (2015). DNA-based storage: trends and methods. *IEEE Trans. Mol. Biol. Mul. Scale Commun.* 1, 230–248. doi: 10.1109/TMBMC.2016.2537305
- Yazdi, S. M. H. T., Gabrys, R., and Milenkovic, O. (2017). Portable and error-free DNA-based data storage. *Sci. Rep. U. K.* 7:5011. doi: 10.1038/s41598-017-05188-1
- Yin, Q., Cao, B., Li, X., Wang, B., Zhang, Q., and Wei, X. (2020). An intelligent optimization algorithm for constructing a DNA storage code: NOL-HHO. *IJMS* 21:2191. doi: 10.3390/ijms21062191
- Zhang, X., Zhang, Q., Liu, Y., Wang, B., and Zhou, S. (2020). A molecular device: a DNA molecular lock driven by the nicking enzymes. *Comput. Struct. Biotec.* 18, 2107–2116. doi: 10.1016/j.csbj.2020.08.004

- Zhao, X. W., Li, X. T., Ma, Z. Q., and Yin, M. H. (2012). Identify DNA-binding proteins with optimal chou's amino acid composition. *Protein. Peptide. Lett.* 19, 398–405. doi: 10.2174/092986612799789404
- Zhirnov, V., Zadegan, R. M., Sandhu, G. S., Church, G. M., and Hughes, W. L. (2016). Nucleic acid memory. *Nat. Mater.* 15, 366–370. doi: 10.1038/nmat4594
- Zhou, S., He, P., and Kasabov, N. (2020). A dynamic DNA color image encryption method based on SHA-512. *Entropy Switz* 22:1091. doi: 10.3390/e22101091
- Zhu, X., Hao, X., and Xia, S. (2013). Feature selection algorithm based on Levy flight. *J. Zhejiang. Univ.* 47, 638–643. doi: 10.3785/j.issn.1008-973X.2013.04.011

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2021 Zheng, Wu and Wang. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.



Stable DNA Sequence Over Close-Ending and Pairing Sequences Constraint

Xue Li^{1†}, Ziqi Wei^{2†}, Bin Wang^{1*} and Tao Song^{3*}

¹ The Key Laboratory of Advanced Design and Intelligent Computing, Ministry of Education, School of Software Engineering, Dalian University, Dalian, China, ² School of Software, Tsinghua University, Beijing, China, ³ College of Computer and Communication Engineering, China University of Petroleum, Qingdao, China

OPEN ACCESS

Edited by:

Pan Zheng,
University of Canterbury, New Zealand

Reviewed by:

Effirul Ikhwan Ramlan,
Ulster University, United Kingdom
Leyi Wei,
Shandong University, China

*Correspondence:

Bin Wang
wangbinpaper@gmail.com
Tao Song
tsong@upc.edu.cn

[†] These authors have contributed
equally to this work

Specialty section:

This article was submitted to
Computational Genomics,
a section of the journal
Frontiers in Genetics

Received: 21 December 2020

Accepted: 12 April 2021

Published: 17 May 2021

Citation:

Li X, Wei Z, Wang B and Song T
(2021) Stable DNA Sequence Over
Close-Ending and Pairing Sequences
Constraint. *Front. Genet.* 12:644484.
doi: 10.3389/fgene.2021.644484

DNA computing is a new method based on molecular biotechnology to solve complex problems. The design of DNA sequences is a multi-objective optimization problem in DNA computing, whose objective is to obtain optimized sequences that satisfy multiple constraints to improve the quality of the sequences. However, the previous optimized DNA sequences reacted with each other, which reduced the number of DNA sequences that could be used for molecular hybridization in the solution and thus reduced the accuracy of DNA computing. In addition, a DNA sequence and its complement follow the principle of complementary pairing, and the sequence of base GC at both ends is more stable. To optimize the above problems, the constraints of Pairing Sequences Constraint (PSC) and Close-ending along with the Improved Chaos Whale (ICW) optimization algorithm were proposed to construct a DNA sequence set that satisfies the combination of constraints. The ICW optimization algorithm is added to a new predator-prey strategy and sine and cosine functions under the action of chaos. Compared with other algorithms, among the 23 benchmark functions, the new algorithm obtained the minimum value for one-third of the functions and two-thirds of the current minimum value. The DNA sequences satisfying the constraint combination obtained the minimum of fitness values and had stable and usable structures.

Keywords: DNA computing, DNA sequence design, constraint, WOA, ICW

INTRODUCTION

DNA computing is a new and promising interdisciplinary subject based on computational science and molecular biology, which shows great potential in solving NP problems (Wang et al., 2019; Zhu et al., 2020). At the end of the 20th century, Adleman (1994) used DNA molecules for calculation and solved the Hamiltonian problem (Heidari, 2014). The successful solution of this problem led DNA computing to become a field of great development. It has since been widely used to solve problems in many domains, including PCR amplification (Sze and Schloss, 2019), DNA sequencing (Shendure et al., 2017), bioinformatics (Zou and Liu, 2019), prediction of disease genes (Zeng et al., 2020a,b), image encryption (Zhou et al., 2020), and DNA data storage (Zhang et al., 2019; Cao et al., 2021), among others.

Benenson et al. (2004) made decisions through simple Boolean logic and successfully used RNA interference to construct molecular computing cores in human kidney cells. Yaakov et al. (Rinaudo et al., 2007) announced a breakthrough DNA computer, which can theoretically release anticancer drugs into cancer cells. In 2017, researchers used the CRISPR-Cas system (Shipman et al., 2017) to encode the pixel values of black-and-white images and short films into the genome of living bacterial populations; they minimized the technical limitations of the information storage system. Thubagere et al. (2017) developed a DNA robot that can control DNA to perform specific actions, such as picking and sorting goods in solution. Han et al. (2017) developed a new strategy called single-stranded origami (ssOrigami), which uses a single-stranded DNA or RNA as long as thin as a noodle to implement a self-folding structure without a topological junction, which could allow drugs to travel directly to the site of injury within the cell. Li et al. (2018) developed a nanorobot based on DNA origami technology that can be used to carry thrombin to accurately target tumor cells, and more broadly, this technology can be used for many types of cancer. Cherry and Qian (2018) used the extended seesaw motif DNA neural network for pattern discrimination and constructed a neural network using DNA sequence to realize the recognition of handwritten digits in model organisms. Palluk et al. (2018) proposed a strategy for the synthesis of oligonucleotides using a template-independent polymerase terminal deoxyribonucleotide transferase and obtained a scheme to repeatedly write a definite sequence. Schickinger et al. (2018) proposed DNA origami for creating a tethered multifluorophore movement experiment and explain interactions between cells. This is easy to use and has a wide range of applications. In order to avoid errors in DNA storage, Deng et al. (2019) adapted a hybrid coding system, which is composed of improved variable length run-length limited (vll-rl) codes and optimized photograph low-density parity check codes (LDPCs). Wang et al. (2020) proposed a deep learning framework, SeqEnhDL, to classify cell type-specific enhancers based on sequence features. This framework can transform folding changes of any DNA sequence into deep learning model features. Zhang et al. (2020) constructed a DNA molecular lock by using the characteristics of enzyme mutual inhibition and realized the information protection function at the molecular level.

In addition, in the face of massive data, the current computer is limited in terms of data storage and computing speed. Biomolecular computers have attracted the interest of scientists. DNA computer, as one of the biomolecular computers, has received much attention due to its small size, large storage capacity, fast operation, low energy consumption, and high parallelism. DNA computers use DNA (deoxyribonucleic acid) as a basis to bind enzymes for biochemical reactions that eventually generate DNA sequences carrying specific genetic information. These sequences are used to perform computation and solve the problem. DNA computing is encoded by A, T, C, and G, which is different from the binary combination of the traditional computer.

The design of DNA sequences is the key to perform DNA computing, and the quantity and quality of sequences can directly

affect the accuracy and efficiency of calculations. Therefore, a good coding method is of great significance to improve the reliability and accuracy of DNA computing. Deb et al. (2002) proposed a fast, non-dominated sorting genetic algorithm, which was used to solve a class of multi-objective optimization problems. With the help of linear coding, Gaborit and King (2005) developed a DNA code that met the anti-complement constraint and GC content requirement. Thus, they constructed an appropriate DNA sequence set. Shin et al. (2005) carried out multi-objective optimization for DNA sequences, including continuity, similarity, hairpin structure, H-measure, and GC content. Because the traditional algorithm cannot address the heterogeneity and conflict of DNA sequences, scientists designed a multi-objective optimization algorithm based on the artificial bee colony (MO-ABC) (Chaves-González et al., 2013), in which six kinds of conflict problems were solved, and finally, a reliable DNA sequence was generated. In 2014, in order to obtain effective DNA sequences, they proposed to use the multi-objective differential evolution algorithm (DEPT) (Chaves-González and Vega-Rodríguez, 2014) to optimize DNA sequences. In 2015, they used a hybrid multi-objective heuristic algorithm (H-MO-TLBO) (Chaves-González, 2015) to design DNA sequences. Yang et al. (2017) proposed to add the niche exclusion mechanism to improve the invasive weed optimization algorithm, which enhanced robustness and obtained the optimal sequence. Wang et al. (2018) improved the fast non-dominated sorting genetic algorithm II (INSGA-II) and achieved a high convergence rate and reliable DNA sequences. In 2019, in order to further optimize the DNA sequence, Chaves-González and Martínez-Gil (2019) introduced an algorithm called pMO-ABC that harvested different numbers of DNA sequences. In 2020, to decrease the error rate, Cao et al. (2020) presented a new constraint, namely, uncorrelated address, and constructed a set of effective DNA codes. Yin et al. (2020) considered multiple constraints to ensure accurate hybridization of DNA sequences.

In this study, to obtain a high-quality DNA sequence set, the constraints of Close-ending and Pairing Sequences Constraint (PSC) were added to the original constraint combination to form a new combinatorial constraint. The PSC addresses non-specific hybridization that occurs within the DNA sequences set, and the constraint adds a sliding method to ensure that each base can be traversed. Adding PSC to the DNA sequence set reduces the probability of interaction between sequences. The reason for the Close-ending constraint is that the G base and the C base in the sequence have three hydrogen bonds, and there are only two hydrogen bonds between the A base and the T base, so the stability of the AT end of the sequence is less than that of the G-C end. The DNA sequence reacts according to the principle of base complementary pairing. When G-C base is selected as the terminal of the sequence, the desired structure can be achieved when the DNA sequence continues to react. In addition, the constraint also include continuity, hairpin structure, H-measure, similarity, GC content, melting temperature, triplet-bases unpaired. The first four constraints are used as objective functions to calculate the fitness value; the remaining constraints are used to narrow the solution space. At the same time, we enrich and improve the WOA algorithm. In

addition, the predatory behavior of another marine mammal, manta ray (Zhao et al., 2020), is added to expand the predation range and maintain the diversity of the population. To improve the global search ability, the sine cosine model (Mirjalili, 2016) is combined with chaos (Shan et al., 2005) to further expand the solution space. After 23 benchmark functions, it is proved that the Improved Chaos Whale (ICW) optimization algorithm is meaningful. It reached the optimal value in most test functions. Under the combined action of the new constraint combination and the improved algorithm, excellent DNA sequences can be selected as elites. These elite sequences have a minimum value of zero in continuity and hairpin structure and the current minimum value in H-measure, followed by the minimum melting temperature change. In the evaluation of NUPACK, the concentrations of all DNA sequences before and after entering the solution were normalized to total values. All sequences showed stable and usable structures, indicating that the DNA sequences have good stability.

The rest of this article is arranged as follows. The second part introduces the constraints of constructing the DNA sequence set, including the new Close-ending constraint and PSC. The third part introduces the ICW optimization algorithm. In the fourth part, the results of fitness analysis and NUPACK evaluation are given. The last part is the summary and conclusion.

THE CONSTRAINTS ON DNA SEQUENCE DESIGN

To ensure the accuracy of DNA calculation and avoid non-specific hybridization of sequences, constraints must be imposed on DNA sequences. The construction of useful and high-quality DNA sequence set is dependent on strict constraints, which can enhance the robustness of the sequences. Continuity and hairpin structure constraints can effectively prevent sequences from generating secondary structures. The addition of triplet-bases unpaired and PSC to the sequences without secondary structure can not only avoid the self-complementary reaction but also effectively avoid the reaction between the sequences to generate other structures. On this basis, by adding similarity and H-measure, well-structured sequences be obtained that reduce unnecessary hybridization with other sequences. The addition of GC content and melting temperature constraints can keep the sequences in a thermodynamically stable state. Combined with Close-ending constraint, the formed DNA double strand is also stable in structure. Applying these constraints can lead to good sequences. In this study, we adopted all the above constraints in the design sequences.

Continuity

Continuity (Chaves-González et al., 2013) refers to the fact that the same bases are displayed side by side in a confined area. The continuous presence of the same base in a limited region can cause the DNA sequence to stack or distort. To avoid such a secondary structure of the DNA sequence, it is necessary to select a DNA sequence with little continuity. The continuity can be visualized with the following example.

Assuming that the DNA sequence threshold is 4, then in the sequence TTAGGGATCCATTTTT, the last sub-sequence with an underscore will trigger the threshold. To improve the quality of DNA sequences, such sequences will be removed from the sequence set. The mathematical formula is as follows:

$$f_{con}(L) = \sum_{p=1}^m Con(L_p) \quad (1)$$

$$Con(x) = \sum_{i=1}^{n-CT} T(cont_a(x, i), CT) \quad (2)$$

$$cont_a(x, i) = \begin{cases} c & \text{if } \exists c, \text{ s.t. } x_i \neq a, x_{i+j} = a \text{ for } 1 \leq j \leq c \\ & \text{and } x_{i+c+1} \neq a \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where m is the number of DNA sequence sets; L_p is a sequence in the DNA set L ; n is the number of bases in the current DNA sequence; CT is a specific continuity threshold; $T(b, CT)$ is a threshold function; when $b > CT$, the result is b ; otherwise, the result is 0. $cont_a(x, i)$ returns the number of consecutive bases, where $a \in \{A, T, C, G\}$.

Hairpin Structure

Hairpin structure (Chaves-González et al., 2013) is a secondary structure caused by the stacking of the DNA sequence itself, which may lead to inaccurate calculation. The hairpin structure is composed of a hair ring and hair stem. The number of bases for the hairpin to form the smallest ring is R_{min} , and P_{min} is the minimum length of the hairpin stem. The mathematical formula for calculating the hairpin value is as follows:

$$f_{hairpin}(L) = \sum_{p=1}^m Hairpin(L_p) \quad (4)$$

$$Hairpin(x) = \sum_{q=P_{min}}^{(n-R_{min})} \sum_{r=R_{min}}^{n-2q} \sum_{i=1}^{n-2q-r} T \left(\sum_{j=1}^{PL_{qri}} cb(x_{i+j}, x_{n-j}), \frac{PL_{qri}}{2} \right) \quad (5)$$

$$PL_{qri} = \min(p + i, l - r - i - p) \quad (6)$$

where r is the ring length of the hairpin structure, and q is the stem length. m is the number of DNA sequence sets, and n is the number of bases in a DNA sequence. For $T(a, y)$, when $a > y$, the result is a ; otherwise, it is 0. The function $cb(a, b)$ means that when a and b are complementary, the result is 1; otherwise, the result is 0. The equations should be inserted in editable format from the equation editor.

H-Measure

H-measure (Chaves-González, 2015) is a parameter to measure the degree of sequence hybridization. The parameter records

the number of complementary bases of two sequences. The calculation formula is as follows:

$$f_{H\text{-measure}}(L) = \sum_{i=1}^m \sum_{j=1, i \neq j}^m H\text{-measure}(L_i, L_j) \quad (7)$$

where m represents the size of sequence L sets; and L_i and L_j represent two sequences in opposite directions. The H-measure is classified into two types: continuous and discontinuous.

$$H_{\text{measure}(x,y)} = \text{Max}_{g,i} (h_{\text{dis}}(x, \text{shift})(y(-)^g y, t)) + h_{\text{cont}}(x, \text{shift})(y(-)^g y, t) \quad (8)$$

where x and y represent different DNA sequences. The shift function defines the offset from y to t .

$$h_{\text{dis}}(x, y) = T \left(\sum_{i=1}^n bp(x_i, x_j), H_{\text{dis}} \times \text{length}_{nb}(y) \right) \quad (9)$$

$$h_{\text{cont}}(x, y) = \sum_{i=1}^n T(cb p(x, y, i), H_{\text{cont}}) \quad (10)$$

where H_{dis} is a number between 0 and 1; H_{con} is a positive integer from 1 to n ; and the function $cb p(x, y, i)$ represents the length of a continuous base pair starting from the i th base of the sequence.

$$bp(x, y) = \begin{cases} 1 & x = y \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

$$cb p(x, y, i) = \begin{cases} c & \text{if } \exists c, \text{ s.t. } bp(x_i, y_j) = 0 \text{ and } bp(x_{i+c+1}, y_{i+c+1}) = 0 \\ \text{for } 1 \leq j \leq c \text{ and } bp(x_{i+c+1}, y_{i+c+1}) = 0 \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

Similarity

Similarity (Chaves-González, 2015) is an important index to evaluate sequence diversity. The higher the similarity, the more likely it is that non-specific hybridization will occur. It can calculate the number of the same base after the shift of two identical sequences. The higher the number of the same base, the more similar is the coding. Similarity is divided into discontinuous similarity and the largest continuous common subset. The formula for calculating similarity is as follows:

$$f_{\text{sim}}(L) = \sum_{i=1}^n \sum_{j=1}^n \text{Max}_{g,t} (s_{\text{dis}}(x, \text{Shift}(y(-)^g y, t)) + s_{\text{cont}}(x, \text{Shift}(y(-)^g y, t))) \quad (13)$$

where L is the set of DNA sequences; n is the number of set L ; and x and y are the different sequences in set L . $(-)$ indicates a gap.

Shift represents the offset of the encoding y through t . $g \in [0, 3]$.

$$s_{\text{dis}}(x, y) = T \left(\sum_{i=1}^n eq(x_i, y_i), DS \times n \right) \quad (14)$$

$$s_{\text{cont}}(x, y) = \sum_{i=1}^n T(ceq(x, y, i), CS) \quad (15)$$

$$eq(x, y) = \begin{cases} 1 & x = y \\ 0 & \text{otherwise} \end{cases} \quad (16)$$

$$ceq(x, y, i) = \begin{cases} c & \text{if } \exists c, \text{ s.t. } eq(x_i, y_j) = 0 \text{ and } eq(x_{i+c+1}, y_{i+c+1}) = 1 \\ \text{for } 1 \leq j \leq c \text{ and } eq(x_{i+c+1}, y_{i+c+1}) = 1 \\ 0 & \text{otherwise} \end{cases} \quad (17)$$

For $T(a, \text{value})$, when $a > \text{value}$, the result is a ; otherwise, it is 0. $ceq(x, y, i)$ is the length of the continuous common subsequence starting from the i th base of the sequence. DS is a real number from 0 to 1; CS is a positive integer from 1 to n .

Melting Temperature

The melting temperature (Yang et al., 2017) of DNA is an important parameter. In the process of DNA denaturation, double stranded DNA molecules undergo physical changes. In the process of denaturation from double strand to single strand, the temperature at which half of the DNA molecules are released is called the melting temperature. This behavior is an important constraint to ensure the thermodynamic stability of DNA molecules. The melting temperature is usually calculated by the gas chromatography content method and the nearest neighbor method. In this article, the melting temperature is calculated by the nearest neighbor method. The calculation formula is as follows:

$$T_m = \Delta H^\circ / (\Delta S^\circ + R \ln C_t) \quad (18)$$

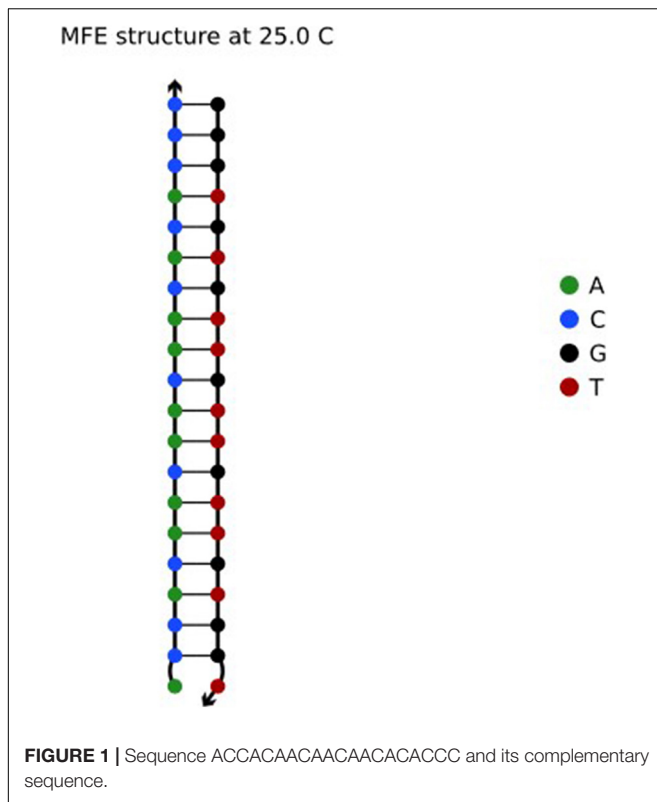
where ΔH and ΔS represent the standard enthalpy change and entropy change in the hybridization reaction, respectively, and the calculation method is the same as that of the free energy change. C_t is the molar concentration of DNA molecules. When the molecule is a symmetric sequence, its molar concentration is $C_t/4$. R is the gas constant of 1.987 cal/Kmol.

GC Content

GC content (Yang et al., 2017) is the ratio of guanine and cytosine in a DNA sequence. GC content is an important constraint; it can directly affect the stability of a DNA sequence. In a DNA sequence, the number of bases is expressed by n ; the number of guanine is a ; and the number of cytosine is b . Generally speaking, the GC content (t) of a sequence is

$$t = \frac{a+b}{n} * 100\% \quad (19)$$

Then, in the sequence GTTCGTACTGATCGTAGC, the GC content is $(5+4)/18 * 100\%$; that is, 50%.



Triplet-Bases Unpaired

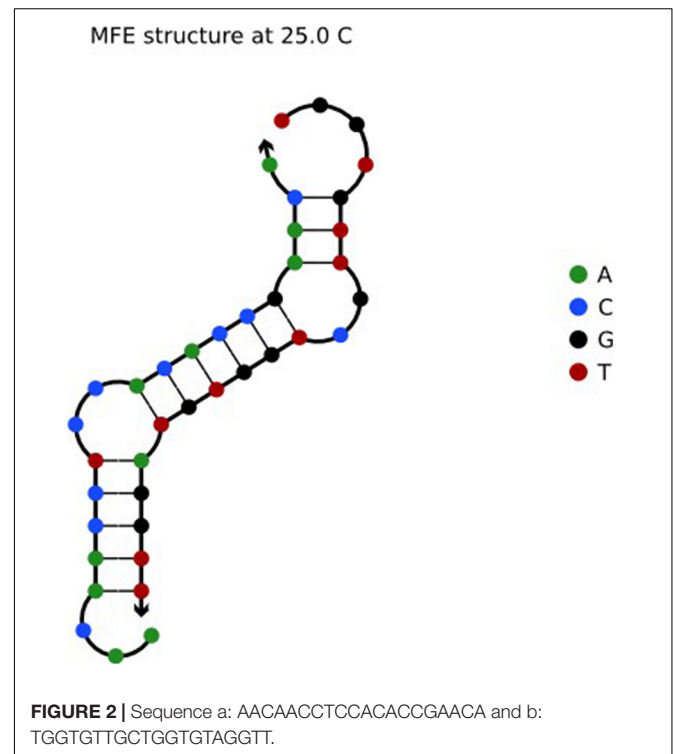
Triplet-bases unpaired (Li et al., 2020) is employed to avoid the complementary reaction of DNA sequence in solution. The sequence entered into NUPACK is evaluated, and the result is denoted by i . $i = C_{output}/C_{input}$; C_{output} is the sum of the sequence concentrations in the solution after NUPACK input; and C_{input} is the sum of the DNA sequence concentrations when NUPACK is input. The closer i is to 1, the higher is the sequence quality. X is a DNA sequence; n is the base number of X ; Y is the inverse sequence of X ; and x, y are the subsequences of X and Y . It is expressed by the following formula:

$$f_{pair}(x) = \begin{cases} pair(x) & subcb(x, y, k) = 3 \\ x & subcb(x, y, k) \neq 3 \end{cases} \quad (20)$$

where $x = (x_i, x_{i+1}, x_{i+2})$; $y = (y_j, y_{j+1}, y_{j+2})$; and $i, j \in [1, n-2]$. The function $subcb(x, y, k)$ calculates the number of base complementary pairs starting from the k th base.

Close-Ending

In a pair of complementary DNA sequences, the sequence usually creates a gap at one end of the A-T base pair, resulting in an unstable structure (Chaves-González and Vega-Rodríguez, 2014) of the sequence in solution. There are three hydrogen bonds between the G and C bases in the DNA sequence, but there are only two hydrogen bonds between A and T, so the bond strength formed by A-T is less than that between G-C. The structure evaluated in NUPACK is shown in **Figure 1**.



As shown in **Figure 1**, one end of the sequence is G-C. Here, the reaction takes place according to the principle of complementary base pairs. However, there is a gap at one end of the A base and T base, which is less stable than that of GC (Zgarbova et al., 2014). Therefore, by choosing GC base as the port of the sequence, the DNA sequence in the solution can achieve the desired structure. Based on the above, the Close-ending constraint is proposed. Assuming that X is a DNA sequence and X_i is the i th base in the DNA sequence, then

$$X = x_1x_2 \dots x_i(x_i, x_i \in \{G, C\}) \quad (21)$$

PSC

In solution, DNA reacts in accordance with the principle of base pairing. When the optimized sequences are put into the solution, there will be a reaction between different sequences. Seven optimized sequences in the article (Chaves-González and Martínez-Gil, 2019) were put into NUPACK for evaluation. The sequence a: AACCAACCTCCACACCGAACA reacted with sequence b: TGGTGTGCTGGTGTAGGTT, and $i(ab) = 0.57/2 = 0.285$. The structural results are shown in **Figure 2**.

As shown in **Figure 2**, sequences a and b reacted in solution to form another structure. $i = 0.285$ means that the sequences react with each other in the solution. In DNA computing, if a DNA sequence in the solution reacts because of the complementary base pairs, the proportion of the DNA sequence in the solution will be reduced, which affects the accuracy of DNA computing. To solve for the reaction between DNA sequences in solution, the PSC is proposed. Different from the Hamming distance constraint, this constraint adds a sliding method to ensure that

every base in the sequence can be traversed. Compared with the H-measure, the comparison between the original sequence and inverse sequence has lower complexity, and the method is simple. Take the a and b sequences as an example. The H-measure calculated values are all 17, while the PSC calculated values are 0. The sequence set, which is constrained by the PSC, reduces the probability of interaction between sequences. This constraint is expressed by the formula:

$$f(L) = \sum_{i=1}^n \sum_{j=i+1}^n \text{Indep}(L_i, L_j') \quad (22)$$

where n is the number of DNA sequences in the L set; L_i is the i th sequence; L_j is the j th sequence; and L_j' is the inverse sequence of L_j . x and y represent two different sequences, and the function $cbp(x, y, i)$ represents the length of a continuous base pair starting from the i th base of the sequence, where x' represents the sequence of five consecutive bases in the x sequence and y' represents the sequence of five consecutive bases in the y sequence. The value of M is four. The value of M is explained in the **Supplementary Material**.

$$\text{Indep}(x, y) = \begin{cases} x, & d = 0 \\ 0, & d > M \end{cases} \quad (23)$$

$$d(x, y) = \sum_{a=1}^{m-4} \sum_{b=1}^{m-4} T(cb p(x'_a, y'_b, i), M) \quad (24)$$

ALGORITHM

The Whale Optimization Algorithm and Chaos Map

The whale optimization algorithm (WOA) (Mirjalili and Lewis, 2016), a kind of meta starting algorithm, is an effective swarm intelligence optimization algorithm. Compared with other group optimization algorithms, the WOA algorithm has the advantages of simple structure and less adjustment parameters. The predation method is to select a random or the current optimal whale position to simulate the behavior of whale predation. The main inspiration of the whale algorithm design is from the unique whale predation method: bubble net predation. In 2016, Mirjalili et al. simulated the predatory behavior of whales with the contraction closed mechanism and spiral update position and selected the random number p as the boundary of the two behaviors. To ensure the scientificity and fairness of the data, 0.5 was taken as the threshold value.

When $p < 0.5$, the whales use the contraction closure mechanism to prey. In particular, we need to determine the absolute value of A in relation to 1. If the absolute value of A is less than 1, select the current optimal whale position to simulate whale hunting behavior; otherwise, the random position of the whale is selected to simulate the predatory behavior of the whale. It is expressed as follows:

$$\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a} \quad (25)$$

$$\vec{X}(t+1) = \begin{cases} \vec{X}^*(t) - \vec{A} \cdot \vec{D} & |A| < 1 \\ X_{rand} - \vec{A} \cdot \vec{D} & |A| \geq 1 \end{cases} \quad (26)$$

and

$$\vec{C} = 2 \cdot \vec{r} \quad (27)$$

$$\vec{D} = \begin{cases} |\vec{C} \cdot \vec{X}^*(t) - \vec{X}(t)| & |A| < 1 \\ |\vec{C} X_{rand} - \vec{X}| & |A| \geq 1 \end{cases} \quad (28)$$

where \vec{a} decreases from 2 to 0 as the number of iterations increases; and \vec{r} is a random number from 0 to 1.

When $p \geq 0.5$, to further expand the scope of whale predation, the whale algorithm is improved by adding a new predator-prey method. The added predator-prey mechanism is somersault foraging by learning manta ray simulation. The general predation formula is as follows:

$$\vec{X}(t+1) = \begin{cases} \vec{D}' \cdot e^{l \cos(2\pi l)} + \vec{X}^*(t) & c < 0.5 \\ \vec{X}(t) + S \cdot (r_1 \cdot \vec{X}^*(t) - r_2 \cdot \vec{X}(t)) & c \geq 0.5 \end{cases} \quad (29)$$

where \vec{D}' represents the distance between the current whale and the optimal whale; $l \in [-1, 1]$; the default value of S is 2; and r_1, r_2 , and c are random numbers between 0 and 1,

$$\vec{D}' = |\vec{X}'(t) - \vec{X}| \quad (30)$$

Chaos has the characteristics of randomness, regularity, and ergodicity. When solving the function optimization problem, the diversity of the population can be maintained, and the global search ability can be improved. Tent (Shan et al., 2005) has better ergodic uniformity and can improve the search speed of the algorithm, and it can generate more evenly distributed values between [0,1]. The formula is as follows:

$$z_{i+1} = \begin{cases} z_i & 0 \leq z_i < 0 \\ \frac{1-z_i}{1-u} & u \leq z_i \leq 1 \end{cases} \quad (31)$$

When $u = 1/2$, tent is the most classical form. The sequence of this form has uniform distribution and approximately uniform distribution density for different parameters. Therefore, the formula of the tent chaotic map in this paper is

$$z_{i+1} = \begin{cases} 2z_i & 0 \leq z_i < \frac{1}{2} \\ 2(1-z_i) & \frac{1}{2} \leq z_i \leq 1 \end{cases} \quad (32)$$

The sine cosine algorithm is a new intelligent optimization algorithm proposed by Mirjalili in 2016. It is based on the mathematical model of outward sine and cosine wave or the wave in the direction of the optimal solution. Using multiple random variables and adaptive variables to calculate the current solution location, different regions in the space can be searched, effectively avoiding local optimization and converging to the global optimum.

ICW Optimization Algorithm

Based on the above algorithm introduction, the ICW optimization algorithm is proposed. To expand the predator-prey of whales, somersault foraging method was added. In this strategy, the position of the candidate solution is regarded as a pivot. Each individual tends to somersault around the pivot to a new position. Therefore, each individual always updates its surrounding location until it finds the best location so far. Because the swarm intelligence algorithm has the disadvantage of falling into local optimization, in order to obtain better global optimization ability and increase the search range, the sine cosine mathematical model is introduced, which makes the optimization direction fluctuate outward or to the direction of the optimal solution. The chaos is added to the sine cosine model to further expand the coverage of the solution space. This makes it easy for the algorithm to escape from the local optimal solution, thus maintaining the diversity of the population and improving the global search ability. In general, this algorithm achieves the desired outcome.

The details of the algorithm are as follows:

- Step 1. Introduce the parameters and generate the initial population.
- Step 2. Calculate the fitness value of the current population and find the optimal solution, which is the minimum fitness value.
- Step 3. Obtain the parameters for every iteration.
- Step 4. Generate random number p and determine the relationship between p and 0.5. If $p < 0.5$, enter step 5; otherwise, enter step 6.
- Step 5. Generate $|A|$ and determine the relationship between $|A|$ and 1. If $|A| < 1$, select the current optimal location for the update operation. Otherwise, select a random location for the update operation.
- Step 6. Generate c and determine the relationship between c and 0.5. If $c < 0.5$, use the spiral upward mechanism to update the position; otherwise, use the somersault mode to update the position.
- Step 7. Use sines and cosines with chaos to increase the global search capability and get the set of desirable populations.
- Step 8. Select populations that satisfy the constraint combinations.
- Step 9. Increase the number of iterations to determine the relationship between the current iteration number and the maximum iteration number. If the current iteration number is less than the maximum, step 2 is performed; otherwise, step 10 is performed.
- Step 10. Calculate the fitness function of the desirable populations and select the optimal populations as the final result.

In this work, the conversion method between numbers and letters is as follows: 0-C, 1-T, 2-A, 3-G.

The flow chart of the ICW optimization algorithm is shown in **Figure 3**, and the **Figure 4** presents the pseudo-code of a general implementation.

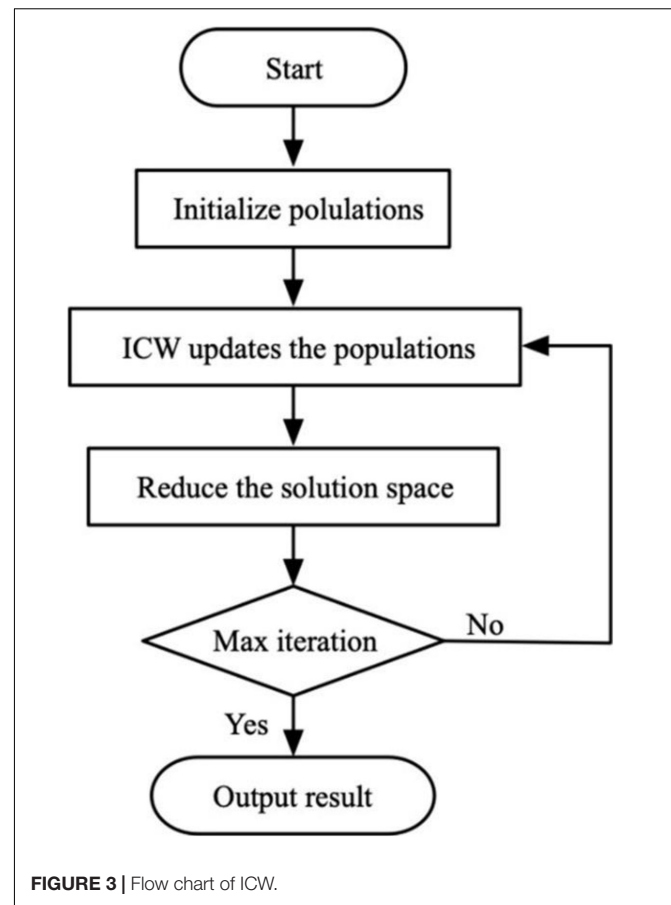


FIGURE 3 | Flow chart of ICW.

Benchmark Test Functions

To better demonstrate the performance of the ICW optimization algorithm, we tested 23 benchmark functions that are widely used. It is worth noting that because each algorithm focuses on solving different types of problems, not every algorithm can get the minimum value of all functions.

For the 23 test functions, the functions can be divided into three categories according to the function types (Digalakis and Margaritis, 2001), namely, unimodal benchmark function (F1–F7), multimodal benchmark function (F8–F13), and fixed-dimension multimodal benchmark function (F14–F23). The equations of the different types of functions are listed in the **Supplementary Material**. In the table of the **Supplementary Material**, Function represents benchmark functions; Dim represents function dimension; rang represents the definition domain value of Function; and f_{min} represents the optimal solution of Function. The unimodal benchmark function has only one global optimal value, so it can be used to test the benchmark development ability of the algorithm. The multimodal benchmark function has one optimal value and several local optimal values. The number of optimal values increases with the increase of the dimension, so it can be used to test the exploration ability of the algorithm and its ability to jump out of the local optimum. Like the multimodal benchmark function, the fixed-dimension multimodal function has only

```

Inputs: The population size N and maximum number of
iterations t
Outputs: DNA sequences that satisfy combination constraint
Initialize the random population Xi (i=1,2...N)
While(t<T)
  Calculate the fitness values of each agent.
  if 1 (p<0.5)
    if 2(|A|<1)
      Update the position of the current search agent by the
      Eq.(26)
    else if 2(|A|≥1)
      Select a random search agent(Xrand)
      Update the position of the current search agent by the
      Eq.(26)
    end if 2
  else if 1(p≥0.5)
    if 3(c<0.5)
      Update the position of the current search by the Eq.(29)
    else if 3(c≥0.5)
      Update the position of the current search by the Eq.(29)
    end if 3
  end if 1
  Expand population by sines and cosines with chaos
End while

```

FIGURE 4 | Pseudo-code of the ICW optimization algorithm.

one global optimal value and many local optimal solutions. However, the solution space of the multi-modal function with fixed dimension is very small, so the step size should be adjusted adaptively.

We compared our algorithm with the algorithms HSWOA (Li et al., 2020), WOA (Mirjalili and Lewis, 2016), GA (Heidari et al., 2019), PSO (Yin et al., 2020), FPA (Yang et al., 2014), GWO (Mirjalili et al., 2014), FA (Gandomi et al., 2011), MFO (Mirjalili, 2015), TLBO (Rao et al., 2011), and DE (Yin et al., 2020). We compared the average (AVG) and standard deviation (STD) of 23 benchmark functions. Other conditions remain unchanged, and each function is iterated 500 times and run 30 times. This operation ensures the validity of the test. The test results are shown in **Tables 1, 2**. **Table 1** shows the results of F1–F13 functions calculated by different algorithms, and **Table 2** shows the results of the remaining functions.

As shown in **Tables 1, 2**, the ICW optimization algorithm is better than the other algorithms in most function values. In particular, our algorithm achieves the optimal values for unimodal functions F1–F4, F9, F11 and F16, F18, F19 of the multimodal functions. In addition, in the functions F5, F7, F10, F11, F15–F17, F19, F21, and F22, the improved algorithm is significantly better than the other algorithms in the table. With respect to the original algorithm WOA, by comparing the average value of the two algorithms, it can be found that the improved ICW optimization algorithm significantly enhances the development ability and gives results that are closer to the global optimization. For standard deviation, the new algorithm has better stability. Compared with the other

functions in the table, the ICW optimization algorithm occupies nearly 75% of the optimal value. So, this algorithm has a strong competitive advantage.

To further illustrate that the improved algorithm has more advantages than the WOA algorithm, the single peak test function, multiple test function, and fixed image of the multimodal function are drawn. We do not select the function that reaches the optimal value, which ensures universal optimization of the new algorithm and enhances the persuasiveness. As shown in the figure, F7, F12, and F20 are selected **Figure 5**. The unimodal benchmark function F7 can jump out of the local optimal solution and converge to the global optimum. The multimodal benchmark function F12 and the fixed-dimension multimodal benchmark function F20 can converge to the local optimal solution quickly.

RESULTS AND ANALYSIS

In this article, the results are obtained by running the algorithm in MATLAB R2018a. The computer had the Windows 10 operating system, Intel (R) CPU 2.70 GHz, and ARM 8.00 GB. The minimum stem length and ring length of the hairpin structure were 6. The minimum continuity threshold was set to 2. In the continuous case, the threshold of similarity and H-measure were six, and the threshold of discontinuous similarity and discontinuous H-measure were 0.17. In addition, T_m was obtained by using the proximity model. The concentration of DNA was set at 10 nM, and salt concentration was set at 1 M.

This part compares ICW with other algorithms, namely NACST/Seq (Shin et al., 2005), DEPT (Chaves-González and Vega-Rodríguez, 2014), MO-ABC (Chaves-González et al., 2013), pMO-ABC (Chaves-González and Martínez-Gil, 2019), and HSWOA (Li et al., 2020). We compare the average values of the above algorithms for continuity, hairpin structure, H-measure, similarity, and melting temperature. We also input the optimal sequence of the algorithm into NUPACK for experimental simulation and compared the simulation results.

Table 3 compares the sequences obtained by our proposed algorithm with those of other algorithms. Seven sequences were used in **Table 3**; each sequence has 20 bases. According to the average values of continuity, hairpin structure, H-measure, similarity, and T_m in **Table 3**, **Figures 6–9**, and **Table 4**, the results show that our work outperforms other algorithms in terms of continuity and hairpin structure. In the aspect of H-measure, the results obtained by our algorithm are much better than other algorithms, which indicates that our algorithm can effectively avoid non-specific hybridization. In the reaction solution, the DNA sequence can also maintain the maximum value. In general, it can effectively avoid non-specific hybridization between sequences. The GC content is always maintained at 50%, representing that the sequences obtained by our algorithm have stable thermodynamic properties.

In **Figure 6**, the result shows the average fitness of continuity and hairpin for different algorithms. From the bar graph, it can be

TABLE 1 | Result of benchmark functions (F1–F13) with 30 dimensions.

ID	Metric	ICW	HSWOA	WOA	GA	PSO	FPA	GWO	FA	MFO	TLBO	DE
F1	AVG	0.00E+00	2.71E–91	1.41E–03	1.03E+03	1.83E+04	2.01E+03	1.18E–27	7.11E–03	1.01E+03	2.17E–89	1.33E–03
	STD	0.00E+00	1.24E–90	4.91E–30	5.79E+02	3.01E+03	5.60E+02	1.47E–27	3.21E–03	3.05E+03	3.14E–89	5.92E–04
F2	AVG	0.00E+00	7.03E–58	1.06E–21	2.47E+01	3.58E+02	3.22E+01	9.71E–17	4.34E–01	3.19E+01	2.77E–45	6.83E–03
	STD	0.00E+00	2.94E–57	2.93E–21	5.68E+00	1.35E+03	5.55E+00	5.60E–17	1.84E–01	2.06E+01	3.11E–45	2.06E–03
F3	AVG	0.00E+00	1.11E+04	5.39E–07	2.65E+04	4.05E+04	1.41E+03	5.12E–05	1.66E+03	2.43E+04	3.91E–18	3.97E+04
	STD	0.00E+00	4.54E+03	2.93E–06	3.44E+03	8.21E+03	5.59E+02	2.03E–04	6.72E+02	1.41E+04	8.04E–18	5.37E+03
F4	AVG	0.00E+00	3.22E+01	0.07E+00	5.17E+01	4.39E+01	2.38E+01	1.24E–06	1.11E–01	7.00E+01	1.68E–36	1.15E+01
	STD	0.00E+00	0.07E+01	0.39E+00	1.05E+01	3.64E+00	2.77E+00	1.94E–06	4.75E–02	7.06E+00	1.47E–36	2.37E+00
F5	AVG	2.54E+01	2.76E+01	2.78E+01	1.95E+04	1.96E+07	3.17E+05	2.70E+01	7.97E+01	7.35E+03	2.54E+01	1.06E+02
	STD	0.25E0+00	0.58E+00	0.76E+00	1.31E+04	6.25E+06	1.75E+05	7.78E–01	7.39E+01	2.26E+04	4.26E–01	1.01E+02
F6	AVG	0.11E+00	0.33E+00	3.11E+00	9.01E+02	1.87E+04	1.70E+03	8.44E–01	6.94E–03	2.68E+03	3.29E–05	1.44E–03
	STD	0.03E+00	0.18E+04	0.53E+00	2.84E+02	2.92E+03	3.13E+02	3.18E–01	3.61E–03	5.84E+03	8.65E–05	5.38E–04
F7	AVG	6.19E–05	1.29E–03	1.42E–03	1.91E–01	1.07E+01	3.41E–01	1.70E–03	6.62E–02	4.50E+00	1.16E–03	5.24E–02
	STD	6.37E–05	1.20E–03	1.14E–03	1.50E–01	3.05E+00	1.10E–01	1.06E–03	4.23E–02	9.21E+00	3.63E–04	1.37E–02
F8	AVG	–1.23E+04	–1.11E+04	–5.08E+03	–1.26E+04	–3.86E+03	–6.45E+03	–5.97E+03	–5.85E+03	–8.48E+03	–7.76E+03	–6.82E+03
	STD	5.26E+02	1.50E+03	6.95E+02	4.51E+00	2.49E+02	3.03E+02	7.10E+02	1.16E+03	7.98E+02	1.04E+03	3.94E+02
F9	AVG	0.00E+00	9.01E+00	0.00E+00	9.04E+00	2.87E+02	1.82E+02	2.19E+00	3.82E+01	1.59E+02	1.40E+01	1.58E+02
	STD	0.00E+00	1.44E+01	0.00E+00	4.58E+00	1.95E+01	1.24E+01	3.69E+00	1.12E+01	3.21E+01	5.45E+00	1.17E+01
F10	AVG	8.88E–16	2.78E–15	7.40E+00	1.36E+01	1.75E+01	7.14E+00	1.03E–13	4.58E–02	1.74E+01	6.45E–15	1.21E–02
	STD	0.00E+00	1.80E–15	9.89E+00	1.51E+00	3.67E–01	1.08E+00	1.70E–14	1.20E–02	4.95E+00	1.79E–15	3.30E–03
F11	AVG	0.00E+00	0.00E+00	2.89E–04	1.01E+01	1.70E+02	1.73E+01	4.76E–03	4.23E–03	3.10E+01	0.00E+00	3.52E–02
	STD	0.00E+00	0.00E+00	1.58E+03	2.43E+00	3.17E+01	3.63E+00	8.57E–03	1.29E–03	5.94E+01	0.00E+00	7.20E–02
F12	AVG	5.06E–03	6.90E–00	3.39E–01	4.77E+00	1.51E+07	3.05E+02	4.83E–02	3.13E–04	2.46E+02	7.35E–06	2.25E–03
	STD	2.54E–03	7.06E–00	0.21E+00	1.56E+00	9.88E+06	1.04E+03	2.12E–02	1.76E–04	1.21E+03	7.45E–06	1.70E–03
F13	AVG	0.10E–00	4.52E–00	1.88E+00	1.52E+01	5.73E+07	9.59E+04	5.96E–01	2.08E–03	2.73E+07	7.89E–02	9.12E–03
	STD	9.40E–02	9.14E–00	3.66E+01	4.52E+00	2.68E+07	1.46E+05	2.23E–01	9.62E–04	1.04E+08	8.78E–02	1.16E–02

Bold values represent the optimal value of the function in the table.

TABLE 2 | Results of benchmark functions (F14–F23) with 30 dimensions.

ID	Metric	ICW	HSWOA	WOA	GA	PSO	FPA	GWO	FA	MFO	TLBO	DE
F14	AVG	3.73E-00	2.18E-00	2.11E+00	9.98E-01	1.39E+00	9.98E-01	4.17E+00	3.51E+00	2.74E+00	9.98E-01	1.23E+00
	STD	4.17E-00	2.11E-00	2.49E+00	4.52E-16	4.60E-01	2.00E-04	3.61E+00	2.16E+00	1.82E+00	4.52E-16	9.23E-01
F15	AVG	3.99E-04	5.25E-04	5.72E-04	3.33E-02	1.61E-03	6.88E-04	6.24E-03	1.01E-03	2.35E-03	1.03E-03	5.63E-04
	STD	1.30E-04	2.54E-04	3.24E-04	2.70E-02	4.60E-04	1.55E-04	1.25E-02	4.01E-04	4.92E-03	3.66E-03	2.81E-04
F16	AVG	-1.03E+00	-1.03E+00	-1.03E-00	-3.78E-01	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00
	STD	1.94E-10	1.32E-08	4.20E-07	3.42E-01	2.95E-03	6.78E-16	6.78E-16	6.78E-16	6.78E-16	6.78E-16	6.78E-16
F17	AVG	3.98E-01	3.97E-01	3.97E-01	5.24E-01	4.00E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01
	STD	8.20E-07	1.19E-07	2.70E-05	6.06E-02	1.39E-03	1.69E-16	1.69E-16	1.69E-16	1.69E-16	1.69E-16	1.69E-16
F18	AVG	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.10E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00
	STD	1.65E-14	5.98E-07	4.22E-15	0.00E+00	7.60E-02	0.00E+00	4.07E-05	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F19	AVG	-3.86E+00	-3.86E+00	-3.85E+00	-3.42E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00
	STD	6.85E-06	3.19E-03	2.70E-03	3.03E-01	1.24E-03	3.16E-15	3.14E-03	3.16E-15	1.44E-03	3.16E-15	3.16E-15
F20	AVG	-3.2821	-3.27	-2.98105	-1.61351	-3.11088	-3.2951	-3.25866	-3.28105	-3.23509	-3.24362	-3.27048
	STD	0.057049	0.060296	0.376653	0.46049	0.029126	0.019514	0.064305	0.063635	0.064223	0.15125	0.058919
F21	AVG	-10.142	-8.7129	-7.04918	-6.66177	-4.14764	-5.21514	-8.64121	-7.67362	-6.8859	-8.64525	-9.64796
	STD	0.011404	2.4655	3.629551	3.732521	0.919578	0.008154	2.563356	3.50697	3.18186	1.76521	1.51572
F22	AVG	-10.3907	-8.1948	-8.18178	-5.58399	-6.01045	-5.34373	-10.4014	-9.63827	-8.26492	-10.2251	-9.74807
	STD	0.015752	2.9941	3.29202	2.605837	1.962628	0.053685	0.000678	2.293901	3.076809	0.007265	1.987703
F23	AVG	-10.527	-8.7416	-9.34238	-4.69882	-4.72192	-5.29437	-10.0836	-9.75489	-7.65923	-10.0752	-10.5364
	STD	0.011063	2.8113	2.414737	3.256702	1.742618	0.356377	1.721889	2.345487	3.576927	1.696222	8.88E-15

Bold values represent the optimal value of the function in the table.

TABLE 3 | Comparing sequences from NACS/Seq, DEPT, MO-ABC, pMO-ABC, and HSWOA.

Seq.	C	P	H	S	Tm	GC%	Seq.	C	P	H	S	Tm	GC%
Our work							NACST/Seq [21]						
CCTCTCCATCCTTATCCTTC	0	0	35	66	60.88	50	CTCTCATCTCTCCGTTCTTC	0	0	37	58	61.43	50
CCAGACCAATACAGAACCAC	0	0	50	57	62.54	50	TATCCTGTGGTGTCTTCTCT	0	0	45	57	64.46	50
CTCCTCTTCTCCTTCTTCTC	0	0	28	82	60.72	50	GTATTCCAAGCGTCCGTGTT	0	0	55	49	65.29	50
CACAACCAATCACTCTCACC	0	0	38	65	63.10	50	TCTCTTACGTTGGTTGGCTG	0	0	51	53	64.63	50
CCACCTGACCGACTAATAAC	0	0	48	61	62.02	50	CTCTTCATCCACCTCTTCTC	0	0	43	58	61.38	50
CCAACCACTCTTCTACAACC	0	0	37	72	62.47	50	ATTCTGTCCGTTGCGTGTC	0	0	52	56	65.82	50
CCTTCTTCTCTCTCTCTCTC	0	0	30	75	60.13	50	AAACCTCCACCAACACACCA	9	0	55	43	66.71	50
DEPT [23]							MO-ABC [22]						
CCATTCCTTAACCTCTCTCC	0	0	59	39	61.39	50	GTAAGGAAGGCAAGGCAGAA	0	0	42	54	64.70	50
ACACACACACACACACAC	0	0	27	49	65.85	50	GTTGGTGGTTGTTGGTGGTT	0	0	46	36	66.00	50
GGAAGGAGGAGGAAGAAGAA	0	0	37	45	62.83	50	GGAGACGGAATGGAAGAGTA	0	0	44	55	62.93	50
GAGAGAAGAGAAGAGGCCAA	0	0	39	53	63.17	50	CCATTCTTCTCTTCTCTCCC	9	0	67	22	61.39	50
ACCACAACAACAACACACCC	9	0	29	55	65.96	50	AGGAGAGGAGAGGAGGAAAA	16	0	31	53	63.80	50
GGAGCAATGGAGAATAAGGG	9	0	48	47	62.42	50	ATAAGAGAGAGAGAGAGGGG	16	0	34	51	61.11	50
CCATACCAGCCAACCGAAAA	16	0	42	56	65.33	50	GAGCCAACAGCCAACCAAAA	16	0	48	45	66.40	50
pMO-ABC [28]							HSWOA [36]						
GGTGGTATTGGTGGTATTGG	0	0	47	47	62.64	50	CTCGTCTAACCTTCTTCAGC	0	0	63	51	62.28	50
CTTCTCTTCTCTTGCCGCTT	0	0	39	56	64.70	50	CTGTGTGGAATGCAAGGATG	0	0	64	48	63.82	50
CTCTCTCTCTCACTCTCTCA	0	0	41	48	61.32	50	CGAGCGTAGTGTAGTCATCA	0	0	63	69	63.56	50
AACAACCTCCACACCGAACA	0	0	62	32	66.69	50	AGTTACAGGACACCACCGAT	0	0	65	51	66.39	50
TGTGGTTGGTTAGTCGGTTG	0	0	46	49	63.80	50	CAGTAGCAGTCATAACGAGC	0	0	64	56	62.69	50
TGGTGTTGCTGGTGTAGGTT	0	0	48	51	66.46	50	GCATAGCACATCGTAGCGTA	0	0	59	54	64.60	50
CTCTCATTCCTTCTTACCCC	16	0	43	51	61.40	50	TGGACCTTGAGAGTGGAGAT	0	0	62	50	64.44	50

TABLE 4 | Comparing the melting temperatures of the various algorithm sequences.

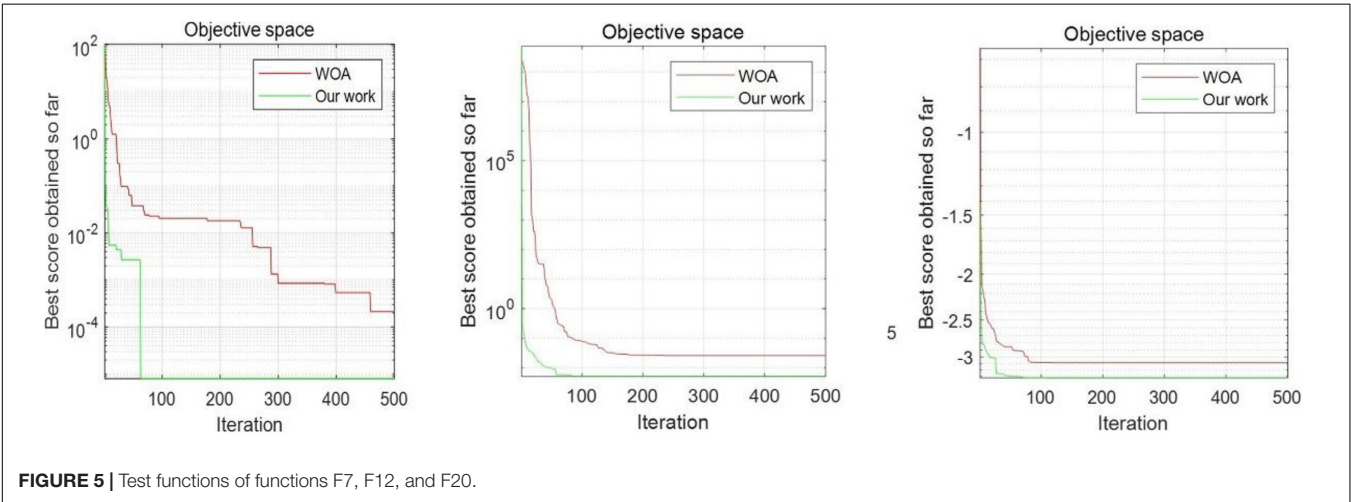
	ICW	NACST/Seq	DEPT	MO-ABC	pMO-ABC	HSWOA
Var	1.24	4.33	3.37	4.37	4.91	1.86

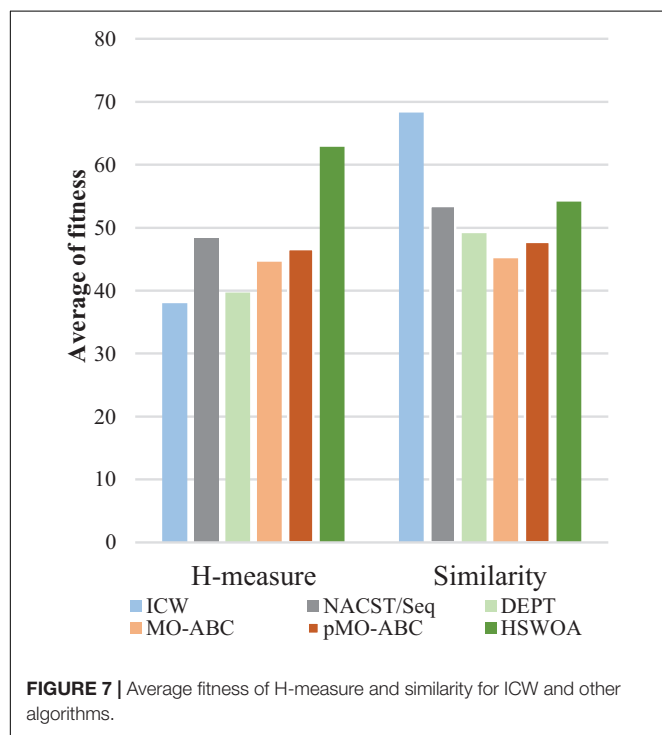
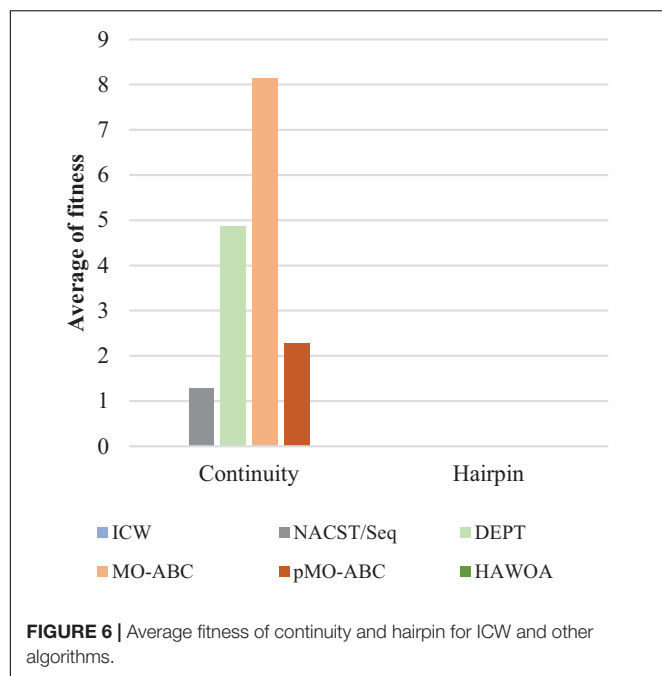
clearly seen that the sequence obtained by the ICW optimization algorithm was better than that obtained by other algorithms in terms of continuity and hairpin structure. ICW optimization

algorithm obtained the minimum value. Therefore, the sequence obtained by our algorithm can effectively avoid the generation of secondary structures.

Using **Table 3**, we calculated the average fitness of different algorithms for H-measure and similarity (**Figure 7**). It can be clearly seen from the graph that our algorithm obtains the minimum H-measure. However, our algorithm is different from other comparable algorithms in terms of the similarity.

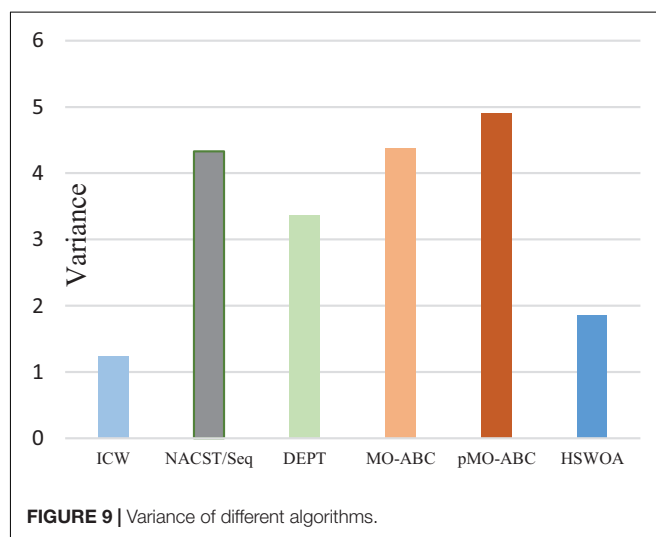
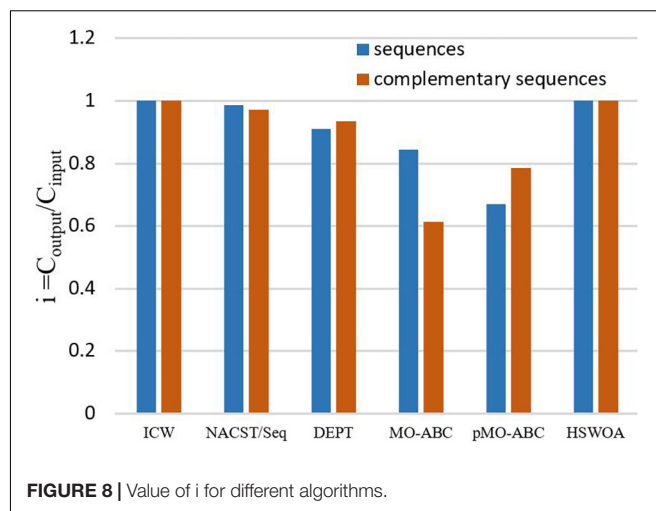
After comparing the above average fitness values, we also studied the evaluation results in NUPACK. Most DNA solution





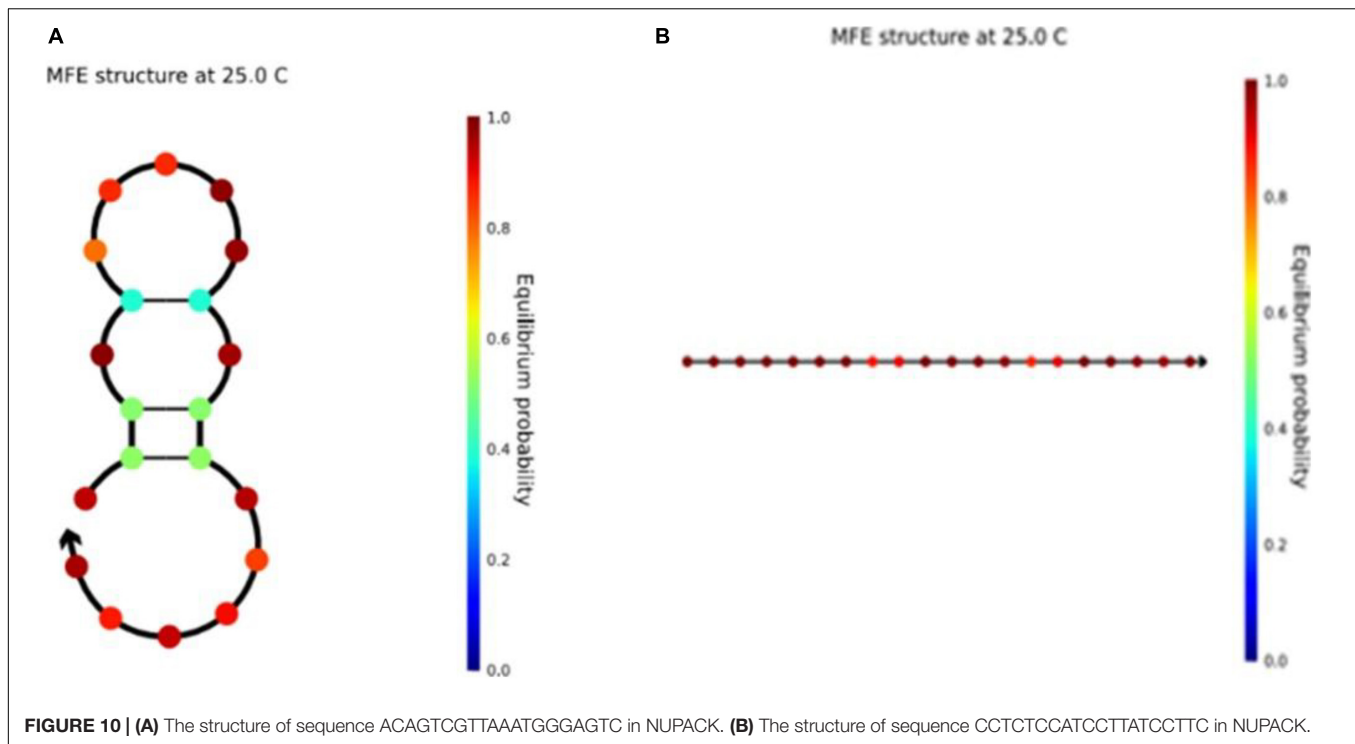
experiments are performed at room temperature, so the temperature was set at 25°C. We input the seven sequences (optimal sequences of each algorithm) of different algorithms in **Table 3** into NUPACK. After evaluation, the values of C_{input} and C_{output} were calculated, and the value of i was obtained, as shown in **Figure 8**.

Figure 8 shows the values of i for different algorithms. It should be noted that the closer i is to 1, the higher is the



quality of the sequences. In the figure, each algorithm has two columnar regions. The first represents seven sequences of the algorithms in **Table 3**, and the second column region represents the complementary sequence of these seven sequences. First, consider the sequences in **Table 3**. In the histogram, ICW and HSWOA, which is our previous work, both have the value 1 for i . In the HSWOA algorithm, the triplet-bases unpaired constraint is added to form a constraint combination to control a single sequence so that the sequence does not react with itself. In the ICW optimization algorithm, PSC is added to form a new constraint combination to reduce the reaction between different sequences. From the value of i , the new algorithm with constraint is better performing than other algorithms.

In addition, considering the double stranded structure of DNA, the complements of seven DNA sequences were evaluated. The evaluation results are shown in **Figure 8**. In the figure, the values of the ICW optimization algorithm and HSWOA algorithm are 1, indicating that there is no reaction between complementary sequences, which is what we expected. In addition, seven DNA sequences and



complements were input into NUPACK for evaluation, finding the Fourteen sequences were complete reactions, which were complementary to each other.

Table 4 and **Figure 9** show the variance of the melting temperature of different algorithms. The values in the table are calculated from the T_m values in **Table 4**. In DNA computing, to ensure the consistency of the biochemical reactions of DNA molecules, all the DNA molecules participating in the biochemical reactions should be uniform. In other words, if the variance is small, the melting temperature change will be small, and the probability of achieving the desired result will be increased.

The significance of this work is evaluated again by the variance of melting temperature (T_m) and the value of i . In order to ensure the consistency of the work, in the comparison of indicators, the control group still chooses sequence a and sequence b and compares with the average value of the indicators in this work. The variance of the T_m of the DNA sequence obtained by the pMO-ABC algorithm is 1.65, while the variance of the T_m in this work is 1.24, which is reduced by 33.1%. Smaller variance of T_m is more conducive to controlling the temperature during the reaction, and smaller temperature fluctuation is more conducive to the reaction. The i value of the two sequences of a and b is $i = 0.285$, while the i of the set of DNA sequences in this work is all 1. The closer i is to 1, the higher the sequence quality.

It is worth noting that the sequence structure of the unstable available structure in the previous article (Li et al., 2020) is shown in **Figure 10A**, and this structure has also been changed in this work. Seven optimized sequences satisfying the new combinatorial constraints were input into NUPACK at the same

time. We found that all the sequences were linear structures. They were stable and available structures that could improve the accuracy of DNA calculation. As shown in the figure, the color bar on the right shows the stability of the sequence. The closer the color is to red, the more stable it is. Compared with the two graphs, the sequence structure of **Figure 10B** is more stable. In the graph of the two DNA sequences, the figures represent the stable structure of the sequences. In addition, other DNA sequence structures obtained from our work are presented in the **Supplementary Material**.

CONCLUSION

In this study, we input existing DNA sequences into NUPACK and evaluated them. It was found that the DNA sequences may react with each other owing to base complementary pairing. Therefore, we propose a new constraint, PSC, to solve this problem. In addition, due to the double strand structure of DNA, if the A-T base is located at one end of the DNA sequence and its complementary sequence, there may be a gap, which leads to a decrease of the accuracy of calculation. The proposed Close-ending constraint can effectively avoid the generation of such sequences. These two new constraints were fused into the previous constraint combination to form a new combination constraint. Then, the new ICW optimization algorithm was used to obtain the sequences satisfying the new combination constraints, and the sequence results were analyzed. The analysis results show that the current minimum is obtained on continuity, hairpin, and H-measure. This shows that the sequence greatly improves the ability to avoid secondary structures. In terms

of the T_m value, the minimum variance was obtained by calculation, which ensured that the DNA molecules participating in biochemical reactions were more uniform and improved the thermodynamic stability. When the sequences were input into NUPACK for evaluation, the concentration of the obtained sequence in the solution was the same as before, indicating that the DNA sequence did not react with itself or other sequences in the solution, and the DNA sequence was stable and available in the solution, which improved the accuracy of DNA calculation.

In the future, to produce stable DNA sequence and ensure the accuracy of DNA computing, we will take the optimization of the DNA sequence set as the primary task. With respect to the algorithm, we will further optimize it. At the same time, reducing the similarity of DNA sequence sets and exploring the linear structure of DNA sequences in solution will also be two important aspects. In addition, to expand the application of DNA computing, we will also work with the machine learning (Song et al., 2019; Gong et al., 2020) and bioinformatics (Zou et al., 2020).

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

REFERENCES

- Adleman, L. M. (1994). Molecular computation of solutions to combinatorial problems. *Science* 266, 1021–1024. doi: 10.1126/science.7973651
- Benenson, Y., Gil, B., Ben-Dor, U., Adar, R., and Shapiro, E. (2004). An autonomous molecular computer for logical control of gene expression. *Nature* 429, 423–429. doi: 10.1038/nature02551
- Cao, B., Li, X., Zhang, X., Wang, B., Zhang, Q., and Wei, X. (2020). Designing uncorrelated address constrain for DNA storage by DMVO algorithm. *IEEE/ACM Trans. Comput. Biol. Bioinform.* doi: 10.1109/tcbb.2020.3011582
- Cao, B., Zhang, X., Wu, J., Wang, B., Zhang, Q., and Wei, X. (2021). Minimum free energy coding for DNA storage. *IEEE Trans. Nanobiosci.* 20, 212–222. doi: 10.1109/tnb.2021.3056351
- Chaves-González, J. M. (2015). Hybrid multiobjective metaheuristics for the design of reliable DNA libraries. *J. Heuristics* 21, 751–788. doi: 10.1007/s10732-015-9298-x
- Chaves-González, J. M., and Martínez-Gil, J. (2019). An efficient design for a multi-objective evolutionary algorithm to generate DNA libraries suitable for computation. *Interdiscip. Sci.* 11, 542–558. doi: 10.1007/s12539-018-0303-6
- Chaves-González, J. M., and Vega-Rodríguez, M. A. (2014). DNA strand generation for DNA computing by using a multi-objective differential evolution algorithm. *Biosystems* 116, 49–64. doi: 10.1016/j.biosystems.2013.12.005
- Chaves-González, J. M., Vega-Rodríguez, M. A., and Granado-Criado, J. M. (2013). A multiobjective swarm intelligence approach based on artificial bee colony for reliable DNA sequence design. *Eng. Appl. Artif. Intell.* 26, 2045–2057. doi: 10.1016/j.engappai.2013.04.011
- Cherry, K. M., and Qian, L. (2018). Scaling up molecular pattern recognition with DNA-based winner-take-all neural networks. *Nature* 559:370. doi: 10.1038/s41586-018-0289-6
- Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T. (2002). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. *IEEE Trans. Evol. Comput.* 6, 182–197.
- Deng, L., Wang, Y., Noor-A-Rahim, M., Guan, Y. L., Shi, Z., Gunawan, E., et al. (2019). Optimized code design for constrained DNA data storage with asymmetric errors. *IEEE Access* 7, 84107–84121. doi: 10.1109/access.2019.2924827

AUTHOR CONTRIBUTIONS

XL and ZW designed the DNA sequences, analyzed the data and the performance, and wrote the manuscript. BW and TS supervised the work, evaluated the performance, and revised the manuscript. All authors have read and agreed to the published version of the manuscript.

FUNDING

This work is supported by the National Key Technology R&D Program of China (No. 2018YFC0910500), the National Natural Science Foundation of China (Nos. 61425002, 61751203, 61772100, 61972266, 61802040, and 61672121), the High-level Talent Innovation Support Program of Dalian City (Nos. 2017RQ060 and 2018RQ75), and the Innovation and Entrepreneurship Team of Dalian University (No. XQN202008).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/fgene.2021.644484/full#supplementary-material>

- Digalakis, J. G., and Margaritis, K. G. (2001). On benchmarking functions for genetic algorithms. *Int. J. Comput. Math.* 77, 481–506. doi: 10.1080/00207160108805080
- Gaborit, P., and King, O. D. (2005). Linear constructions for DNA codes. *Theor. Comput. Sci.* 334, 99–113. doi: 10.1016/j.tcs.2004.11.004
- Gandomi, A. H., Yang, X.-S., and Alavi, A. H. (2011). Mixed variable structural optimization using firefly algorithm. *Comput. Struct.* 89, 2325–2336. doi: 10.1016/j.compstruc.2011.08.002
- Gong, F. M., Ma, Y. H., Zheng, P., and Song, T. (2020). A deep model method for recognizing activities of workers on offshore drilling platform by multistage convolutional pose machine. *J. Loss Prev. Process Ind.* 64:104043. doi: 10.1016/j.jlp.2020.104043
- Han, D. R., Qi, X. D., Myhrvold, C., Wang, B., Dai, M. J., Jiang, S. X., et al. (2017). Single-stranded DNA and RNA origami. *Science* 358:eaao2648.
- Heidari, A. (2014). An investigation of the role of DNA as molecular computers: a computational study on the Hamiltonian path problem. *Int. J. Sci. Eng. Res.* 5, 1884–1889.
- Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., and Chen, H. (2019). Harris hawks optimization: algorithm and applications. *Future Gener. Comput. Syst.* 97, 849–872. doi: 10.1016/j.future.2019.02.028
- Li, S., Jiang, Q., Liu, S., Zhang, Y., Tian, Y. H., Song, C., et al. (2018). A DNA nanorobot functions as a cancer therapeutic in response to a molecular trigger in vivo. *Nat. Biotechnol.* 36, 258–264. doi: 10.1038/nbt.4071
- Li, X., Wang, B., Lv, H., Yin, Q., Zhang, Q., and Wei, X. (2020). Constraining DNA sequences with a triplet-bases unpaired. *IEEE Trans. Nanobiosci.* 19, 299–307. doi: 10.1109/tnb.2020.2971644
- Mirjalili, S. (2015). Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. *Knowl. Based Syst.* 89, 228–249. doi: 10.1016/j.knsys.2015.07.006
- Mirjalili, S. (2016). SCA: a sine cosine algorithm for solving optimization problems. *Knowl. Based Syst.* 96, 120–133. doi: 10.1016/j.knsys.2015.12.022
- Mirjalili, S., and Lewis, A. (2016). The whale optimization algorithm. *Adv. Eng. Softw.* 95, 51–67.
- Mirjalili, S., Mirjalili, S. M., and Lewis, A. (2014). Grey wolf optimizer. *Adv. Eng. Softw.* 69, 46–61.

- Palluk, S., Arlow, D. H., De Rond, T., Barthel, S., Kang, J. S., Bector, R., et al. (2018). De novo DNA synthesis using polymerase-nucleotide conjugates. *Nat. Biotechnol.* 36:645. doi: 10.1038/nbt.4173
- Rao, R. V., Savsani, V. J., and Vakharia, D. (2011). Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. *Comput. Aided Des.* 43, 303–315. doi: 10.1016/j.cad.2010.12.015
- Rinaudo, K., Bleris, L., Maddamsetti, R., Subramanian, S., Weiss, R., and Benenson, Y. (2007). A universal RNAi-based logic evaluator that operates in mammalian cells. *Nat. Biotechnol.* 25, 795–801. doi: 10.1038/nbt1307
- Schickinger, M., Zacharias, M., and Dietz, H. (2018). Tethered multifluorophore motion reveals equilibrium transition kinetics of single DNA double helices. *Proc. Natl. Acad. Sci. U.S.A.* 115, E7512–E7521.
- Shan, L., Qiang, H., Li, J., and Wang, Z.-Q. (2005). Chaotic optimization algorithm based on Tent map. *Control Decis.* 20, 179–182.
- Shendure, J., Balasubramanian, S., Church, G. M., Gilbert, W., Rogers, J., Schloss, J. A., et al. (2017). DNA sequencing at 40: past, present and future. *Nature* 550, 345–353. doi: 10.1038/nature24286
- Shin, S. Y., Lee, I. H., Kim, D., and Zhang, B. T. (2005). Multiobjective evolutionary optimization of DNA sequences for reliable DNA computing. *IEEE Trans. Evol. Comput.* 9, 143–158. doi: 10.1109/tevc.2005.844166
- Shipman, S. L., Nivala, J., Macklis, J. D., and Church, G. M. (2017). CRISPR-Cas encoding of a digital movie into the genomes of a population of living bacteria. *Nature* 547:345. doi: 10.1038/nature23017
- Song, T., Pan, L., Wu, T., Zheng, P., Wong, M. D., and Rodríguez-Patón, A. (2019). Spiking neural P systems with learning functions. *IEEE Trans. NanoBioscience* 18, 176–190. doi: 10.1109/tnb.2019.2896981
- Sze, M. A., and Schloss, P. D. (2019). The impact of DNA polymerase and number of rounds of amplification in PCR on 16S rRNA gene sequence data. *Mosphere* 4, e00163–19.
- Thubagere, A. J., Li, W., Johnson, R. F., Chen, Z., Doroudi, S., Lee, Y. L., et al. (2017). A cargo-sorting DNA robot. *Science* 357:eaan6558. doi: 10.1126/science.aan6558
- Wang, Y., Jaime-Lara, R., Roy, A., Sun, Y., Liu, X., and Joseph, P. V. (2020). SeqEnhDL: sequence-based classification of cell type-specific enhancers using deep learning models. *bioRxiv* [Preprint] doi: 10.1101/2020.05.13.093997
- Wang, Y., Noor-A-Rahim, M., Gunawan, E., Guan, Y. L., and Poh, C. L. (2019). Construction of bio-constrained code for DNA data storage. *IEEE Commun. Lett.* 23, 963–966. doi: 10.1109/lcomm.2019.2912572
- Wang, Y., Shen, Y., Zhang, X., Cui, G., and Sun, J. (2018). An improved non-dominated sorting genetic algorithm-II (NSGA-II) applied to the design of DNA codewords. *Math. Comput. Simul.* 151, 131–139. doi: 10.1016/j.matcom.2018.03.011
- Yang, G., Wang, B., Zheng, X., Zhou, C., and Zhang, Q. (2017). IWO algorithm based on niche crowding for DNA sequence design. *Interdiscip. Sci.* 9, 341–349. doi: 10.1007/s12539-016-0160-0
- Yang, X.-S., Karamanoglu, M., and He, X. (2014). Flower pollination algorithm: a novel approach for multiobjective optimization. *Eng. Optim.* 46, 1222–1237. doi: 10.1080/0305215x.2013.832237
- Yin, Q., Cao, B., Li, X., Wang, B., Zhang, Q., and Wei, X. (2020). An intelligent optimization algorithm for constructing a DNA storage code: NOL-HHO. *Int. J. Mol. Sci.* 21:2191. doi: 10.3390/ijms21062191
- Zeng, X., Lin, Y., He, Y., Lu, L., Min, X., and Rodriguez-Paton, A. (2020a). Deep collaborative filtering for prediction of disease genes. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 17, 1639–1647.
- Zeng, X., Zhong, Y., Lin, W., and Zou, Q. (2020b). Predicting disease-associated circular RNAs using deep forests combined with positive-unlabeled learning methods. *Brief. Bioinform.* 21, 1425–1436. doi: 10.1093/bib/bbz080
- Zgarbova, M., Otyepka, M., Sponer, J., Lankas, F., and Jurecka, P. (2014). Base pair fraying in molecular dynamics simulations of DNA and RNA. *J. Chem. Theory Comput.* 10, 3177–3189. doi: 10.1021/ct500120v
- Zhang, S., Huang, B., Song, X., Zhang, T., Wang, H., and Liu, Y. (2019). A high storage density strategy for digital information based on synthetic DNA. *3 Biotech* 9:342.
- Zhang, X., Zhang, Q., Liu, Y., Wang, B., and Zhou, S. (2020). A molecular device: a DNA molecular lock driven by the nicking enzymes. *Comput. Struct. Biotec.* 18, 2107–2116. doi: 10.1016/j.csbj.2020.08.004
- Zhao, W., Zhang, Z., and Wang, L. (2020). Manta ray foraging optimization: an effective bio-inspired optimizer for engineering applications. *Eng. Appl. Artif. Intell.* 87:103300. doi: 10.1016/j.engappai.2019.103300
- Zhou, S., He, P., and Kasabov, N. (2020). A dynamic DNA Color image encryption method based on SHA-512. *Entropy* 22:1091. doi: 10.3390/e22101091
- Zhu, E., Jiang, F., Liu, C., and Xu, J. (2020). Partition independent set and reduction based approach for partition coloring problem. *IEEE T. Cybern.* doi: 10.1109/TCYB.2020.3025819
- Zou, Q., and Liu, Q. (2019). Advanced machine learning techniques for bioinformatics. *IEEE Comput. Archit. Lett.* 16, 1182–1183. doi: 10.1109/tcbb.2019.2919039
- Zou, Q., Lin, G., Jiang, X., Liu, X., and Zeng, X. (2020). Sequence clustering in bioinformatics: an empirical study. *Brief. Bioinform.* 21, 1–10.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2021 Li, Wei, Wang and Song. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.



NMFNA: A Non-negative Matrix Factorization Network Analysis Method for Identifying Modules and Characteristic Genes of Pancreatic Cancer

Qian Ding¹, Yan Sun¹, Junliang Shang^{1*}, Feng Li¹, Yuanyuan Zhang² and Jin-Xing Liu¹

¹ School of Computer Science, Qufu Normal University, Rizhao, China, ² School of Information and Control Engineering, Qingdao University of Technology, Qingdao, China

OPEN ACCESS

Edited by:

Pan Zheng,
University of Canterbury, New Zealand

Reviewed by:

Shanfeng Zhu,
Fudan University, China
Tao Song,
Polytechnic University of Madrid,
Spain

*Correspondence:

Junliang Shang
shangjunliang110@163.com

Specialty section:

This article was submitted to
Computational Genomics,
a section of the journal
Frontiers in Genetics

Received: 10 March 2021

Accepted: 03 June 2021

Published: 22 July 2021

Citation:

Ding Q, Sun Y, Shang J, Li F,
Zhang Y and Liu J-X (2021) NMFNA:
A Non-negative Matrix Factorization
Network Analysis Method
for Identifying Modules
and Characteristic Genes
of Pancreatic Cancer.
Front. Genet. 12:678642.
doi: 10.3389/fgene.2021.678642

Pancreatic cancer (PC) is a highly fatal disease, yet its causes remain unclear. Comprehensive analysis of different types of PC genetic data plays a crucial role in understanding its pathogenic mechanisms. Currently, non-negative matrix factorization (NMF)-based methods are widely used for genetic data analysis. Nevertheless, it is a challenge for them to integrate and decompose different types of genetic data simultaneously. In this paper, a non-NMF network analysis method, NMFNA, is proposed, which introduces a graph-regularized constraint to the NMF, for identifying modules and characteristic genes from two-type PC data of methylation (ME) and copy number variation (CNV). Firstly, three PC networks, i.e., ME network, CNV network, and ME–CNV network, are constructed using the Pearson correlation coefficient (PCC). Then, modules are detected from these three PC networks effectively due to the introduced graph-regularized constraint, which is the highlight of the NMFNA. Finally, both gene ontology (GO) and pathway enrichment analyses are performed, and characteristic genes are detected by the multimeasure score, to deeply understand biological functions of PC core modules. Experimental results demonstrated that the NMFNA facilitates the integration and decomposition of two types of PC data simultaneously and can further serve as an alternative method for detecting modules and characteristic genes from multiple genetic data of complex diseases.

Keywords: pancreatic cancer, non-negative matrix factorization, module, network analysis, characteristic gene

INTRODUCTION

Pancreatic cancer (PC) is a highly fatal disease of the digestive system and it is becoming an increasingly common cause of cancer mortality, yet its pathogenic mechanisms remain unclear (Mizrahi et al., 2020). Therefore, comprehensively analyzing multiple types of PC genetic data to understand its pathogenic mechanisms has become a hot topic and many studies have been conducted. For instance, Wu et al. (2011) applied the lasso penalized Cox regression to transcriptome data to identify genes that are directly related to PC survival. Yang et al. (2013) identified thousands of differentially expressed genes of PC and then six genes were predicted

to be involved in PC development. Gong et al. (2014) integrated pathway information into PC survival analysis and applied the doubly regularized Cox regression model to microarray data to identify both PC-related genes and pathways. Kwon et al. (2015) used the support vector machine to evaluate the diagnostic performance of PC biomarkers based on miRNA and mRNA expression data. Tao et al. (2016) performed a comprehensive search of electronic literature sources to evaluate the association between *K-ras* gene mutations and PC survival. Li et al. (2018) identified two hub genes of PC from the integrated microarray data and then validated them in RNA-sequencing data by *k*-nearest neighbor and random forest algorithms. These studies provided several underlying biomarkers and can help cancer researchers design new strategies for the early detection and diagnosis of PC (Gong et al., 2014).

Currently, non-negative matrix factorization (NMF)-based methods are widely used for genetic data analysis. For example, Mishra and Guda (2016) applied the NMF to genome-scale methylome analysis of PC data and detected three distinct molecular subtypes. Wang et al. (2013) proposed the maximum correntropy criterion-based NMF (NMF-MCC) method for cancer clustering on gene expression (GE) data. Zhao et al. (2018) used the NMF bi-clustering method to identify subtypes of pancreatic ductal adenocarcinoma, which is the most common type of PC. Xiao et al. (2018) proposed the graph-regularized NMF to discover potential associations between miRNAs and diseases. These methods show that the NMF is a powerful tool for genetic data analysis. Nevertheless, it is a challenge for them to integrate and decompose different types of genetic data simultaneously. Zhang et al. (2012) adopted the joint NMF (jNMF) method to address this challenge, which projects multiple types of genomic data onto a common coordinate system, and applied the jNMF to the methylation (ME), GE, and miRNA expression data of ovarian cancer to identify cancer-related multidimensional modules. Yang and Michailidis (2016) introduced the integrative NMF (iNMF) to analyze multimodal data, which includes a sparsity option for jointly decomposing heterogeneous data, and also evaluated the iNMF on ME, GE, and miRNA expression data of ovarian cancer. These integrated NMF methods can reveal pathogenic mechanisms that would have been overlooked with only a single type of data, and uncover associations between different layers of cellular activities (Zhang et al., 2012).

However, most of these NMF-based methods only consider individual genetic effects and ignore interaction effects among different features. It has been widely accepted that interaction effects could unveil a large portion of unexplained pathogenic mechanisms of cancers (Ding et al., 2019). For capturing these interaction effects, several NMF-based network analysis methods have been proposed due to network facilitating presenting interactions between features. Liu et al. (2014) developed a network-assisted co-clustering algorithm for the identification of cancer subtypes, which first assigns weights to genes based on their impact in the network, and then utilizes the non-negative matrix trifactorization (TriNMF) to cluster cancer patients (Ding et al., 2006). Chen and Zhang adopted the NMF framework in a

network manner (NetNMF) to integrate pairwise genomic data for identifying two-level modular patterns and the relationships among these modules (Chen and Zhang, 2018). Elyanow et al. (2020) proposed the netNMF-sc method to cluster cells based on prior knowledge of gene–gene interactions. Nevertheless, the netNMF-sc ignored interaction effects among different features and used the decomposed submatrix to construct the network, which might weaken the internal connection between nodes in the network. Gao et al. (2019) proposed the integrated graph-regularized NMF (iGMFNA) model for clustering and network analysis of cancers, which decomposes the integrated data into submatrices for constructing networks. Zheng et al. (2019) used the NMF to integrate ME and copy number variation (CNV) networks for identifying prognostic biomarkers in ovarian cancer. These NMF-based network analysis methods provide new insights into the pathogenic mechanisms of cancers, especially their interaction effects.

Inspired by both integration and network-assisted strategies of the NMF, in this paper, we presented a NMF network analysis method, NMFNA for short, based on graph-regularized constraint, to identify modules and characteristic genes from integrated ME and CNV data of PC. Firstly, the Pearson correlation coefficient (PCC) is employed to construct three PC networks, i.e., ME network, CNV network, and ME–CNV network. Then, these networks are further integrated and decomposed simultaneously to identify modules effectively due to the introduced graph-regularized constraint, which is the highlight of the NMFNA. Finally, both gene ontology (GO) and pathway enrichment analyses are performed, and characteristic genes are detected by the multimeasure score, to deeply understand biological functions of PC core modules. Experimental results demonstrated that the NMFNA facilitates the integration and decomposition of two types of PC data simultaneously and can further serve as an alternative method for detecting modules and characteristic genes from multiple genetic data of complex diseases.

METHODS

Non-negative Matrix Factorization Methods

The NMF (Lee and Seung, 1999) and its variants have been increasingly applied to identify modules in biological networks (Chen and Zhang, 2018; Wang et al., 2018; Gao et al., 2019). For a biological network $\mathbf{X}^{m \times n}$, the NMF can decompose it into two non-negative matrices $\mathbf{U}^{m \times k}$ and $\mathbf{V}^{k \times n}$, such that $\mathbf{X} \approx \mathbf{U}\mathbf{V}$, where $k < \min(m, n)$. The Euclidean distance between \mathbf{X} and its approximation matrix \mathbf{UV} is applied to minimize the factorization error, which can be written as,

$$\begin{aligned} \min_{\mathbf{U}, \mathbf{V}} \|\mathbf{X} - \mathbf{UV}\|_F^2 \\ \text{s.t. } \mathbf{U} \geq 0, \mathbf{V} \geq 0 \end{aligned} \quad (1)$$

where $\|\cdot\|_F^2$ is the Frobenius norm of a matrix. Since it is difficult to find a global minimal solution by optimizing the convex and

non-linear objective function, the NMF adopts the multiplicative iterative update algorithm to approximate \mathbf{U} and \mathbf{V} ,

$$u_{ik} \leftarrow u_{ik} \frac{(\mathbf{XV}^T)_{ik}}{(\mathbf{UVV}^T)_{ik}} \quad (2)$$

$$v_{kj} \leftarrow v_{kj} \frac{(\mathbf{U}^T \mathbf{X})_{kj}}{(\mathbf{U}^T \mathbf{UV})_{kj}} \quad (3)$$

In addition to the two-factor NMF, the three-factor NMF also plays an important role in matrix factorization, which constrains scales of \mathbf{U} and \mathbf{V} by an extra factor \mathbf{S} , i.e., $\mathbf{X} \approx \mathbf{USV}$. This factored matrix \mathbf{S} not only provides an additional degree of freedom to make the approximation tight, but also indicates associations between identified modules (Chen and Zhang, 2018). A three-factor NMF variant TriNMF (Ding et al., 2006) is defined as,

$$\min_{\mathbf{U}, \mathbf{V}} \|\mathbf{X} - \mathbf{USV}\|_F^2 \quad (4)$$

$$s.t. \mathbf{U} \geq 0, \mathbf{S} \geq 0, \mathbf{V} \geq 0$$

which minimizes the objective function by,

$$u_{ik} \leftarrow u_{ik} \frac{(\mathbf{XVS}^T)_{ik}}{(\mathbf{UU}^T \mathbf{XVS}^T)_{ik}} \quad (5)$$

$$v_{kj} \leftarrow v_{kj} \frac{(\mathbf{X}^T \mathbf{US})_{kj}}{(\mathbf{VV}^T \mathbf{X}^T \mathbf{US})_{kj}} \quad (6)$$

$$s_{kk} \leftarrow s_{kk} \frac{(\mathbf{U}^T \mathbf{XV})_{kk}}{(\mathbf{U}^T \mathbf{USV}^T \mathbf{V})_{kk}} \quad (7)$$

Particularly, if \mathbf{X} is the symmetric similarity matrix, it could be decomposed into \mathbf{USU}^T . For pairwise biological networks with the same samples but two types of features, $\mathbf{X}_1^{m_1 \times n}$ and $\mathbf{X}_2^{m_2 \times n}$, combining the idea of two-factor and three-factor NMF, the NetNMF (Chen and Zhang, 2018) is defined as,

$$\min_{\mathbf{G}_1, \mathbf{G}_2, \mathbf{S}_{11}, \mathbf{S}_{22}} \|\mathbf{R}_{11} - \mathbf{G}_1 \mathbf{S}_{11} \mathbf{G}_1^T\|_F^2 + \alpha \|\mathbf{R}_{12} - \mathbf{G}_1 \mathbf{G}_2^T\|_F^2 + \beta \|\mathbf{R}_{22} - \mathbf{G}_2 \mathbf{S}_{22} \mathbf{G}_2^T\|_F^2 \quad (8)$$

$$s.t. \mathbf{G}_1 \geq 0, \mathbf{G}_2 \geq 0, \mathbf{S}_{11} \geq 0, \mathbf{S}_{22} \geq 0$$

where $\mathbf{R}_{11}^{m_1 \times m_1}$ and $\mathbf{R}_{22}^{m_2 \times m_2}$ are the symmetric feature similarity matrices of \mathbf{X}_1 and \mathbf{X}_2 , respectively, that is, their respective co-expression networks; $\mathbf{R}_{12}^{m_1 \times m_2}$ is the two-type feature similarity matrix (co-expression network) between \mathbf{X}_1 and \mathbf{X}_2 ; $\mathbf{G}_1^{m_1 \times k}$ and $\mathbf{G}_2^{m_2 \times k}$ are the non-negative factored matrices used for identifying modules in their respective networks; $\mathbf{S}_{11}^{k \times k}$ and $\mathbf{S}_{22}^{k \times k}$ are also symmetric matrices whose diagonal elements can be used for measuring associations between identified modules; k is the user prespecified dimension parameter; α and β are used to balance three terms of the objective function and default settings are m_1/m_2 and $(m_1/m_2)^2$, respectively (Chen and Zhang, 2018). The NetNMF minimizes the objective function by,

$$(g_1)_{ik} \leftarrow (g_1)_{ik} \frac{(2\mathbf{R}_{11} \mathbf{G}_1 \mathbf{S}_{11} + \alpha \mathbf{R}_{12} \mathbf{G}_2)_{ik}}{(2\mathbf{G}_1 \mathbf{S}_{11} \mathbf{G}_1^T \mathbf{G}_1 \mathbf{S}_{11} + \alpha \mathbf{G}_1 \mathbf{G}_2^T \mathbf{G}_2)_{ik}} \quad (9)$$

$$(g_2)_{kj} \leftarrow (g_2)_{kj} \frac{(2\beta \mathbf{R}_{22} \mathbf{G}_2 \mathbf{S}_{22} + \alpha \mathbf{R}_{12}^T \mathbf{G}_1)_{kj}}{(2\beta \mathbf{G}_2 \mathbf{S}_{22} \mathbf{G}_2^T \mathbf{G}_2 \mathbf{S}_{22} + \alpha \mathbf{G}_2 \mathbf{G}_1^T \mathbf{G}_1)_{kj}} \quad (10)$$

$$(s_{11})_{kk} \leftarrow (s_{11})_{kk} \frac{(\mathbf{G}_1^T \mathbf{R}_{11} \mathbf{G}_1)_{kk}}{(\mathbf{G}_1^T \mathbf{G}_1 \mathbf{S}_{11} \mathbf{G}_1^T \mathbf{G}_1)_{kk}} \quad (11)$$

$$(s_{22})_{kk} \leftarrow (s_{22})_{kk} \frac{(\mathbf{G}_2^T \mathbf{R}_{22} \mathbf{G}_2)_{kk}}{(\mathbf{G}_2^T \mathbf{G}_2 \mathbf{S}_{22} \mathbf{G}_2^T \mathbf{G}_2)_{kk}} \quad (12)$$

Non-negative Matrix Factorization Network Analysis Method

In order to further improve the capability of identifying modules and capturing interaction effects, we proposed the NMFNA method by introducing the graph-regularized constraint into the NetNMF, which can preserve the inherent geometrical structure of input networks (Deng et al., 2011). For demonstrating its effectiveness, in the study, we applied the NMFNA to two-type PC data of ME and CNV to identify modules and characteristic genes. In fact, the NMFNA is universally useful and can be applied to any type of genetic data in various complex diseases.

The overall workflow of the NMFNA for identifying modules and characteristic genes by integrating ME and CNV data of PC is shown in **Figure 1**. It is seen that the NMFNA mainly has three stages. In the first stage, three co-expression networks are constructed from ME and CNV data of PC. In the second stage, these three networks are applied to the objective function to identify modules. In the third stage, both GO and pathway enrichment analyses are performed, and characteristic genes are detected, to deeply understand biological functions of PC core modules. Among them, the objective function, which introduces the graph-regularized constraint, is the highlight of the NMFNA.

The graph-regularized constraint indicates the inherent geometrical structure of the input networks. In other words, the graph-regularized constraint ensures that interactive features in the Euclidean space are also close to each other in the low-dimensional space, which is defined as,

$$GRC = \frac{1}{2} \sum_{ij} \left\| v_i - v_j \right\|^2 \mathbf{Z}_{ij} = \text{Tr}(\mathbf{G}^T \mathbf{L} \mathbf{G}) \quad (13)$$

where \mathbf{Z} is the sparse weight matrix established using the geometrical information of \mathbf{X} (Xiao et al., 2018), and \mathbf{Z}_{ij} represents the similarity between gene v_i and v_j . $\text{Tr}(\cdot)$ represents the trace of a matrix, and \mathbf{G} is the factored matrix of the biological network \mathbf{X} by the NMF. Define a diagonal matrix \mathbf{D} whose elements are column sums of matrix \mathbf{Z} , that is, $\mathbf{D}_{ii} = \sum_j \mathbf{Z}_{ij}$, and \mathbf{L} is the graph Laplacian matrix defined as:

$$\mathbf{L} = \mathbf{D} - \mathbf{Z}. \quad (14)$$

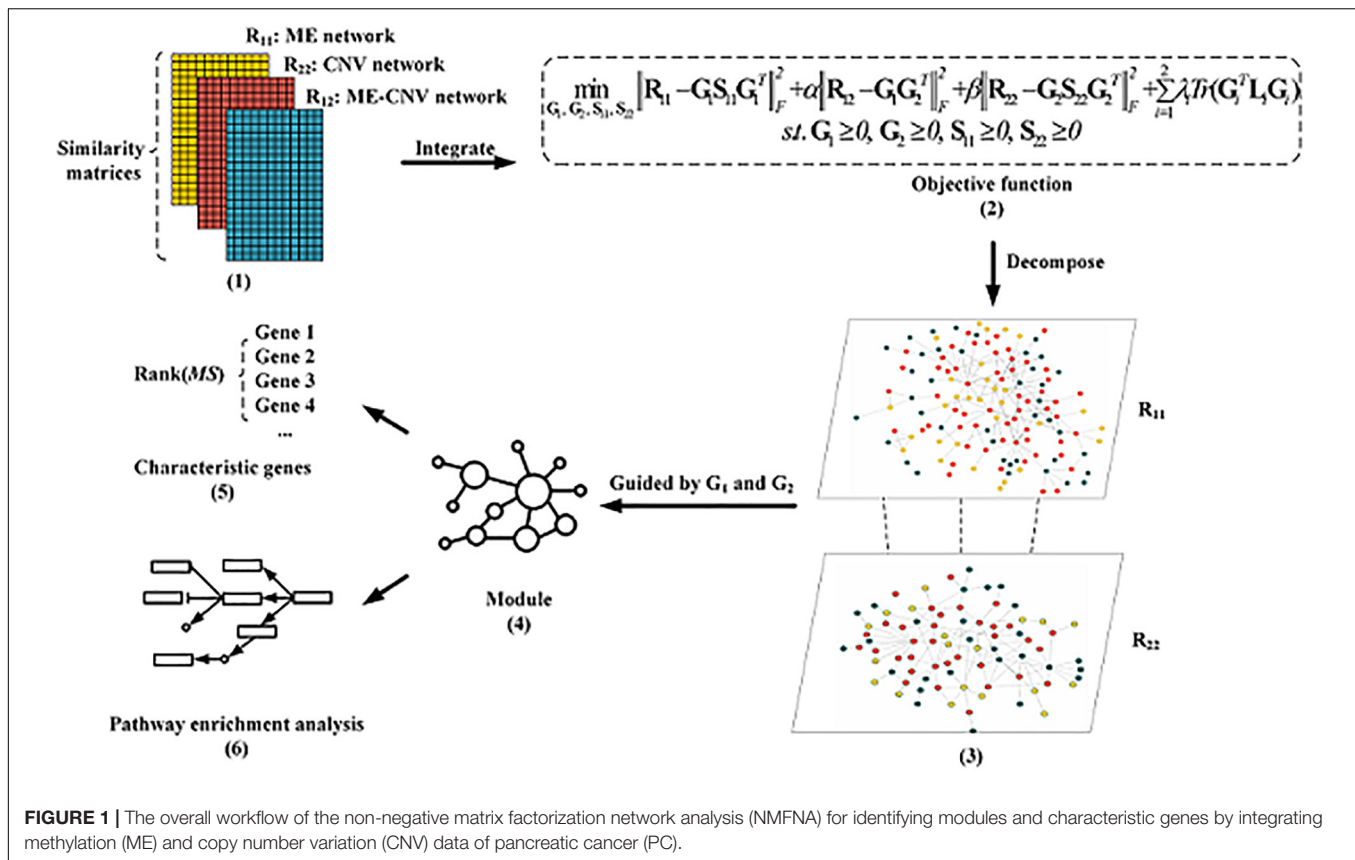


FIGURE 1 | The overall workflow of the non-negative matrix factorization network analysis (NMFNA) for identifying modules and characteristic genes by integrating methylation (ME) and copy number variation (CNV) data of pancreatic cancer (PC).

Based on the NetNMF and the graph-regularized constraint, the objective function of the NMFNA is defined as,

$$\min_{G_1, G_2, S_{11}, S_{22}} \left[\|R_{11} - G_1 S_{11} G_1^T\|_F^2 + \alpha \|R_{12} - G_1 G_2^T\|_F^2 + \beta \|R_{22} - G_2 S_{22} G_2^T\|_F^2 + \sum_{i=1}^2 \lambda_i \text{Tr}(G_i^T L_i G_i) \right] \quad (15)$$

s.t. $G_1 \geq 0, G_2 \geq 0, S_{11} \geq 0, S_{22} \geq 0$

where R_{11} and R_{22} are the ME and CNV co-expression networks, R_{12} is the ME-CN co-expression network, λ is the tuning parameter to adjust the closeness between interactive features, and other notation meanings and parameter settings are the same as those in the NetNMF.

The multiplicative iterative update algorithm is adopted here to minimize the objective function of the NMFNA. Suppose Ψ_1, Ψ_2, Ψ_3 , and Ψ_4 are matrices of Lagrange multipliers that, respectively, constrain $S_{11} \geq 0, S_{22} \geq 0, G_1 \geq 0$, and $G_2 \geq 0$, the Lagrange function f of the NMFNA is,

$$\begin{aligned} f = & \text{tr} \left((R_{11} - G_1 S_{11} G_1^T)^T (R_{11} - G_1 S_{11} G_1^T) \right) + \\ & \alpha \text{tr} \left((R_{12} - G_1 G_2^T)^T (R_{12} - G_1 G_2^T) \right) + \\ & \beta \text{tr} \left((R_{22} - G_2 S_{22} G_2^T)^T (R_{22} - G_2 S_{22} G_2^T) \right) + \\ & \lambda_1 \text{Tr}(G_1^T L_1 G_1) + \lambda_2 \text{Tr}(G_2^T L_2 G_2) \\ & + \text{tr}(\Psi_1^T S_{11}) + \text{tr}(\Psi_2^T S_{22}) + \text{tr}(\Psi_3^T G_1) + \text{tr}(\Psi_4^T G_2) \end{aligned} \quad (16)$$

Hence, partial derivatives of f with respect to S_{11}, S_{22}, G_1 , and G_2 are,

$$\frac{\partial f}{\partial S_{11}} = -2G_1^T R_{11} G_1 + 2G_1^T G_1 S_{11} G_1^T G_1 + \Psi_1 \quad (17)$$

$$\frac{\partial f}{\partial S_{22}} = -2G_2^T R_{22} G_2 + 2G_2^T G_2 S_{22} G_2^T G_2 + \Psi_2 \quad (18)$$

$$\begin{aligned} \frac{\partial f}{\partial G_1} = & 4 \left(G_1 S_{11} G_1^T G_1 S_{11} - R_{11} G_1 S_{11} \right) + 2\alpha \\ & \left(G_1 G_2^T G_2 - R_{12} G_2 \right) + 2\lambda_1 L_1 G_1 + \Psi_3 \end{aligned} \quad (19)$$

$$\begin{aligned} \frac{\partial f}{\partial G_2} = & 4\beta \left(G_2 S_{22} G_2^T G_2 S_{22} - R_{22} G_2 S_{22} \right) + 2\alpha \\ & \left(G_2 G_1^T G_1 - R_{12} G_1 \right) + 2\lambda_2 L_2 G_2 + \Psi_4 \end{aligned} \quad (20)$$

According to Karush-Kuhn-Tucher conditions (Dreves et al., 2011), i.e., $\Psi_1 S_{11} = 0, \Psi_2 S_{22} = 0, \Psi_3 G_1 = 0$, and $\Psi_4 G_2 = 0$, iterative formulas can be written as,

$$(s_{11})_{kk} \leftarrow (s_{11})_{kk} \frac{(G_1^T R_{11} G_1)_{kk}}{(G_1^T G_1 S_{11} G_1^T G_1)_{kk}} \quad (21)$$

$$(s_{22})_{kk} \leftarrow (s_{22})_{kk} \frac{(G_2^T R_{22} G_2)_{kk}}{(G_2^T G_2 S_{22} G_2^T G_2)_{kk}} \quad (22)$$

$$(g_1)_{ik} \leftarrow (g_1)_{ik} \frac{(\alpha \mathbf{R}_{12} \mathbf{G}_2 + 2\mathbf{R}_{11} \mathbf{G}_1 \mathbf{S}_{11} + 2\lambda_1 \mathbf{Z}_1 \mathbf{G}_1)_{ik}}{(2\mathbf{G}_1 \mathbf{S}_{11} \mathbf{G}_1^T \mathbf{G}_1 \mathbf{S}_{11} + \alpha \mathbf{G}_1 \mathbf{G}_2^T \mathbf{G}_2 + 2\lambda_1 \mathbf{D}_1 \mathbf{G}_1)_{ik}} \quad (23)$$

$$(g_2)_{kj} \leftarrow (g_2)_{kj} \frac{(\alpha \mathbf{R}_{12}^T \mathbf{G}_1 + 2\beta \mathbf{R}_{22} \mathbf{G}_2 \mathbf{S}_{22} + \lambda_2 \mathbf{Z}_2 \mathbf{G}_2)_{kj}}{(2\beta \mathbf{G}_2 \mathbf{S}_{22} \mathbf{G}_2^T \mathbf{G}_2 \mathbf{S}_{22} + \alpha \mathbf{G}_2 \mathbf{G}_1^T \mathbf{G}_1 + \lambda_2 \mathbf{D}_2 \mathbf{G}_2)_{kj}} \quad (24)$$

Two types of modules, namely, ME modules and CNV modules, can be identified from \mathbf{R}_{11} and \mathbf{R}_{22} guided by \mathbf{G}_1 and \mathbf{G}_2 , respectively. In particular, \mathbf{G}_1 and \mathbf{G}_2 are first z-score normalized; then for each column $(1, \dots, k)$ of them, those genes whose corresponding weights are greater than or equal to the threshold are considered as a cluster; finally, according to these clusters, subnetworks of \mathbf{R}_{11} and \mathbf{R}_{22} can be captured as ME modules and CNV modules. Here, the threshold is set to be 2 according to the reference (Chen and Zhang, 2018). In addition, similar to previous studies (Hou et al., 2019; Zhao et al., 2020), modules with the most genes are known as core modules.

To identify characteristic genes from core modules, which may play an important role in deeply understanding the biological functions of modules, we employ the multimeasure score (MS) to numerically quantify the importance of each gene, which is defined as,

$$MS(v) = \frac{DC(v) \cdot BC(v)}{EC(v)} \quad (25)$$

where $DC(v)$, $BC(v)$, and $EC(v)$ are the degree centrality, betweenness centrality, and eccentricity centrality of gene v in \mathbf{R}'_{11} and \mathbf{R}'_{22} , which are networks filtered from \mathbf{R}_{11} and \mathbf{R}_{22} , respectively, with edge weights higher than a given threshold. Betweenness centrality and eccentricity centrality focus on the global feature of a gene in the network, while degree centrality focuses on the local feature of a gene in the network (Shang et al., 2019); hence, the MS combines both global and local features of a gene.

RESULTS AND DISCUSSION

Data and Parameter Settings

Two types of PC data, i.e., ME data and CNV data, are downloaded from the TCGA database¹. These two data have the same samples (176 tumor samples and 4 normal samples) but different features: 21,031 methylations in ME data and 23,627 CNVs in CNV data. Based on these PC data, three co-expression networks, i.e., the ME network $\mathbf{R}_{11}^{21,031 \times 21,031}$, the CNV network $\mathbf{R}_{22}^{23,627 \times 23,627}$, and the ME–CNV network $\mathbf{R}_{12}^{21,031 \times 23,627}$, are constructed using the PCC. Besides, to further prove the experimental results of two types of PC data, we also analyze the GEO datasets of PC. Four profile datasets, i.e., GSE62452, GSE15471, GSE16515, and GSE28735, of PC are

downloaded from the GEO database² for this study. Details of these four datasets are shown in **Table 1**.

In the NMFNA, four parameters should be set, which is, tuning parameters λ_1 and λ_2 , the dimension parameter k , and the iteration number. λ reflects the degree of imposed graph-regularized constraint. A large one focuses on reaching consensus across views, while a small one cannot tolerate matrix factorization error (Liu et al., 2013). Since these different items have no distinction of importance, and the dimension reduction parameter k has a greater impact compared with parameter λ , considering the convenience of comparison, both λ_1 and λ_2 are set to be the same value. We run the NMFNA with different λ values ranging from 0 to 0.1 to select the proper one based on the measure of total module similarity (Wang et al., 2018), which is defined as,

$$TMS = \sum_{x,y} \frac{|M_x \cap M_y|}{\min(|M_x|, |M_y|)} \quad (26)$$

where M_x represents members in module x . According to experiment results (**Figure 2A**), λ_1 and λ_2 are set to be 0.03. The dimension parameter k is determined by the singular value decomposition method (Qiao, 2015), and its first inflection point, i.e., 6,834, is selected to k . In order to reduce the decomposition error, a large iteration number is used. In the study, we set it to 200 since the decomposition error here has already reached a relatively stable state (**Figure 2B**).

GO and Pathway Enrichment Analyses

To demonstrate the validity of the NMFNA, we compared it with the NMF, TriNMF, and NetNMF by performing GO and pathway enrichment analyses on their respective identified core modules. The GO enrichment analysis was carried out by an online tool, DAVID Bioinformatics Resources³ (Dennis et al., 2003). The pathway enrichment analysis was conducted using the KOBAS v3.0 web server⁴ (Kanehisa et al., 2016), in which, the KEGG pathway, BioCyc, PANTHER, and Reactome databases were used. The numbers of GO terms and pathways (p -value < 0.05) obtained from enrichment analyses of ME and CNV core modules identified by the compared methods are shown in **Figure 3**. It is seen that either ME or CNV core modules identified by the NMFNA have more GO terms and pathways than those identified by other methods, implying that modules identified by the NMFNA might contain more

²<http://www.ncbi.nlm.nih.gov/geo/>

³<https://david.ncifcrf.gov>

⁴<https://github.com/xmao/kobas>

TABLE 1 | Details of GEO datasets of PC.

Datasets	Normal	Tumor	Genes
GSE62452	61	69	20,358
GSE15471	36	36	22,188
GSE16515	16	36	22,187
GSE28735	45	45	20,314

¹<https://cancergenome.nih.gov/>

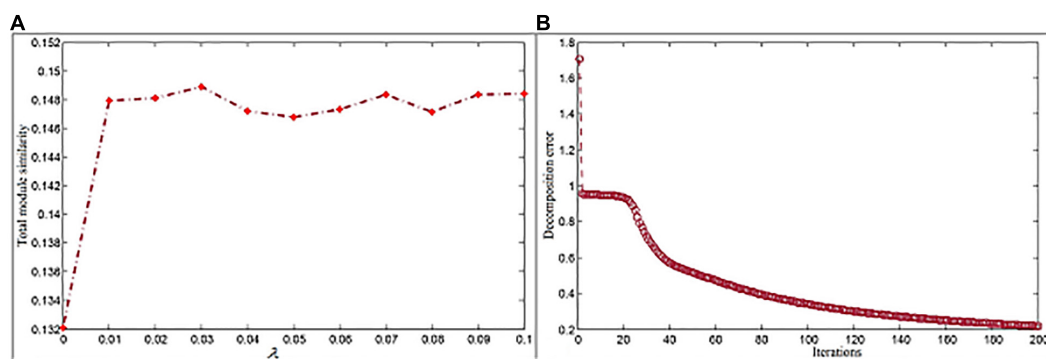


FIGURE 2 | Parameter settings for the NMFNA. (A) is the influence of parameter and (B) is the convergence analysis of NMFNA.

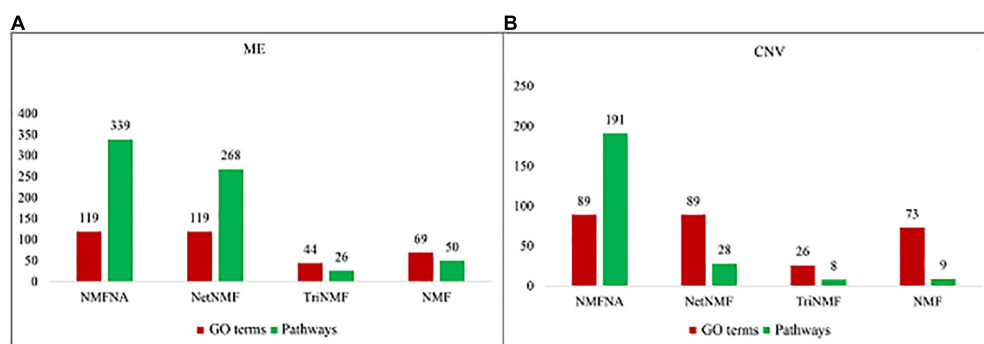


FIGURE 3 | Numbers of GO terms and pathways obtained from enrichment analyses of ME and CNV core modules that identified by compared methods. (A) Represents the details of the ME core module and (B) represents the details of the CNV core module.

biological information related to understanding the pathogenic mechanisms of PC.

The numbers of enriched genes in GO terms obtained from the enrichment analyses of ME and CNV core modules identified by the NMFNA are shown in **Figure 4**. It is seen that for the ME core module, genes are mainly enriched in GO:0005515, GO:0005737, GO:0005829, GO:0070062, GO:0005654, GO:0016020, GO:0005615, and GO:0005739, corresponding to protein binding, cytoplasm, cytosol, extracellular exosome, nucleoplasm, membrane, extracellular space, and mitochondrion; for the CNV core module, genes are mainly enriched in GO:0005515, GO:0006351, GO:0003676, and GO:0005789, corresponding to protein binding, DNA-templated, DNA binding, and endoplasmic reticulum. Details of these significant GO terms are listed in **Table 2**. Several studies have confirmed that these GO terms contribute to the development of PC cells. The protein binding (GO:0005515) is the most significantly enriched GO term among molecular functions in both ME and CNV core modules. As one of the specific binding protein, IGF binding protein-1 has been confirmed to inhibit the activity of insulin-like growth factor I, which has growth-promoting effects on PC cells (Wolpin et al., 2007). The cytoplasm (GO:0005737) plays an important role in the development of PC by regulating the expression of carbonic anhydrase IX (Juhász et al., 2003). As a member of

the cadherin/catenin family, p120(ctn) has been found in the cytosol (GO:0005829) of PC cells (Mayerle et al., 2003), which is correlated to the degree of tumor dedifferentiation. The fractional volume of the extracellular space (GO:0005615) in the PC tissue has been reported to be statistically larger than that in the normal tissue (Yao et al., 2012). The novel mitochondrion interfering compound NPC-26 may effectively inhibit the growth of PC cells by destroying the mitochondria (GO:0005739) (Dong et al., 2016). Genes involved in the DNA-templated (GO:0006351) have been clinically used for treating lung cancer (Lu et al., 2016), which might be speculated to affect other cancers by their pan-cancer co-regulation mechanisms. The nicotine can induce the inhibitor of the DNA binding (GO:0003676) and has been confirmed as an established risk factor for PC (Treviño et al., 2011). The endoplasmic reticulum (GO:0005789) has been identified as the key target in PC, which shows its potential for antitumor drug development (Gajate et al., 2012). Other three GO terms, including extracellular exosome (GO:0070062), nucleoplasm (GO:0005654), and membrane (GO:0016020) also have been reported to associate with PC (Sakai et al., 2019; Zhou et al., 2019).

Among all pathways obtained from enrichment analyses of ME and CNV core modules identified by the compared methods (**Figure 3**), we recorded common pathways that appear in at least three out of four methods in **Table 3**.

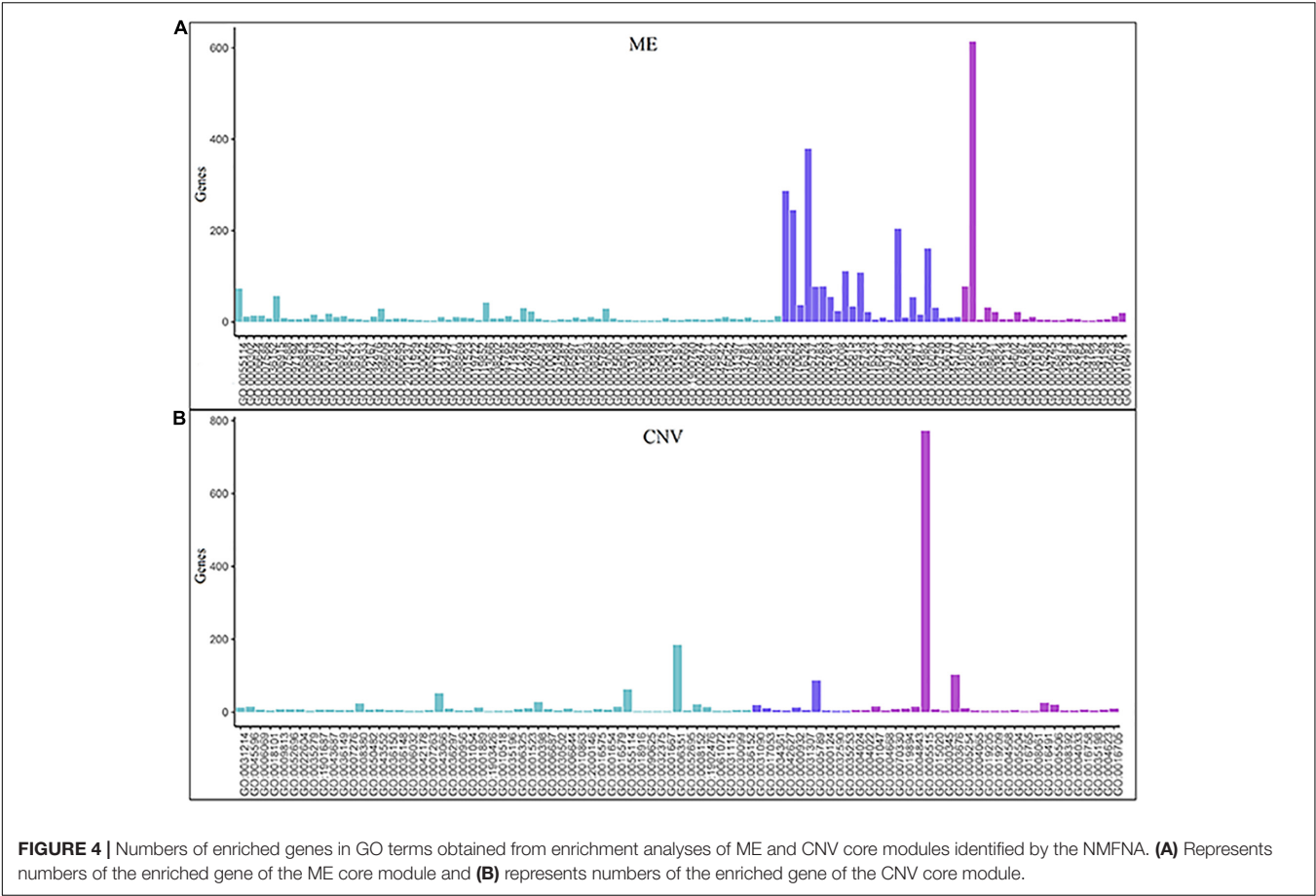


FIGURE 4 | Numbers of enriched genes in GO terms obtained from enrichment analyses of ME and CNV core modules identified by the NMFNA. (A) Represents numbers of the enriched gene of the ME core module and (B) represents numbers of the enriched gene of the CNV core module.

Module	ID	Name	Count	p-value	adj.P
ME, CNV	GO:0005515	Protein binding	614	1.62E-04	4.80E-02
ME	GO:0005737	Cytoplasm	379	3.26E-04	6.93E-07
ME	GO:0005829	Cytosol	287	1.15E-09	5.25E-06
ME	GO:0070062	Extracellular exosome	245	1.74E-08	2.85E-01
ME	GO:0005654	Nucleoplasm	204	8.85E-03	4.77E-01
ME	GO:0016020	Membrane	161	2.13E-02	1.32E-01
ME	GO:0005615	Extracellular space	111	2.10E-03	2.01E-01
ME	GO:0005739	Mitochondrion	108	4.09E-03	9.85E-01
CNV	GO:0006351	Transcription, DNA-templated	185	3.75E-02	7.27E-01
CNV	GO:0003676	Nucleic acid binding	103	1.08E-02	9.26E-01
CNV	GO:0005789	Endoplasmic reticulum membrane	87	2.79E-02	9.53E-02

ME, methylation; CNV, copy number variation.

It is seen that in terms of *p*-values, as well as *p*-values corrected by the false discovery rate (FDR), namely adj.P, the NMFNA performs best among all compared methods, implying that pathways enriched in core modules identified by the NMFNA are more significant than those enriched in core modules identified by other compared methods. To further analyze these core modules, the top 10 pathways according to their adj.P enriched in ME and CNV core

modules identified by the NMFNA are listed in **Figure 5**, in which, the node size and color represent the number of genes enriched in the pathway and the significance of the pathway, respectively. Specifically, two pathways, i.e., transport of small molecules and arachidonic acid metabolism, have already been reported to be associated with PC. The former can filter downregulated differentially expressed genes of PC, while the latter can promote the progress of PC

TABLE 3 | Common pathways appearing in at least three out of four methods.

Pathways	NMFNA		NetNMF		TriNMF		NMF	
	p-value	adj.P	p-value	adj.P	p-value	adj.P	p-value	adj.P
Metabolism	4.58E-16	2.30E-12	3.65E-15	1.86E-11	4.94E-06	1.19E-02	7.70E-04	1.87E-01
Immune system	1.11E-05	4.46E-03	7.72E-06	3.93E-03	1.15E-03	1.33E-01	1.02E-03	1.87E-01
Disease	1.29E-06	1.08E-03	5.27E-04	4.71E-02	1.05E-04	6.63E-02	\	\
Transport of small molecules	6.46E-09	1.08E-05	2.42E-07	3.08E-04	\	\	4.61E-03	3.56E-01
Arachidonic acid metabolism	6.79E-04	5.08E-02	4.62E-03	1.52E-01	4.38E-03	2.13E-01	\	\
Metabolic process	1.27E-04	4.14E-02	4.28E-04	1.26E-01	4.65E-04	4.79E-02	\	\

adj.P is the p-value corrected by the FDR. “\” represents that the pathway cannot be obtained from enrichment analyses of ME and CNV core modules that were identified by the corresponding method.

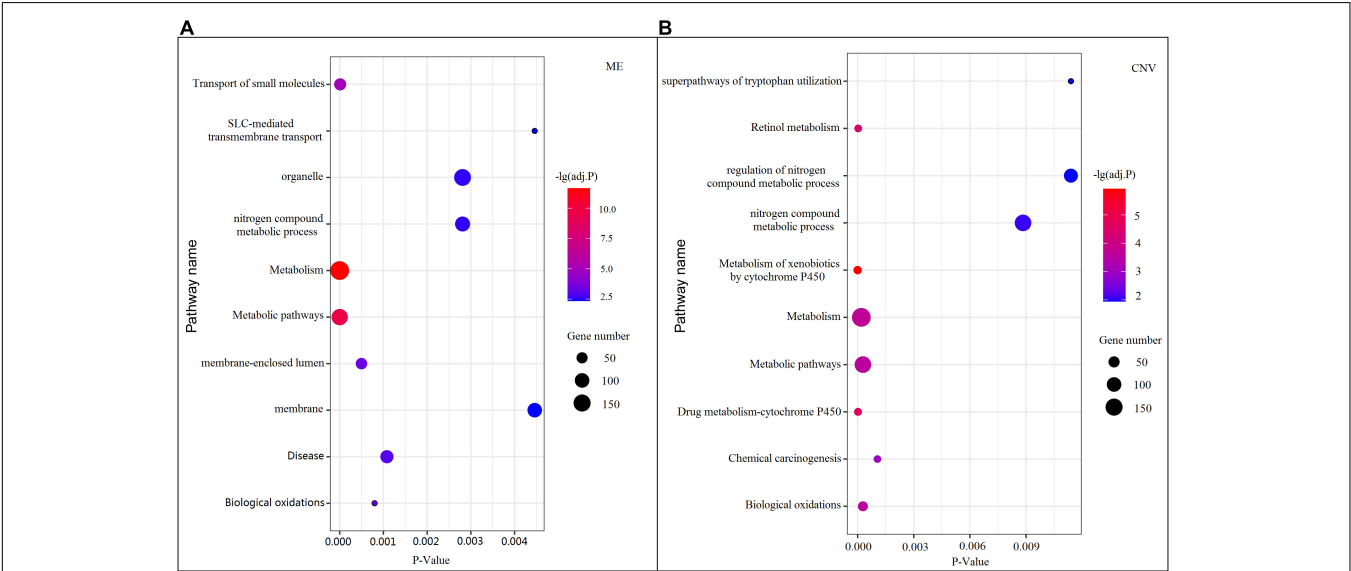


FIGURE 5 | Top 10 pathways enriched in ME and CNV core modules identified by the NMFNA. **(A)** Represents pathways enriched in the ME core module and **(B)** represents pathways enriched in the CNV core module.

(Biswas et al., 2014; Long et al., 2016). To aid early diagnosis, the metabolism pathway can be used individually or in combination to differentiate people with and without PC. The metabolism of xenobiotics by cytochrome P450 pathway has been considered as an important pathway associated with the progression of cancer (Hu and Chen, 2012). Besides, three other metabolism-related pathways, namely, lipid metabolism and autophagy, glutamine-regulatory enzymes, and Akt/c-Myc pathway (Dando et al., 2013; Blum and Kloog, 2014), have been identified to directly affect the growth of PC cells. The small molecule metabolic process also has been found as the enriched pathway for the biological process of PC (Hu et al., 2017). The identification of immune system-related regulation pathways has been reported to provide several new insights for PC treatment and prognosis (Yang and Michailidis, 2016).

Analysis of Characteristic Genes

In order to further demonstrate the validity of the NMFNA and deeply understand biological functions of core modules, we

detect and analyze characteristic genes from these core modules. First, ME and CNV core modules identified by four compared methods are filtered by removing weak edges with their PCC less than or equal to 0.8. Second, based on these filtered networks, the MS of each gene in the corresponding module is calculated, which can be considered as its contribution to interactions among genes in the module. Third, genes in each core module are sorted in descending order according to the MS, and the top ones are viewed as characteristic genes. In the study, the top 10, 30, and 50 characteristic genes in each core module are, respectively, selected and sent to GeneCards⁵ to measure their relevance scores, which represent association strengths between corresponding genes and PC. After that, for each compared method, relevance scores of characteristic genes in both ME and CNV core modules are summated together and recorded in **Table 4**. It is seen that in all scenarios, scores of the NMFNA are significantly larger than the scores of other compared methods, which implies that characteristic genes detected by the NMFNA are more relevant

⁵<http://www.genecards.org>

TABLE 4 | Summated relevance scores of characteristic genes in both ME and CNV core modules identified by each compared method.

Method	Top 10	Top 30	Top 50
NMFNA	32.80	90.74	220.54
NetNMF	22.51	58.62	97.70
TriNMF	12.03	52.87	76.29
NMF	12.83	81.05	117.23

TABLE 5 | PC-related genes in the top 10 characteristic genes of both ME and CNV core modules.

Method	Count	Gene
NMFNA	8	<i>TTC21A TNKS1BP1 TK1 KCNJ1 USF1 SDC3 MAN1C1 NR3C2</i>
NetNMF	6	<i>YTHDF1 TK1 KCNJ1 USF1 CALCOCO1 LIMD1</i>
TriNMF	5	<i>MTF1 LTBP4 SDC3 NEDD4 ASCC2</i>
NMF	5	<i>UBL4B FCGR1A KMT2D ANK1 PARG</i>

TABLE 6 | Numbers of common genes in ME and CNV core modules.

Module	NMFNA	NetNMF	TriNMF	NMF
ME	1,139	1,135	1,058	1,147
CNV	1,470	1,435	1,019	819

to PC and are more likely to reveal the pathogenic mechanisms of PC.

In addition, for each compared method, PC-related genes in the top 10 characteristic genes of both ME and CNV core modules are retrieved from GeneCards and listed in **Table 5**. It is seen that the NMFNA hits eight genes and is the winner of all compared methods. We then analyzed in detail the biological functions of these PC-related genes. The *TNKS1BP1* has been reported to regulate cancer cell invasion, which might further affect the progression of PC (Mayerle et al., 2003). The *TK1* level is upregulated 4-fold in the mice PC specimen (Yao et al., 2012); therefore, we naturally speculate that it might also play a potential role in the human PC. The variation of the *KCNJ1* has been claimed to be associated with diabetes (Farook et al., 2002), which is a closely related disease to PC and is generally thought of as the important risk factor of PC. As an independent prognostic biomarker, the *SDC1* has been confirmed to be upregulated in PC (Juuti et al., 2005). Since the *SDC1* is an important paralog of the *SDC3*, we infer that the *SDC3* might be related to PC. The *NR3C2* has been identified as the target of miR-135b-5p, which promotes migration and invasion of PC cells (Zhang et al., 2017). Though the *TTC21A*, *USF1*, and *MAN1C1* have been marked as PC-related genes in GeneCards, and they are indeed associated with several PC complicating diseases, including hyperlipidemia and alcohol-induced mental disorder, there are few supporting literature studies.

Besides, the experiments of the GEO datasets are also performed to further verify the effectiveness of modules and characteristic genes identified by the NMFNA method. Firstly, we calculated the numbers of common genes of the four datasets GSE62452, GSE15471, GSE16515, and GSE28735. As shown in

Supplementary Figure 1, there are 17,327 common genes, which are more likely to be related to PC. Secondly, **Table 6** lists the number of genes in the modules obtained by different methods that overlap with these common genes. It can be seen from **Table 6** that genes of the core modules identified by the NMFNA method contain the largest number of common genes, which indicates that these core modules have been verified to be related to PC in different databases.

CONCLUSION

Pancreatic cancer is a disease with a poor prognosis, in which malignant cells originate in the pancreatic tissue. To understand its pathogenic mechanisms, in this study, based on NMF and graph-regularized constraint, we presented NMFNA to identify modules and characteristic genes from integrated ME and CNV data of PC. First, the ME network, CNV network, and ME–CNV network are constructed by the PCC. Then, these networks are further integrated and decomposed simultaneously to identify modules effectively due to the introduced graph-regularized constraint, which is the highlight of the NMFNA. Finally, both GO and pathway enrichment analyses are performed, and characteristic genes are detected by the multimeasure score, to deeply understand the biological functions of PC core modules. Compared with the NMF, TriNMF, and NetNMF, the NMFNA identified more PC-related GO terms, pathways, and characteristic genes in core modules, demonstrating that the NMFNA facilitates the integration and decomposition of two types of PC data simultaneously and can further serve as an alternative method for detecting modules and characteristic genes from multiple genetic data of complex diseases.

The NMFNA has several advantages. First, it performs well in integrating and decomposing different types of genetic data simultaneously. Second, introducing the graph-regularized constraint into the NMFNA eases the heterogeneity of multiple networks, which is beneficial to detect core modules. Third, the NMFNA can not only consider individual genetic effects but also capture interaction effects among different features contributing to the development of PC. Nevertheless, it still has some limitations. For instance, associations between ME modules and CNV modules, i.e., S_{11} , S_{22} , are not deeply analyzed in theory and experiment; it only supports the integration and decomposition of two types of genetic data and fails three or higher types. These limitations inspire us to continue working in the future.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/**Supplementary Material**, further inquiries can be directed to the corresponding author/s.

AUTHOR CONTRIBUTIONS

QD and YS designed the NMFNA method. QD and JS implemented and performed the experiments. QD, FL, YZ, and

J-XL analyzed the experiment results and wrote the manuscript. All authors read and approved the final manuscript.

FUNDING

This work was supported by the National Natural Science Foundation of China (61972226, 61902216, 61902430, and 61872220) and the China Postdoctoral Science Foundation (2018M642635). The funder played no role in the design of the

study and collection, analysis, and interpretation of data and in the writing of the manuscript.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/fgene.2021.678642/full#supplementary-material>

Supplementary Figure 1 | Common genes of four GEO datasets.

REFERENCES

- Biswas, N. K., Das, S., Maitra, A., Sarin, R., and Majumder, P. P. (2014). Somatic mutations in arachidonic acid metabolism pathway genes enhance oral cancer post-treatment disease-free survival. *Nat. Commun.* 5:5835.
- Blum, R., and Kloog, Y. (2014). Metabolism addiction in pancreatic cancer. *Cell Death Dis.* 5:e1065. doi: 10.1038/cddis.2014.38
- Chen, J., and Zhang, S. (2018). Discovery of two-level modular organization from matched genomic data via joint matrix tri-factorization. *Nucleic Acids Res.* 46, 5967–5976. doi: 10.1093/nar/gky440
- Dando, I., Donadelli, M., Costanzo, C., Pozza, E. D., and Palmieri, M. (2013). Cannabinoids inhibit energetic metabolism and induce AMPK-dependent autophagy in pancreatic cancer cells. *Cell Death Dis.* 4:e664. doi: 10.1038/cddis.2013.151
- Deng, C., He, X., Han, J., and Huang, T. S. (2011). Graph regularized nonnegative matrix factorization for data representation. *IEEE Trans. Pattern Anal. Mach. Intell.* 33, 1548–1560. doi: 10.1109/tpami.2010.231
- Dennis, G., Sherman, B. T., Hosack, D. A., Yang, J., Gao, W., Lane, H. C., et al. (2003). DAVID: database for annotation. *Vis. Integrat. Discov. Genome Biol.* 4:3.
- Ding, C., Li, T., Peng, W., and Park, H. (2006). “Orthogonal nonnegative matrix t-factorizations for clustering,” in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (Philadelphia, PA).
- Ding, Q., Shang, J., Sun, Y., Wang, X., and Liu, J.-X. (2019). HC-HDSD: A method of hypergraph construction and high-density subgraph detection for inferring high-order epistatic interactions. *Computat. Biol. Chem.* 78, 440–447. doi: 10.1016/j.compbiolchem.2018.11.031
- Dong, Y. Y., Zhuang, Y. H., Cai, W. J., Liu, Y., and Zou, W. B. (2016). The mitochondrion interfering compound NPC-26 exerts potent anti-pancreatic cancer cell activity in vitro and in vivo. *Tumour Biol.* 37, 15053–15063. doi: 10.1007/s13277-016-5403-5
- Dreves, A., Facchinei, F., Kanzow, C., and Sagratella, S. (2011). On the solution of the KKT conditions of generalized Nash equilibrium problems. *SIAM J. Opt.* 21, 1082–1108. doi: 10.1137/100817000
- Elyanow, R., Dumitrascu, B., Engelhardt, B. E., and Raphael, B. J. (2020). NetNMF-SC: leveraging gene-gene interactions for imputation and dimensionality reduction in single-cell expression analysis. *Genome Res.* 30, 195–204. doi: 10.1101/gr.251603.119
- Farook, V. S., Hanson, R. L., Wolford, J. K., Bogardus, C., and Prochazka, M. (2002). Molecular analysis of KCNJ10 on 1q as a candidate gene for type 2 diabetes in pima Indians. *Diabetes* 51, 3342–3346. doi: 10.2337/diabetes.51.11.3342
- Gajate, C., Matos-Da-Silva, M., Dakir, L. H., Fonteriz, R. I., Alvarez, J., and Mollinedo, F. (2012). Antitumor alkyl-lysophospholipid analog edelfosine induces apoptosis in pancreatic cancer by targeting endoplasmic reticulum. *Oncogene* 31, 2627–2639. doi: 10.1038/ncr.2011.446
- Gao, Y., Hou, M., Liu, J., and Kong, X. (2019). An integrated graph regularized non-negative matrix factorization model for gene co-expression network analysis. *IEEE Access* 7, 126594–126602. doi: 10.1109/access.2019.2939405
- Gong, H., Wu, T. T., and Clarke, E. M. (2014). Pathway-gene identification for pancreatic cancer survival via doubly regularized Cox regression. *BMC Syst. Biol.* 8(Suppl. 1):S3.
- Hou, M.-X., Liu, J.-X., Gao, Y.-L., Shang, J., Wu, S.-S., and Yuan, S.-S. (2019). A new model of identifying differentially expressed genes via weighted network analysis based on dimensionality reduction method. *Curr. Bioinform.* 14, 762–770. doi: 10.2174/1574893614666181220094235
- Hu, B., Shi, C., Jiang, H. X., and Qin, S. Y. (2017). Identification of novel therapeutic target genes and pathway in pancreatic cancer by integrative analysis. *Medicine (Baltimore)* 96:e8261. doi: 10.1097/md.00000000000008261
- Hu, K., and Chen, F. (2012). Identification of significant pathways in gastric cancer based on protein-protein interaction networks and cluster analysis. *Genet. Mol. Biol.* 35, 701–708. doi: 10.1590/s1415-47572012005000045
- Juhász, M., Chen, J., Lendeckel, U., Kellner, U., and Ebert, M. P. A. (2003). Expression of carbonic anhydrase IX in human pancreatic cancer. *Aliment. Pharmacol. Therapeut.* 18, 837–846. doi: 10.1046/j.1365-2036.2003.01738.x
- Juuti, A., Nordling, S., Lundin, J., Louhimo, J., and Haglund, C. (2005). Syndecan-1 expression - a novel prognostic marker in pancreatic cancer. *Oncology* 68, 97–106.
- Kanehisa, M., Furumichi, M., Tanabe, M., Sato, Y., and Morishima, K. (2016). KEGG: new perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Res.* 45, D353–D361.
- Kwon, M. S., Kim, Y., Lee, S., Namkung, J., Yun, T., Yi, S. G., et al. (2015). Integrative analysis of multi-omics data for identifying multi-markers for diagnosing pancreatic cancer. *BMC Genomics* 16(Suppl. 9):S4.
- Lee, D. D., and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature* 401, 788–791. doi: 10.1038/44565
- Li, C., Zeng, X., Yu, H., Gu, Y., and Zhang, W. (2018). Identification of hub genes with diagnostic values in pancreatic cancer by bioinformatics analyses and supervised learning methods. *World J. Surgical Oncol.* 16, 1–12.
- Liu, J., Wang, C., Gao, J., and Han, J. (2013). “Multi-view clustering via joint nonnegative matrix factorization,” in *Proceedings of the 2013 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics* (Philadelphia PA), 252–260.
- Liu, Y., Gu, Q., Hou, J. P., Han, J., and Ma, J. (2014). A network-assisted co-clustering algorithm to discover cancer subtypes based on gene expression. *BMC Bioinform.* 15:37. doi: 10.1186/1471-2105-15-37
- Long, J., Liu, Z., Wu, X., Xu, Y., and Ge, C. (2016). Screening for genes and subnetworks associated with pancreatic cancer based on the gene expression profile. *Mol. Med. Rep.* 13, 3779–3786. doi: 10.3892/mmr.2016.5007
- Lu, J., Zhang, X., Shen, T., Chao, M., Wu, J., Kong, H., et al. (2016). Epigenetic profiling of H3K4Me3 reveals herbal medicine jinfukang-induced epigenetic alteration is involved in anti-lung cancer activity. *Evid Based Comp. Alternat. Med.* 2016, 1–13. doi: 10.1155/2016/7276161
- Mayerle, J., Friess, H., Büchler, M. W., Schnenburger, J., Weiss, F. U., Zimmer, K. P., et al. (2003). Up-regulation, nuclear import, and tumor growth stimulation of the adhesion protein p120ctn in pancreatic cancer. *Gastroenterology* 124, 949–960. doi: 10.1053/gast.2003.50142
- Mishra, N. K., and Guda, C. (2016). Genome-wide DNA methylation analysis reveals molecular subtypes of pancreatic cancer. *Oncotarget* 8, 28990–29012. doi: 10.18632/oncotarget.15993
- Mizrahi, J. D., Surana, R., Valle, J. W., and Shroff, R. T. (2020). Pancreatic cancer. *The Lancet* 395, 2008–2020.

- Qiao, H. (2015). New SVD based initialization strategy for non-negative matrix factorization. *Pattern Recog. Lett.* 63, 71–77. doi: 10.1016/j.patrec.2015.05.019
- Sakai, Y., Honda, M., Matsui, S., Komori, O., Murayama, T., Fujiwara, T., et al. (2019). Development of novel diagnostic system for pancreatic cancer, including early stages, measuring mRNA of whole blood cells. *Cancer Sci.* 110, 1364–1388. doi: 10.1111/cas.13971
- Shang, J., Ding, Q., Yuan, S., Liu, J.-X., Li, F., and Zhang, H. (2019). Network analyses of integrated differentially expressed genes in papillary thyroid carcinoma to identify characteristic genes. *Genes* 10:45. doi: 10.3390/genes10010045
- Tao, L. Y., Zhang, L. F., Xiu, D. R., Yuan, C. H., Ma, Z. L., and Jiang, B. (2016). Prognostic significance of K-ras mutations in pancreatic cancer: a meta-analysis. *World J. Surgical Oncol.* 14:146.
- Treviño, J. G., Pillai, S. R., Kunigal, S. S., and Chellappan, S. P. (2011). Nicotine regulates DNA-binding protein inhibitor (Id1) through a Src-dependent pathway promoting tumorigenic properties and chemoresistance in pancreatic adenocarcinoma. *Cancer Res.* 71, 1972–1972.
- Wang, J. Y., Wang, X., and Gao, X. (2013). Non-negative matrix factorization by maximizing correntropy for cancer clustering. *BMC Bioinform.* 14:107. doi: 10.1186/1471-2105-14-107
- Wang, P., Gao, L., Hu, Y., and Li, F. (2018). Feature related multi-view nonnegative matrix factorization for identifying conserved functional modules in multiple biological networks. *BMC Bioinform.* 19:394.
- Wolpin, B. M., Michaud, D. S., Giovannucci, E. L., Schernhammer, E. S., Stampfer, M. J., Manson, J. E., et al. (2007). Circulating insulin-like growth factor binding protein-1 and the risk of pancreatic cancer. *Cancer Res.* 67, 7923–7928.
- Wu, T. T., Gong, H., and Clarke, E. M. (2011). A transcriptome analysis by lasso penalized Cox regression for pancreatic cancer survival. *J. Bioinform. Computat. Biol.* 9(Suppl. 1), 63–73. doi: 10.1142/s0219720011005744
- Xiao, Q., Luo, J., Liang, C., Cai, J., and Ding, P. (2018). A graph regularized non-negative matrix factorization method for identifying microRNA-disease associations. *Bioinformatics* 34, 239–248. doi: 10.1093/bioinformatics/btx545
- Yang, D., Zhu, Z., Wang, W., Shen, P., Wei, Z., Wang, C., et al. (2013). Expression profiles analysis of pancreatic cancer. *Eur. Rev. Med. Pharmacol. Sci.* 17, 311–317.
- Yang, Z., and Michailidis, G. (2016). A non-negative matrix factorization method for detecting modules in heterogeneous omics multi-modal data. *Bioinformatics* 32, 1–8.
- Yao, X., Zeng, M., Wang, H., Sun, F., Rao, S., and Ji, Y. (2012). Evaluation of pancreatic cancer by multiple breath-hold dynamic contrast-enhanced magnetic resonance imaging at 3.0 T. *Eur. J. Radiol.* 81, 917–922.
- Zhang, S., Liu, C.-C., Li, W., Shen, H., Laird, P. W., and Zhou, X. J. (2012). Discovery of multi-dimensional modules by integrative analysis of cancer genomic data. *Nucleic Acids Res.* 40, 9379–9391. doi: 10.1093/nar/gks725
- Zhang, Z., Che, X., Yang, N., Bai, Z., Wu, Y., Zhao, L., et al. (2017). miR-135b-5p Promotes migration, invasion and EMT of pancreatic cancer cells by targeting NR3C2. *Biomed. Pharmacother.* 96, 1341–1348. doi: 10.1016/j.biopha.2017.11.074
- Zhao, L., Zhao, H., and Yan, H. (2018). Gene expression profiling of 1200 pancreatic ductal adenocarcinoma reveals novel subtypes. *BMC Cancer* 18:603.
- Zhao, S., Lv, N., Li, Y., Liu, T., Sun, Y., and Chu, X. (2020). Identification and characterization of methylation-mediated transcriptional dysregulation dictate methylation roles in preeclampsia. *Human Genomics* 14, 1–10. doi: 10.1155/2012/170172
- Zheng, M., Hu, Y., Gou, R., Wang, J., Nie, X., Li, X., et al. (2019). Integrated multi-omics analysis of genomics, epigenomics, and transcriptomics in ovarian carcinoma. *Aging (Albany NY)* 11, 4198–4215. doi: 10.18632/aging.102047
- Zhou, Y., Cui, J., and Du, H. (2019). Autoantibody-targeted TAAs in pancreatic cancer: a comprehensive analysis. *Pancreatol.* 19, 760–768. doi: 10.1016/j.pan.2019.06.009

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2021 Ding, Sun, Shang, Li, Zhang and Liu. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.



Graph Neural Networks and Their Current Applications in Bioinformatics

Xiao-Meng Zhang¹, Li Liang¹, Lin Liu^{1,2*} and Ming-Jing Tang^{2,3*}

¹ School of Information, Yunnan Normal University, Kunming, China, ² Key Laboratory of Educational Informatization for Nationalities Ministry of Education, Yunnan Normal University, Kunming, China, ³ School of Life Sciences, Yunnan Normal University, Kunming, China

OPEN ACCESS

Edited by:

Xiangxiang Zeng,
Hunan University, China

Reviewed by:

Ling-Yun Wu,
Academy of Mathematics
and Systems Science (CAS), China
Lei Wang,
Changsha University, China

*Correspondence:

Lin Liu
liulinrachel@163.com
Ming-Jing Tang
tmj@ynnu.edu.cn

Specialty section:

This article was submitted to
Computational Genomics,
a section of the journal
Frontiers in Genetics

Received: 02 April 2021

Accepted: 28 May 2021

Published: 29 July 2021

Citation:

Zhang X-M, Liang L, Liu L and
Tang M-J (2021) Graph Neural
Networks and Their Current
Applications in Bioinformatics.
Front. Genet. 12:690049.
doi: 10.3389/fgene.2021.690049

Graph neural networks (GNNs), as a branch of deep learning in non-Euclidean space, perform particularly well in various tasks that process graph structure data. With the rapid accumulation of biological network data, GNNs have also become an important tool in bioinformatics. In this research, a systematic survey of GNNs and their advances in bioinformatics is presented from multiple perspectives. We first introduce some commonly used GNN models and their basic principles. Then, three representative tasks are proposed based on the three levels of structural information that can be learned by GNNs: node classification, link prediction, and graph generation. Meanwhile, according to the specific applications for various omics data, we categorize and discuss the related studies in three aspects: disease prediction, drug discovery, and biomedical imaging. Based on the analysis, we provide an outlook on the shortcomings of current studies and point out their developing prospect. Although GNNs have achieved excellent results in many biological tasks at present, they still face challenges in terms of low-quality data processing, methodology, and interpretability and have a long road ahead. We believe that GNNs are potentially an excellent method that solves various biological problems in bioinformatics research.

Keywords: bioinformatics, graph neural networks, deep learning, omics data, network biology

INTRODUCTION

In recent years, deep learning has met with great success in machine learning tasks such as speech recognition and image classification. Nevertheless, most of the theories of deep learning are focused on explaining regular Euclidean data (**Figure 1A**). With the rapid accumulation of non-Euclidean data represented by graph structure data (**Figure 1B**), more and more researchers begin to pay attention to the processing of graph structure data that can represent complex relationships between objects. For example, graph embedding algorithms are used to perform the mapping of graph structure data to simpler representations (Scarselli et al., 2008). However, this method may lose the topological information of the graph structure in the pre-treating stage, thereby affecting the final prediction result. Gori et al. (2005) proposed the concept of graph neural networks (GNNs) and designed a model that can directly process graph structure data based on research results in the field of neural networks. Scarselli et al. (2008) elaborated on this model, which showed that GNNs could deliver significantly better results than traditional methods due to using the topological information of graphs in an iterative process. Subsequently, new models and application research on

GNNs have been proposed. With the increasing interest in graph structure data mining, the research direction and application fields of GNNs have been greatly expanded.

In general, GNNs are actually a connectionist model that captures the dependence of graphs through message passing between nodes, which take into account the scale, heterogeneity, and deep topological information of input data simultaneously. At present, GNNs show reliable performance in mining deep-level topological information, extracting the key features of data, and realizing the rapid processing of massive data, such as predicting the properties of chemical molecules (Duvenaud et al., 2015), extracting text relationship (Peng et al., 2017), reasoning the structure of graphics and images (Wang et al., 2018), link prediction and node clustering of social networks (Zhang and Chen, 2018), network completion of missing information (Bojchevski and Günnemann, 2017), drug interaction prediction (Zitnik et al., 2018), etc.

In the era of biomedicine “big data,” the aggregation and growth of large amounts of multiform data created enormous challenges to bioinformatics studies. In response to the characteristics and demands of these data, many algorithms in the field of machine learning, especially in deep learning, have been widely used in bioinformatics and propelling the development of bioinformatics. In many cases, biological data is constructed as a biological network in non-Euclidean domains, such as the molecular structure of proteins and RNAs, genetic disease association networks, and protein interaction networks. These biological networks have a great contribution to bioinformatics studies, especially for revealing the complex mechanisms of diseases. Network-based disease prediction methods had been proposed in 2011 (Barabasi et al., 2011), which was based on the assumption that “if a few disease components are identified, other disease-related components are likely to be found in their network-based vicinity.” Goh et al. (2007) pointed out that the number of interactions between proteins in the same disease pathway was 10 times higher than in random experiments. Navlakha and Kingsford (2010) proved that the network topology method was effective in predicting the association of diseases and even the interaction of biomolecules. Compared with other models in deep learning, the natural advantage of GNNs in capturing hidden information in biological networks brings new opportunities to design computational models in the biology field. Besides this, GNNs are not only suitable for non-Euclidean data but also able to extract potential graph structures from data without apparent graph structures like images and make inferences and judgments based on this structure. Therefore, GNNs have been widely adopted in the field of medical imaging.

Through extensive literature investigation, we find that the application of GNNs in bioinformatics has been rapidly developed in recent years, and the number of research papers in this field has shown a rapid growth. **Figure 2** shows the statistics of published GNN articles in bioinformatics from 2015 to 2020.

Although previous studies have surveyed deep learning applications in bioinformatics (Wood and Hirst, 2004; Min et al., 2016; Sun et al., 2019) recently published a review paper of GCN in bioinformatics, these studies are confined to a narrow field such as drug discovery in reference. To the best of our

knowledge, this is the first effort to review the application and development of GNNs for bioinformatics. The rest of this paper is carried out from the following aspects: (1) several standard GNN models are introduced for a better understanding on how GNNs extract potential information from biological data; (2) three levels of GNN applications (node level, edge level, and graph level) are illustrated in specific biological tasks. Meanwhile, existing applications of GNNs for bioinformatics are classified based on various biological problems and data forms, and the role of GNNs in these studies is discussed; (3) according to the discussions on existing studies, we summarize the limitations of this field, including imbalances of biological data, and the methodological and interpretability challenges of GNNs. Finally, future research directions on various applications are proposed.

MODEL PRINCIPLE AND DEVELOPMENT

There have been various GNN models for processing graph structure data. In this section, we present the original GNN and its variant models, including graph convolutional network (GCN), graph attention network (GAT), and graph autoencoders.

Graph Neural Network

Gori et al. (2005) proposed a novel neural network model capable of processing graph structure data—graph neural network in 2005. As a pioneering work of deep learning methods in non-Euclidean spaces, the goal of GNN is to learn how to generate an accurate state embedding vector h_i , that is, the state of the node is constantly updated with the information dissemination mechanism on the graph; each update depends on the state information of the neighboring nodes at the previous time.

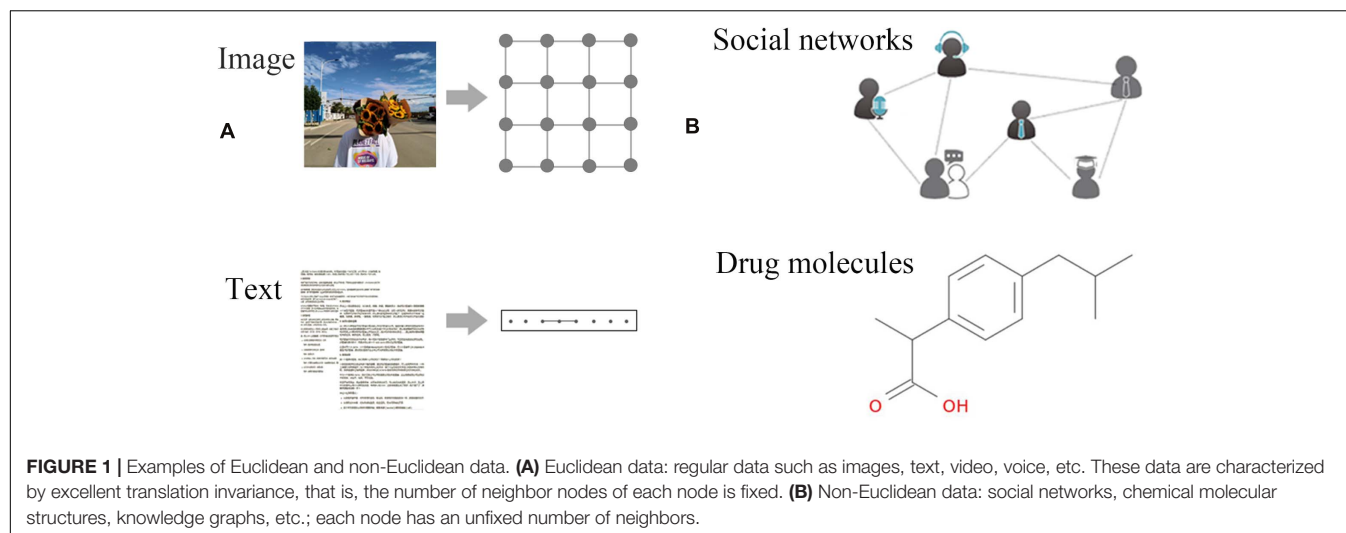
The related concepts are introduced as follows: let the input graph be $G = (V, E, X_V, X_E)$, $V = \{v_1, v_2, \dots, v_n\}$ represents the set of nodes, and $E = \{(i, j) \mid \text{when } v_i \text{ is adjacent to } v_j\}$ is the set of edges. x_i denotes the feature vector of node v_i , and $X_V = \{x_1, x_2, \dots, x_n\}$ is the set of feature vectors of all nodes. $x_{(i,j)}$ denotes the feature vector of edge (i, j) , and $X_E = \{x_{(i,j)} \mid (i, j) \in E\}$ is the set of feature vectors of all edges.

The input graph G is converted into a dynamic graph $G^t = (V, E, X_V, X_E, H^t)$ in the graph neural network model, where $t = 1, 2, \dots, T$ represents time and $H^t = (h_1^{(t)}, h_2^{(t)}, \dots, h_n^{(t)})$, $h_i^{(t)}$ represents the state vector of node v_i at time t , which depends on the graph G^{t-1} at time $t - 1$. The equation of $h_i^{(t)}$ is as follows:

$$h_i^{(t)} = f_w(x_i, x_{co(i)}, h_{ne(i)}^{t-1}, x_{ne(i)}) \quad (1)$$

where $f_w(\cdot)$ denotes the local transformation function with parameter w , $x_{ne(i)}$ is the set of feature vectors of all nodes adjacent to node v_i , $x_{co(i)}$ is the set of feature vectors of all edges connected to node v_i , and $h_{ne(i)}^{(t)}$ is the set of state vectors of all nodes adjacent to node v_i at time t . GNN updates the node status in an iterative manner, and this process is shown in **Figure 3**.

The long-term dependency problem of the original GNN (it is difficult for the node features to affect the state after multiple



updates) makes it laborious to learn the deep structure. Based on GNN, some variant models appear successively.

Graph Convolutional Networks

The existing GCN models can be divided into two categories: spectral-based and spatial-based GCN. Spectral-based GCN is defined from the perspective of graph signal processing, which exploits the principle of Laplacian and Fourier transform to map the irregular structure of a graph to a regular Euclidean space for convolution operation. Spatial-based GCN directly utilizes the information dissemination mechanism on the graph to define the

convolution operation, and its propagation method is similar to the original GNN. These two models will be discussed next.

Spectral-Based GCN

Spectral-based GCN uses the graph Laplace matrix as an important tool to extend the Fourier transform to the graph structure. Let A be the adjacency matrix with weighted undirected graph G , and the element $A(i, j)$ in the i -th row and j column of the matrix is the weight of the edge (i, j) . Degree matrix D is defined as follows:

$$D(i, j) = \sum_{j=1}^n A(i, j) \quad (2)$$

The symmetric normalized Laplace matrix of graph G is defined as follows:

$$L = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \quad (3)$$

As a real symmetry positive semidefinite matrix, L can be decomposed into:

$$L = U \Lambda U^T \quad (4)$$

where $U = (u_0, u_1, \dots, u_{n-1})$ is the eigenvector matrix, and

$\Lambda = \begin{bmatrix} \lambda_1 & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & \lambda_n \end{bmatrix}$ is the diagonal matrix of eigenvalues. The

normalized Laplacian matrix L and its eigenvector u form an orthogonal space as the Fourier transform ecosystem on the graph. The graph signal represents the feature vector of all nodes in the graph, expressed by $x = (x_0, x_1, \dots, x_{n-1}) \in \mathbb{R}^n$. The Fourier transform of the graph signal x is given below.

$$\hat{x} = U^T x \quad (5)$$

Calculate the convolution between the two signals as:

$$x * g = U((U^T x) \odot (U^T g)) \quad (6)$$

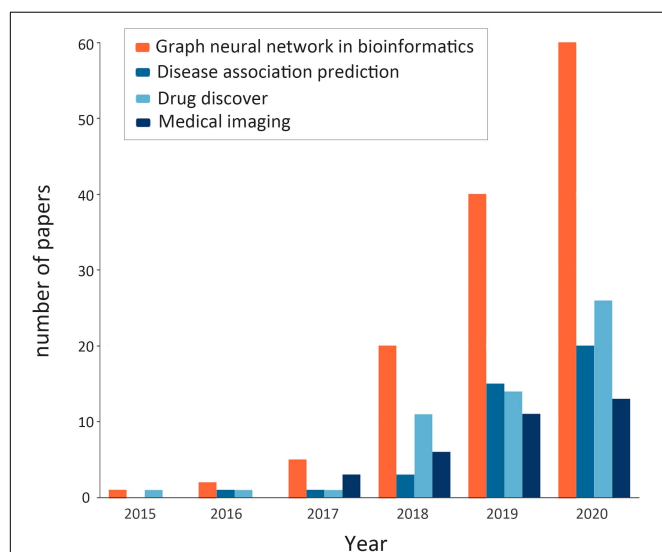


FIGURE 2 | Statistics of published graph neural network (GNN) articles in bioinformatics from 2015 to 2020. The orange bar shows the total number of GNN papers in bioinformatics that year (note: the number of papers in 2020 is counted until October). The remaining colors, in turn, represent the number of papers related to GNNs in bioinformatics in terms of disease association prediction, drug research, and medical image processing, which are the components of the orange bar.

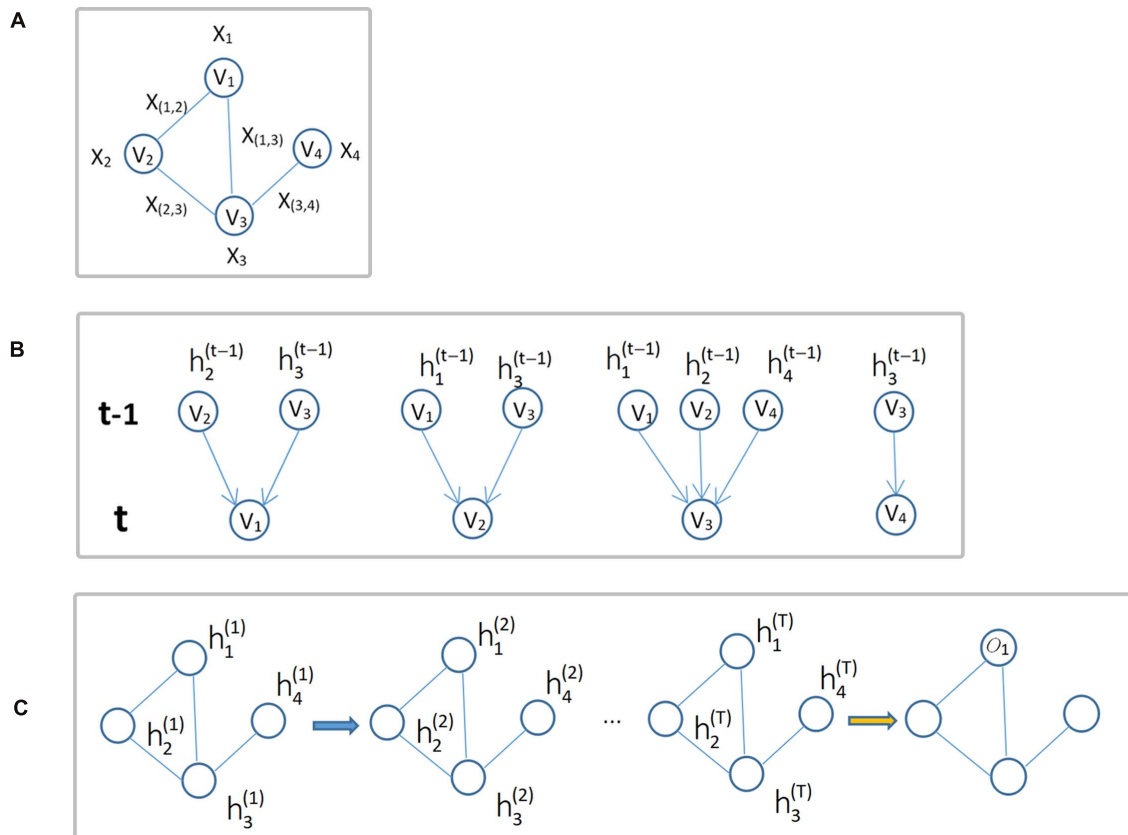


FIGURE 3 | Update of node status. **(A)** Input graph structure data G . **(B)** Diagram of each node iteration from time $t-1$ to time t . **(C)** The overall iteration process, where o_i is the output of the i -th node iteration.

If $\mathbf{g}_\theta = \text{diag}(\mathbf{U}^T \mathbf{g})$ is used as a filter for the graph signal \mathbf{x} , we can define the graph convolution as follows:

$$\mathbf{x} * \mathbf{g}_\theta = \mathbf{U} \mathbf{g}_\theta (\mathbf{A}) \mathbf{U}^T \mathbf{x} \quad (7)$$

This is the first generation of a spectral-based GCN model proposed by Bruna et al. (2013), which contains multiple convolutional layers. Spectral-based GCN maps the graph structure to Euclidean space through the Laplacian matrix of the graph. Nevertheless, due to matrix-vector multiplication, the computational complexity of the model is relatively high, which is $O(n^2)$. To solve this problem, Defferrard et al. (2016) proposed a model ChebNets that uses a K -degree polynomial filter in the convolutional layer. The κ -th polynomial filter of the spectrum in the model is expressed as shown below.

$$\mathbf{g}_\theta = \sum_{k=0}^K \theta_k \lambda_l^k \quad (8)$$

The K -th-order polynomial filter of the spectrum is expressed in the node domain as aggregating K -th-order neighborhoods to maintain spatial locality, and the number of filter parameters is also controlled to $O(K) = O(1)$. In order to further reduce the computational complexity, the model uses Chebyshev

polynomial $T_k(x) = 2T_{k-1}(x) - T_{k-2}(x)$ for recursive calculation, where $T_0(x) = 1$ and $T_1(x) = x$. Therefore, the convolution of the graph signal \mathbf{x} and the filter is defined as shown below.

$$\mathbf{x} * \mathbf{g}_\theta = \mathbf{U} \left(\sum_{k=0}^K \theta_k T_k(\tilde{\mathbf{L}}) \right) \mathbf{U}^T \mathbf{x} \quad (9)$$

As a simplification of the above-mentioned ChebNets, the graph convolutional network model proposed by Kipf and Welling truncates the Chebyshev polynomial to one time (Kipf and Welling, 2016a). For numerical stability, the adjacency matrix \mathbf{A} is adjusted to obtain $\tilde{\mathbf{A}}$, which results in a simplified combined convolutional layer.

$$\mathbf{H} = \mathbf{X} * \mathbf{g}_\theta = f(\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}} \mathbf{X} \Theta) \quad (10)$$

where $\tilde{\mathbf{A}} = \mathbf{I} + \mathbf{A}$, and $\tilde{\mathbf{D}}_{ij} = \sum_j \tilde{\mathbf{A}}_{ij}$, $f(\cdot)$ is the activation function; Θ is the filter parameter matrix.

Although the above-mentioned methods based on frequency domain perform well in feature extraction, their limitations are also obvious. First of all, due to the problem of data volume, the method based on the Laplacian matrix of graphs is hard to calculate on large graphs. Second, the trained GCN can only

be applied to a fixed graph structure rather than to arbitrary-structure graphs.

Spatial-Based GCN

The above-mentioned methods are based on the convolution theorem and define the graph convolution in the spectral domain, while the spatial method starts from the node domain and aggregates each central node and its neighboring nodes along the edge. Diffusion convolutional neural network (DCNN) (Atwood and Towsley, 2015) proposes that convolution is a process of diffusion between nodes and uses the k -hop transition probability obtained after random walking to define the weight between nodes. The structure of layer m is as follows:

$$\mathbf{H}^{(m+1)} = f(\mathbf{W}\mathbf{P}^k\mathbf{H}^m) \quad (11)$$

where \mathbf{P}^k denotes the k -hop reachability probability between two nodes in a random walk, and \mathbf{W} is a learnable model parameter. DCNN describes the high-order information between nodes, but it is hard to extend to a large graph because the computational complexity of the model is $O(n^2K)$.

GraphSage (Hamilton et al., 2017) randomly samples the neighboring nodes so that the neighboring nodes of each node are less than the given number of samples so as to adapt to the application on large-scale networks. The graph convolution operation is as follows:

$$\mathbf{h}_v^{(k)} = \sigma(\mathbf{W}^k \cdot f_k(\mathbf{h}_v^{(k-1)}, \{\mathbf{h}_u^{(k-1)}, \forall u \in S_{N(v)}\})) \quad (12)$$

where $f_k(\cdot)$ is the aggregate function, and $S_{N(v)}$ is the random sampling result of neighbors of node v . GraphSage gives a variety of forms of aggregation functions, which are mean aggregator, LSTM aggregator, and pooling aggregator.

There are also some studies aimed at defining the general framework of GCNs. Among them, mixture model networks (MoNet) (Monti et al., 2017) focus on the lack of translation invariance on the graph and map the local structure of each node to a vector of the same size by defining a mapping function. Finally, learn the shared convolution kernel on the result of the mapping. The message passing neural network (MPNN) (Gilmer et al., 2017) is based on information dissemination and aggregation between nodes and proposes a framework by defining a general form of the aggregation function.

Mixture model networks defines a coordinate system on the graph and expresses the relationship between nodes as a low-dimensional vector in the new coordinate system. At the same time, a weight function is defined on all adjacent nodes centered on a node, and a vector representation of the same size is obtained for each node.

$$D_j(x)f = \sum_{y \in N(x)} w_j(\mathbf{u}(x, y))f(y), \quad j = 1, \dots, J \quad (13)$$

where $N(x)$ represents the set of adjacent nodes of x , $f(y)$ represents the value of node y on the signal f , $\mathbf{u}(x, y)$ refers to the low-dimensional vector representation of the node relationship in the coordinate system \mathbf{u} , w_j represents the j -th weight function, and J represents the weight function number. This operation

makes each node get a J -dimensional representation, and the shared convolution kernel is defined on this.

Differently from MoNet, MPNNs point out that the core of graph convolution is to define the aggregation function between nodes using the aggregation function to get the local structure expression of each node and its neighboring nodes and then applying the update function to itself and the local structure expression to get the new expression of the current node. The convolution operation is as follows:

$$\mathbf{h}_v^{(k)} = U_k(\mathbf{h}_v^{(k-1)}, \sum_{u \in N(v)} M_k(\mathbf{h}_v^{(k-1)}, \mathbf{h}_u^{(k-1)}, \mathbf{x}_{vu}^e)) \quad (14)$$

where U_k and M_k are update function and aggregate function, respectively. The aggregate function learned under the spatial framework can be adapted to the task and the specific graph structure and has greater flexibility.

Graph Attention Networks

In order to solve the shortcomings of GCN and its similar structure, GAT (Veličković et al., 2017) introduces the attention mechanism into the propagation step of the graph to learn the weight between two connected nodes. In the GAT model, input the set $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ of the node feature to an attention layer; a new learned set $\mathbf{h} = \{\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_n\}$ of node feature will be output. The attention coefficient of edge (i, j) is represented by α_{ij} , and the equation is as follows:

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(a^T [\mathbf{W}\mathbf{x}_i || \mathbf{W}\mathbf{x}_j]))}{\sum_{k \in N_i} \exp(\text{LeakyReLU}(a^T [\mathbf{W}\mathbf{x}_i || \mathbf{W}\mathbf{x}_k]))} \quad (15)$$

where N_i is a set composed of adjacent nodes of node V_i , a represents the learnable weight vector, and \mathbf{W} is a shared linear transformation weight matrix. The output features of each node are calculated by the following equation:

$$\mathbf{h}_i = \sigma(\sum_{j \in N_i} \alpha_{ij} \mathbf{W}\mathbf{x}_j) \quad (16)$$

Multi-head attention expands the attention layer into K independent attention mechanisms to make the learning process of self-attention more stable, and the final expression is given as shown below.

$$\mathbf{h}_i = \sigma(\frac{1}{K} \sum_{k=1}^K \sum_{j \in N_i} \alpha_{ij}^k \mathbf{W}^k \mathbf{x}_j) \quad (17)$$

Parallel computing operations give GAT a higher efficiency, and the applicability of GAT on completely unknown graphs makes up for the limitation of spectral GCN.

Graph Autoencoder Networks

The wide application of autoencoder (AE) and its variants in the field of unsupervised learning has led to an increasing number of AE-based graph generation models. Sparse autoencoder

(SAE) (Tian et al., 2014) is the source of AE-based graph neural network. It uses the following \mathcal{L}_2 reconstruction loss:

$$\min_{\theta} \mathcal{L}_2 = \sum_{i=1}^N \|P(i, :) - \hat{P}(i, :)\|_2, \quad (18)$$

$$\hat{P}(i, :) = G(h_i), h_i = F(P(i, :)), \quad (19)$$

where P is the transition matrix, and \hat{P} is the reconstruction matrix; $h_i \in \mathbb{R}^d$ represents the low-dimensional representation of node v_i , and F and G are encoder and decoder, respectively. d is the dimension of hidden variables, and N is the number of nodes and $d \ll N$. On the basis of SAE, Wang et al. (2016) proposed a structural deep network embedding (SDNE) model, which modified the reconstruction loss function as shown below.

$$\min_{\theta} \mathcal{L}_2 = \sum_{i=1}^N \|(A(i, :) - G(h_i)) \odot b_i\|_2 \quad (20)$$

when $A(i, j) = 0$, $b_{ij} = 1$; otherwise, $b_{ij} = \beta > 1$, and β is a hyperparameter. The supervised learning method is used to learn the first-order approximation. The loss function is as follows:

$$\mathcal{L}_1 = \sum_{i,j=1}^N (A(i, j) |h_i - h_j|_2^2) \quad (21)$$

Finally, the loss function of SDNE is obtained:

$$\mathcal{L} = \mathcal{L}_2 + \alpha \mathcal{L}_1 + \mathcal{L}_{\text{reg}} \quad (22)$$

The variational autoencoder (VAE) (Kingma and Welling, 2013) is suitable for learning graph node representation without supervision information. Kipf and Welling (2016b) proposed a variational graph autoencoder (VGAE), which was the first time that VAE was extended to graphs. The generation model of VGAE is as follows:

$$p(A|H) = \prod_{i=1}^N \prod_{j=1}^N p(A(i, j) | h_i, h_j) \quad (23)$$

$$p(A(i, j) = 1 | h_i, h_j) = \text{sigmoid}(h_i^T, h_j) \quad (24)$$

Variational graph autoencoder learns parameters by minimizing the lower bound of variation L .

$$L = E_{q(H|F^V, A)}[\log p(A|H)] - KL[q(H|F^V, A) || p(H)] \quad (25)$$

Among them, $KL(\cdot)$ represents the Kullback–Leibler divergence function, which is used to measure the distance between two distributions. The time complexity of the model is $O(N^2)$.

APPLICATION PRINCIPLE OF GNNS IN BIOINFORMATICS

Modeling Methods

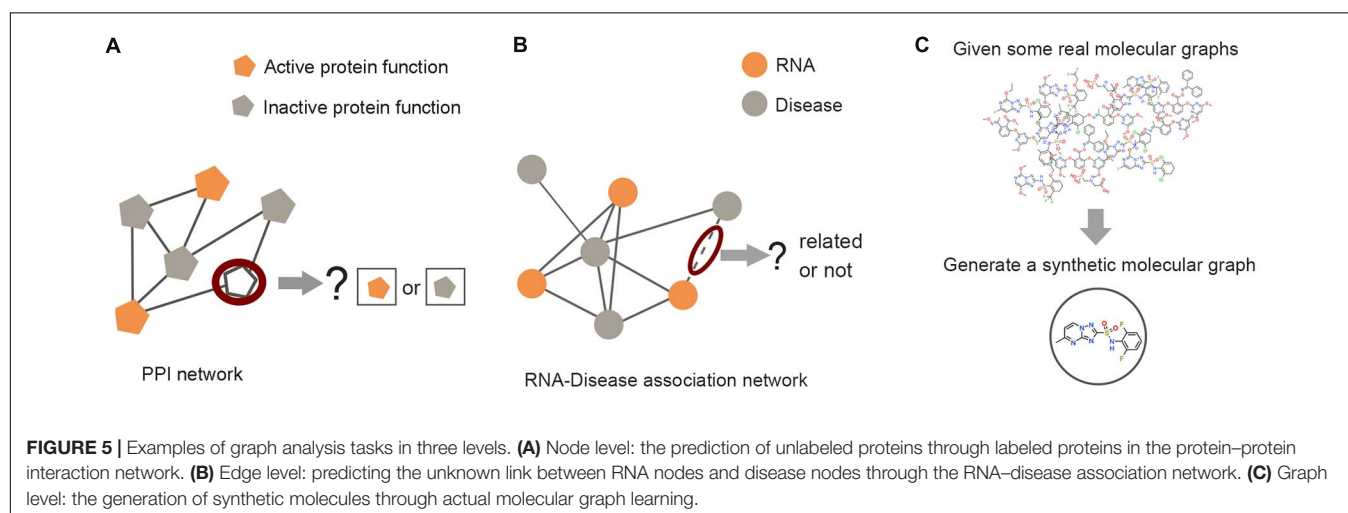
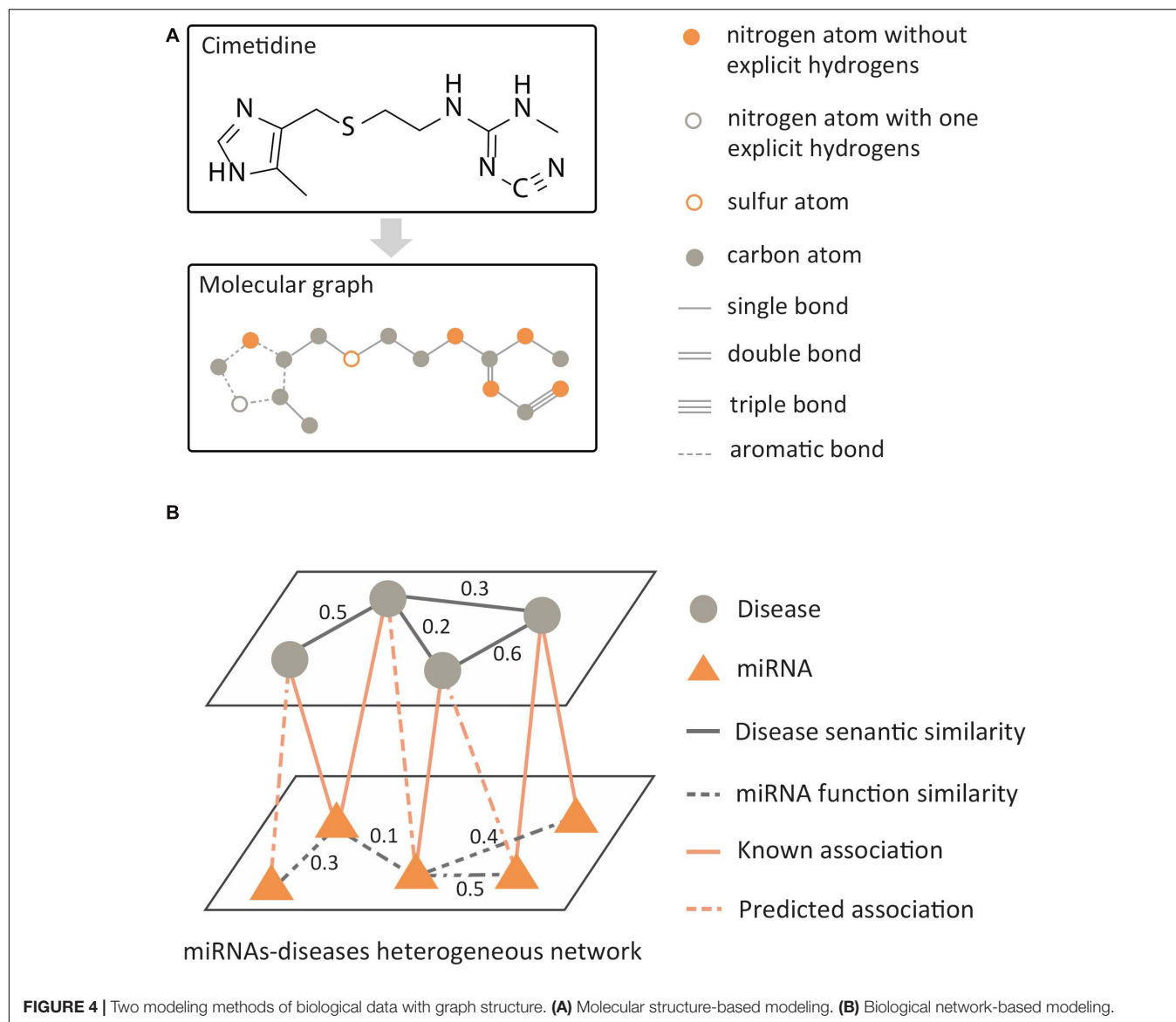
In the data analysis of bioinformatics, the biological data with graph structure can be modeled in two ways: molecular structure-based modeling and biological network-based modeling. For molecular structure-based modeling, atoms or valid chemical substructures (Jin et al., 2018) are used as nodes, bonds are used as edges, and then the molecular graph is constructed as shown in **Figure 4A**. Molecular graphs have a wide range of applications in predicting the properties of molecules and *de novo* molecular design. For biological network-based modeling, various entities are used as nodes, such as gene, disease, RNA, etc. The edges between nodes mean that there is a known association between pairs of entities, such as miRNA–disease interaction. Then, a relational network is generated, as shown in **Figure 4B**. GNNs are known to have a perfect performance in extracting potential information from graph structures, so they can process omics data in the biological field, including genomics, proteomics, RNomics, and radiomics. Combined with the above-mentioned two modeling methods, applying GNNs in these omics data can be employed for a variety of tasks, such as molecular property prediction, *de novo* molecular design, link prediction, node classification in biological networks, etc.

The Tasks of GNNs in Bioinformatics

Based on the modeling methods cited above, the structural information learned by GNNs provided a basis for different levels of graph analysis tasks: node level, edge level, and graph level (shown in **Figure 5**).

Node Level

Node classification is the typical task at the node level (**Figure 5A**), which can be performed by way of supervised learning, unsupervised learning, and semi-supervised learning. As the most commonly used method of node classification, semi-supervised learning combines the characteristics of supervised learning and unsupervised learning. Compared with supervised learning and unsupervised learning, semi-supervised learning on the graph extracts high-level node representations through information dissemination, which does not need to label all nodes and make good use of some known associated information. This setting is powerful for the task of inferring the association between entities in the biological network. For example, Ioannidis et al. (2019) constructed multiple protein–protein interaction (PPI) networks based on protein connectivity for different types of cells and proposed a graph residual neural network (GRNN) architecture for semi-supervised learning over multi-relational graphs. The influence of different relations was measured by learnable parameters. For the protein function prediction in generic cell, brain cell, and circulation cell data sets, GRNN had a macro F1 score of 0.86, 0.77, and 0.80, which was far better than the baseline model. In allusion to population disease prediction, Parisot et al. (2017) modeled population information as a graph, medical imaging data as the feature of the subject



node, and phenotype data as the weight of the edge. GCN was utilized to simultaneously model individual features and associations between subjects from potentially large populations. In the setting of semi-supervised learning, conditioning the GCN on the adjacency matrix provides the representation learning for all nodes. Compared with the standard linear classifier, their work improved the quality of prediction.

Edge Level

As the main task of the edge level, link prediction is defined as, given some graphs, an edge prediction model is trained based on the features of nodes or edges for predicting the connectivity probability between node pairs in these graphs or newly given graphs, as indicated in **Figure 5B**. The link prediction task has captured the attention of different research fields due to its broad applicability. Predicting the interaction between biological entities from complex biological networks also plays an important role in the research of bioinformatics and has become increasingly important and more challenging. The GNN models are also effective for solving the link prediction tasks. Zhang and Chen (2018) proposed the SEAL (learning from subgraphs, embeddings, and attributes for link prediction) model based on information dissemination, which used GNN to replace the fully connected neural network in the traditional Weisfeiler–Lehman neural machine method and learned general graph structure features from local subgraphs. Its performance on public biological network data sets such as yeast, *Caenorhabditis elegans*, and *Escherichia coli* was superior to the traditional graph embedding models. In addition, GCN has been utilized to predict various interactions in biological networks. For example, PPI networks with a small amount of label information that were encoded to predict the relationship between drugs and diseases (Bajaj et al., 2017), disease similarity networks, and microRNA (miRNA) similarity networks were built to indicate the association between miRNA and disease by VGAE (Ding et al., 2020).

Graph Level

The task of graph level is mainly related to graph generation (**Figure 5C**). Learning to generate graph structure data by training on a set of representative data is the core of graph generation tasks. For discovering new chemical structures, a graph generative model based on GNNs was first proposed with the motivation of molecular graphs generation. Simonovsky and Komodakis (2018) combined GNN and VAE to propose GraphVAE, which was used for small-scale molecular graph generation. Experiments on the QM9 database and ZINC database proved that GraphVAE has a higher accuracy than the previous methods. Jin et al. (2018) proposed a junction tree variational autoencoder (JT-VAE), which allowed the model to gradually expand the molecule while maintaining the chemical validity of each step. The experimental results on the ZINC database had shown that JT-VAE could generate better results than the traditional model and GraphVAE. MolGAN (De Cao and Kipf, 2018) is a generative model for small graphs, which is able to generate discrete graph structures and promote the generation of molecules with specific chemical properties through reinforcement learning methods (**Figure 6**).

The MolGAN model produced nearly 100% effective compounds in experiments on the QM9 chemical database. In practice, the abilities of graph generative models for protein structure prediction and chemical molecular map generation play an important role in related applications, such as drug design and protein structure design.

In general, as a novel type of graph embedding method, GNNs can perfectly integrate the features of nodes and the structural information between nodes in various specific applications. To better illustrate the application status and mechanisms of GNNs in bioinformatics, more comprehensive existing research on these three levels are summarized for specific biological problems, which are discussed in the next section.

TYPICAL APPLICATION OF GNNS IN BIOINFORMATICS

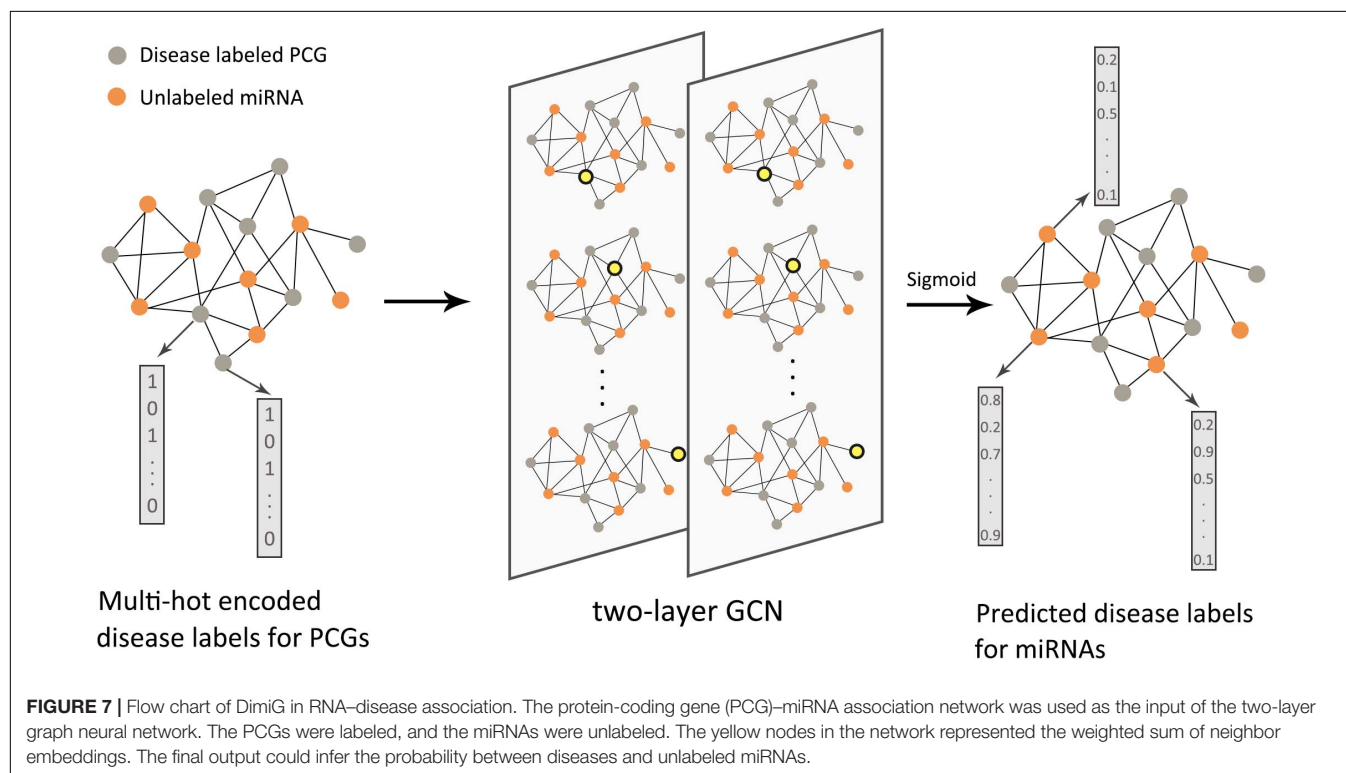
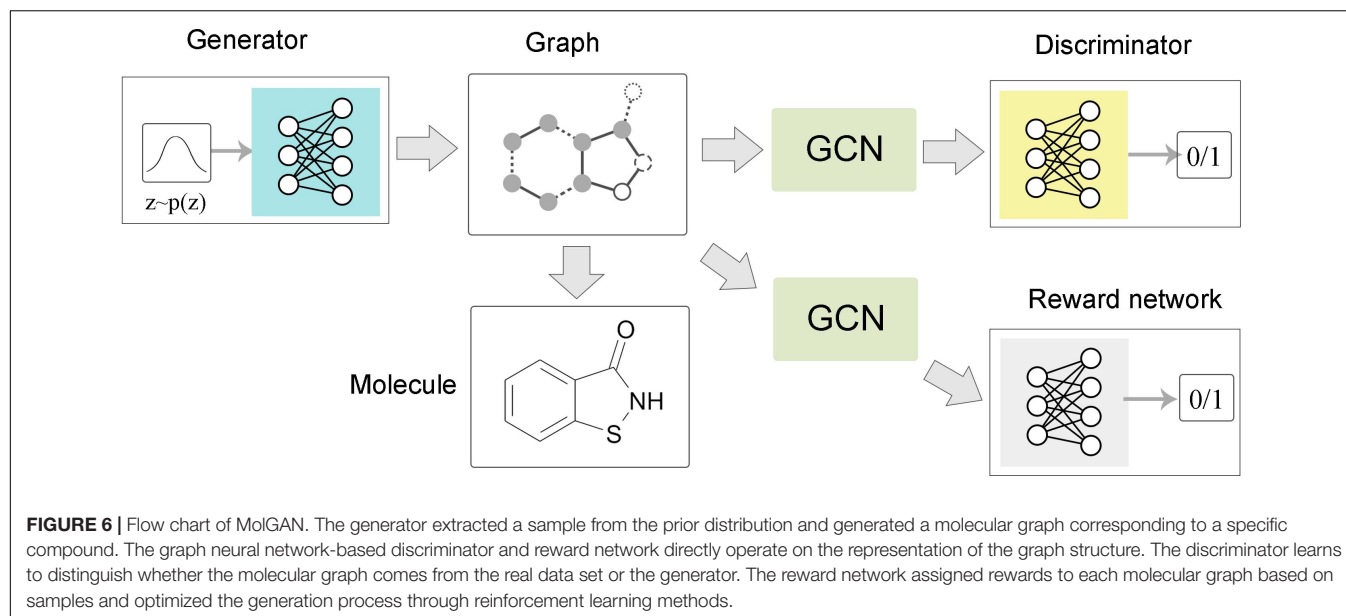
Based on various biological tasks, the existing application of GNNs in bioinformatics can be categorized into three typical topics: disease association prediction, drug development and discovery, and medical imaging. Note that these applications are also based on the three levels of graph analysis tasks in the previous section. In this section, the way and the development of GNNs handle representative problems are described in more detail.

Disease Association Prediction

Discovering the associated factors with various diseases is an important task in bioinformatics. At present, the existing methods of disease association prediction mainly include matrix decomposition (Koren et al., 2009; Wang et al., 2017) network propagation (Lee et al., 2011; Guan et al., 2012; Li and Li, 2012; Sun et al., 2014; Zhou et al., 2015), and machine learning (Luo et al., 2016; Zhou and Skolnick, 2016; Frasca, 2017; Xuan et al., 2019b; Jiang and Zhu, 2020). Essentially, some machine learning methods are also based on similarity measures and matrix decomposition. Nevertheless, matrix factorization methods map the features of entities to a latent space but ignore the representation of topological relationships between entities. In other methods, the shallow models ignore the rich structural information in disease-related networks, which ultimately affects the quality of entity feature representation. Recently, GNNs have been used to capture the nonlinear relationship between diseases and other entities in biological networks. More and more methods have introduced convolution operations into heterogeneous networks for extracting features of local subgraphs. All of the studies discussed in this section have directly or indirectly contributed to the development of deep learning methods in the field of disease prediction. Different biological networks were constructed in these studies, which were based on RNA–disease associations, disease–gene associations, and other association information.

RNA–Disease Association

A large amount of evidence has shown that microRNAs (miRNAs), long non-coding RNAs (lncRNAs), circular RNAs (circRNAs), and Piwi-interacting RNA are widely involved in the



occurrence and development of diseases (Xuan et al., 2019a; Li J. et al., 2020; Wang L. et al., 2020; Zheng et al., 2020). Therefore, the identification of these RNA–disease associations plays a crucial role in exploring the pathogenesis of complex diseases. The RNA and disease data analysis methods based on computational models make up for the high cost and time-consuming defects of biological experimental verification methods.

Starting in 2019, GNNs have been introduced into this type of research. Pan and Shen (2019) proposed a semi-supervised

multi-label graph convolution model (DimiG), which did not rely on known association information between miRNAs and diseases. DimiG integrated multiple networks related to protein-coding genes and used network knowledge transfer to indirectly predict the association between miRNAs and diseases; taking DimiG as an example, **Figure 7** shows how GCN uses the associated information in the network to generate features for unlabeled nodes. Li J. et al. (2020) used GCN to learn the feature representations of miRNAs and diseases from the miRNA

functional similarity network and disease semantic similarity network, respectively, and utilized the neural induction matrix to generate an association matrix, combining known miRNAs and disease association information to train the model. This model can predict all miRNAs related to breast cancer without any known related miRNAs. Based on the similarity method, the association prediction between RNAs and diseases can also integrate more useful information. Li C. et al. (2019) integrated miRNA–disease, miRNA–gene, disease–gene, and PPI networks. Furthermore, based on the information extracted by GCN, the top 10 unknown interactions between miRNAs and diseases were analyzed. Using the FastGCN algorithm and the Forest by Penalizing Attributes (Forest PA) classifier, Wang L. et al. (2020) can accurately predict potential circRNA disease associations. In order to better learn the hidden representation of node features, Zhang J. et al. (2019) used GCN combined with an attention mechanism to extract domain features and conducted experimental tests on two different RNA disease networks. There have also been some studies that used the autoencoder method on the graph to reconstruct node features. Wu et al. (2020) used GCN as an encoder to learn the feature representation of lncRNAs and diseases from the bipartite graph associated with lncRNA–disease, and the score of the lncRNA–disease interaction was calculated from the inner product of the two potential factor vectors. In the research of Ding et al. (2020), VGAE was used to reduce the noise effect caused by randomly selecting negative samples. In the prediction of disease-related RNAs with limited known data, the integration of multi-view information can help us understand complex biological networks more comprehensively. Therefore, to capture a deeper interaction mode between multiple related data, the integration method of different types of data by graph deep learning model needs to be further explored.

Disease–Gene Association

Single-cell RNA sequencing technology provides gene expression data for a single cell. GNNs can infer the interaction between cells (Jiahua et al., 2020; Zeng et al., 2020; Wang et al., 2021) and simulate cell differentiation (Bica et al., 2020) and disease state prediction (Ravindra et al., 2020). Precise gene–disease association prediction can help researchers reveal the function of disease-causing genes and provide evidence for disease prevention. Prioritizing candidate genes for various diseases is able to accelerate the development of early treatments and solve the DGP problem to a certain extent. Rao et al. (2018) proposed a rare disease gene sequencing method, which was different from the previous association network method. They integrated the combination pairwise ontological and curated associations into a heterogeneous network and used the frequency qualifier from Orphanet to calculate the edge weights. The qualifiers included terms such as “obligation,” “very frequent,” and “frequent.” Since the learning algorithm outlined in the standard VGAE does not focus on learning the relationships between different node types, Singh and Lio (2019) proposed a constrained VGAE variant for predicting specific node associations on the disease–gene association network by improving the optimization objectives of an algorithm. Wang et al. (2019a) defined a new cluster loss

function and a dropout mechanism based on the GCN and graph embedding method to improve the generalization ability.

Although a large amount of medical data has been reserved in various database, accurate prediction of cancer remains a challenge. As a group of complex diseases, cancer is caused by multiple gene defects, and there is synthetic lethality between genes. Therefore, the interaction network of genes plays an important role in cancer prediction (Iglehart and Silver, 2009). In order to analyze the underlying mechanism of cancer, Schulte-Sasse et al. (2019) initially used GCN to classify and predict cancer genes. Layer-wise relevance propagation was used to identify the gene input signals and the network topology of the learned model, which is in the neighborhood of a gene. The synthetic lethality between genes is extremely sparse. In order to solve the over-fitting problem, Cai et al. (2020) proposed a new GCN model based on fine-grained edge dropout and coarse-grained node dropout to reduce the over-fitting in sparse graphs. Chereda et al. (2021) combined the PPI network and gene expression data for patients and utilized GCN to classify the nodes in the patient's sub-network for predicting breast cancer metastasis. In the classification of breast cancer subtypes, there are also related studies based on local GCN, which was used to combine with the PPI network and the gene expression matrix information of multiple patients (Rhee et al., 2018). The correlation generated by the GNNs for each data point not only improves the interpretability of the model but also makes it more advantageous in predicting tasks related to patient-specific disease networks.

Others

In addition to the studies listed above, GNNs are also introduced into some research of other related fields, for instance, the discovery of disease proteins (Eyuboglu and Freeman, 2004). The disease protein prediction problem can naturally be defined as a semi-supervised classification problem on the protein–protein interaction network. The realization of the neighborhood positioning for visualized disease pathways proved that most diseases do not have obvious neighborhood positioning. Some studies utilized graph structures to model RNA secondary molecular structures for RNA classification (Rossi et al., 2019) and RNA-binding proteins prediction (Uhl et al., 2019; Yan et al., 2020), where bases were considered as nodes in the graph and phosphodiester bonds and hydrogen bonds were two different types of edges. In other studies, miRNA, lncRNA, and other elements were used to construct heterogeneous networks for predicting the interactions between miRNA and lncRNA as well as lncRNA-targeted genes. For multi-group biomedical data classification, a weighted patient similarity network was constructed based on various omics data and cosine similarity method (Wang T. et al., 2020), and GCN performs a feature extraction on these networks so as to find the cross-omics correlation in label space for integrating multi-omics effectively.

Drug Development and Discovery

The drug development process mainly includes drug target determination, lead compound discovery and optimization, candidate drug determination, preclinical research, and clinical

research (Vohora and Singh, 2017). However, the lack of drug targets, the poor clinical transformation of animal models, disease heterogeneity, and the inherent complexity of biological systems have made drug development a long and arduous process. The purpose of modern drug development is to speed up the intermediate steps through machine learning methods so as to save development costs. Therefore, more and more researchers tend to utilize machine learning models for predicting early molecular properties, which can tremendously reduce the workload of later experiments. As the most concerning machine learning method, deep learning in the field of biomedicine showed the following limitations: first of all, most deep learning models cannot learn structural information directly from the original input data, which rely on high-quality, labeled data sets, and secondly, traditional CNN or other deep models have difficulties in directly processing unstructured data like molecular graphs, so the internal structure information of molecules is usually not fully taken into account. Therefore, GNNs, which extend deep learning methods to non-Euclidean domains, have become the latest method to deal with drug-related tasks.

Protein Structure and Function Prediction

Protein function research occupies an important position in life sciences, and most diseases are closely related to protein dysfunction. Anfinsen (1973) found that the denatured ribonuclease that only retained the primary structure could refold and restore biological activity, which indicated that the amino acid sequence representing the primary structure of the protein contains important information about the secondary and tertiary structure of proteins. At present, significant progress has been made in protein structure prediction. The most accurate structure prediction can fully clarify the biological mechanism of protein action on a molecular scale, and its application in drug development and other fields is of great significance to biochemical research.

High computational cost and interpretability are problems in common methods of molecular structure analysis, such as 3D CNN and 2D CNN. In recent years, some studies have shown the powerful capabilities of GNNs in learning the effective structure of proteins from simplified graphical representations. Zamora-Resendiz and Crivelli (2019) proposed a protein structure learning method that was more suitable for large data sets. Unlike the previous 3D and 2D representations, this model could apply to the natural spatial representation of molecular structures, which brought a high transferability to the application direction. Aiming at the inverse protein folding problem, Ingraham et al. (2019) proposed a protein design framework based on a similar graph attention method, which could construct a conditional generation model for a given target structure protein sequence directly, and greatly improved the design efficiency. For protein function prediction, there are two types of methods: based on protein structure (Ioannidis et al., 2019) and based on PPI networks (Gligorijevic et al., 2019). Similar to a previous work, Gligorijevic et al. (2019) modeled the protein structure as a graph to predict the protein function. Ioannidis et al. (2019) used a multi-relation diagram method based on PPI network modeling with semi-supervised learning. The structural characteristics of

a protein determine the breadth and complexity of its function. However, the large number of invalid fragments contained in the protein sequence may affect the judgment of its function. Using GNNs to integrate the feature of protein relationship networks is one of the ways to solve the problem of differences in protein sequences and functions.

Protein-Protein Interaction Prediction

Protein-protein interaction information is able to provide theoretical assistance for drug development indirectly. Fout et al. (2017) proposed a space-based convolution operator to predict the interface between protein pairs, which was suitable for graphs with any size and structure. In the identification of protein complexes, the high rate of “false positive/negative” in the PPI network makes the detection of protein complexes arduous. Therefore, Yao et al. (2020) proposed a denoising method based on a variational graph autoencoder. They embedded the PPI network into the vector space through multi-layer GCN and deleted some interactions with credibility below the threshold so as to obtain a reliable association network. The experimental results on multiple datasets showed that the recognition accuracy of protein complexes increases by 5–200%. Liu X. et al. (2020) used an unsupervised GNN to predict the changes in protein binding properties after mutations and recognized abnormal interactions between atoms without annotations. By improving on the existing sorting algorithm, Cao and Shen (2020) and Johansson-Åkhe et al. (2020), respectively, proposed a scoring mechanism for the evaluation of protein docking models and doctored peptides. The convolution operation on graph encodes the structure and features of protein into the graph embedding representation and aggregates information along the edges of the network nodes for association scores, which solves the spatial limitations of conventional convolution methods.

Ligand-Protein (Drug-Target) Interaction Prediction

Drug targets are relevant to the pathological state of diseases or biomolecules, so the identification of drugs and their targets is the core problem in the development of new drugs. Drug target interaction prediction is essentially the interaction prediction problem between ligands and proteins, and many related studies have been performed. Nonetheless, there exist some problems as follows: (1) Traditional machine learning algorithms express the prediction results in a binary classification, but the real association relationship is not limited to the binary level. For some target proteins that do not exist in the test set but appear in practical applications, the prediction accuracy cannot be grasped; (2) It is more difficult to deal with the chemical space where drug molecules can be synthesized; (3) The prediction results lack biological interpretation. Although the test results of the model seem good, it is still unconvincing; and (4) The information about the ligand bound to a specific protein is always easy to obtain, but there is a lack of data on the real negative ligand-protein relationship for learning.

In response to these existing problems, Feng et al. (2018) introduced GCN into drug target identification for the first time, which learned the molecular structure information of drugs, and combined protein information as input. This study realized

the prediction of the real-valued interaction strength between drugs and targets and solved the cold-target problem. There are also some studies similar to the general thinking of the above-mentioned method but differ in data processing (Gao et al., 2018; Nguyen et al., 2021). Miyazaki et al. (2020) provided a drug–target interaction prediction model that ligands were specifically targeting toward proteins without using true negative interaction information. Torng and Altman (2019) established an unsupervised graph autoencoder to learn the representation of protein pockets without relying on the target–ligand complexes, where features were extracted from the pocket graph and 2D ligand graph by GCN, respectively. Jiang et al. (2020) proposed an association prediction method that constructs a molecular graph and a protein contact graph. These graphs were based on the structural information of drug molecules and the sequence information of proteins, which were performed by a three-layer GCN for providing an accurate prediction.

Unlike the above-cited methods of representing drugs as graphs, Zhao et al. (2021) proposed a network-based prediction model that incorporated the drug–protein association network into existing methods. The feature information of each drug–protein pair learned by GCN was used as the input of a deep neural network for predicting the final label. In order to make up for the ignorance of cellular context-dependent effects in previous studies, Zhong et al. (2020) used the information in gene transcriptional profiles to indirectly predict drug-targeted binding.

Besides this, knowledge graphs are also critical in biomedicine, which can be processed by GNNs. In these studies, the result of all graph-based interaction predictions was inseparable from the quality of the knowledge graph, but the content of the knowledge graphs extracted in real life is complex and contains interference information. Therefore, Neil et al. (2018) proposed a model that could adapt to noise data and reduced the influence of noisy data on the overall prediction effect of the model by assigning low weights to unreliable edges.

Prediction of Molecular Properties

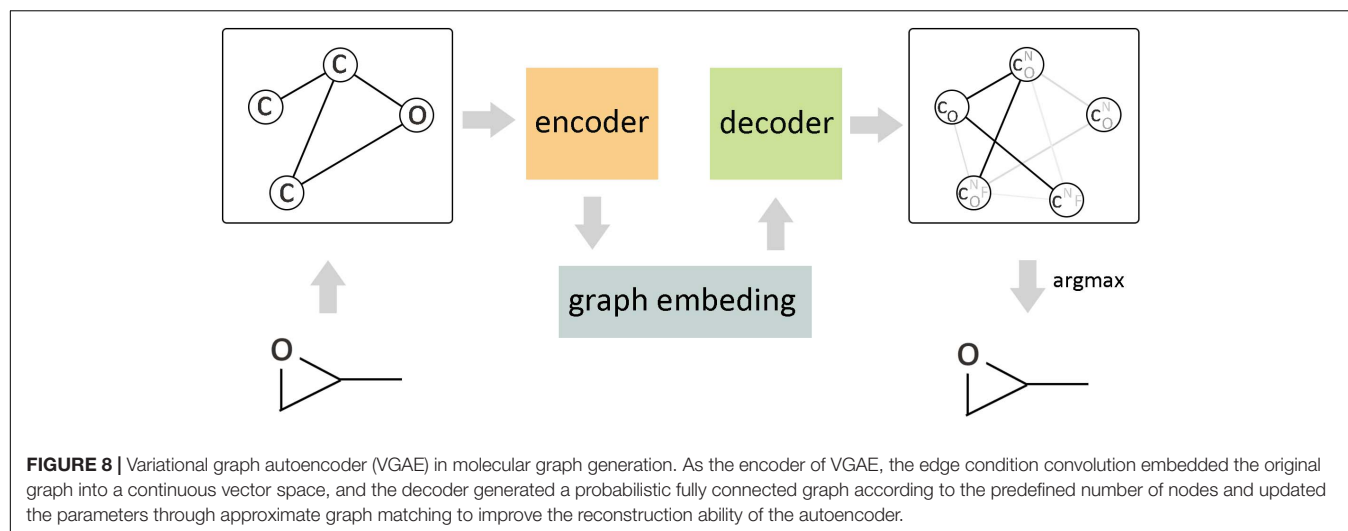
The prediction of molecular properties is a basic and important part of drug development. The first work to introduce graph convolution into the field of molecular properties learning (Duvenaud et al., 2015) was based on extended-connectivity circular fingerprints (ECFP), which created differentiable fingerprints to replace discrete operations in circular fingerprints and replaced hash functions with single-layer neural networks. The experimental results showed that, when the weights were large and random, this model exhibited a similar performance to ECFP, and as the weights were adjusted through training, its performance was better than ECFP. In view of the large space requirements of fingerprint-based methods and the large information noise of fingerprint encoding, Kearnes et al. (2016) proposed a molecular graph convolution method based on deep neural networks instead of molecular fingerprints. The molecular structure was represented by molecular graphs, and the distance between graphs forms the level of molecules. Although this method is not always superior in performance over molecular fingerprinting methods, it opens a new path for molecular

property prediction. In addition, the emergence of multi-task deep neural networks (MT-DNNs) makes neural networks more powerful in drug discovery. Liu et al. (2019) combined GCN with MT-DNNs to further improve the prediction accuracy and realized a completely data-driven deep learning method that did not rely on domain-specific feature descriptors or fingerprints for drug property prediction. It is well known that the electrostatic calculations are useful for the prediction of the chemical reactivity of molecules and their ability to form certain types of interactions. Rath et al. (2019) proposed a method to generate electrostatic potential surfaces close to quantum mechanics quality for ligand molecules within the time frame of interactive drug design, which provided an effective tool for medicinal chemists and modelers. In the process of drug discovery, the false positives or false negatives of bioassay conclusions caused by unstable compounds in storage make it hard to complete the stability prediction of compounds. Differently from the traditional rule-based method, Li et al. (2019b) proposed an end-to-end, attention-based GCN model to predict the stability of compounds. The model dynamically learned structural information from molecular graphs instead of pre-defined structural features, thereby reducing the risk of false alarms. The graph convolution operation can capture the local features of molecular sub-structure effects, thereby generating accurate global descriptors from the composite structure data.

De novo Molecule Design

The ultimate goal of drug design is to discover molecules with ideal chemical properties. Nonetheless, the hugeness and complexity of chemical space and the discontinuity of the spatial structure of compounds make it demanding to explore the chemical space of new molecules. By reducing the consumption of labor costs, computer-aided drug design is dedicated to accelerating the process of *de novo* molecular design. Although the generative model in machine learning can effectively generate molecules based on SMILES strings (Weininger, 1988), it cannot effectively represent the topological information of molecular structure. Based on the above-mentioned analysis, GNNs can be directly used to generate molecular graphs. Therefore, GNN is a kind of high-precision and low-cost method to determine molecular properties by analyzing the topological information of graphs.

The arbitrary connectivity and discrete structures of the graph make it laborious to generate a graph from the vectors in continuous code space, but if the maximum number of nodes in the generated graph is constrained, it is still computationally controllable. **Figure 8** shows a continuous embedding method of VAE to generate small molecular graphs, which was proposed by Simonovsky and Komodakis (2018). This early generation mode avoids the difficulties that may be encountered in generating graphs; the upper bound of negative log-likelihood was minimized in the model training, but it only applies to the generation of small molecules. Another way is based on the probability description of GCN to generate the graph gradually (Li et al., 2018a) rather than directly generating the entire graph. Compared with the routine method, this method has achieved better results, but there are still challenges in



the generation of macromolecular graphs. Li et al. (2018b) proposed a conditional graph generation model and explored two types of graph generation architectures. One treated graph generation as a Markov process, and the other introduced molecular-level recursive units to increase the scalability of the model. You et al. (2018) combined the prior knowledge of the example molecule dataset to generate goal-directed molecules. This method integrated and expanded three ideas of graph representation, reinforcement learning, and adversarial training, where reinforcement learning and confrontation training were combined to form a unified framework so as to reach the desired goal by continuously guiding the generation process and limiting the output space according to basic chemical rules. The experimental results show that this method achieves the most superior performance in terms of optimizing chemical properties and constrained properties under conditions similar to known molecules. Differently from the node-by-node method of generating graphs, Jin et al. (2018) proposed a method of connecting tree self-encoding, which utilized effective sub-graphs as components to generate molecular graphs in two stages. The first stage generated a connection tree structure as a sub-graph component, and then these sub-graphs were combined into a complete molecular graph in the second stage. This model avoided the generation of invalid intermediate states of molecules and improved the work efficiency. Khemchandani et al. (2020) learned the interactive binding model from the binding data through GCN and proposed an attribute prediction module, which utilized a scoring mechanism to determine the more useful molecules with the specific attribute during the generation process.

At present, the graph-based method of molecular generation has more advantages than the grammar-based generation method. Although the new compound obtained by the molecular graph method has higher scores on various evaluation indicators, it has also been drawn into question. Besides this, the method of generating molecular graphs is still limited to 2D space, and the 3D information of molecules is completely ignored, which may become the focus in the future.

Drug Response Prediction

The combination of genomics data and drug information for drug response prediction has promoted the development of personalized medicine. Huang et al. (2020) combined GCN with autoencoder to predict the association between miRNA and drug resistance. In this study, the association prediction was regarded as a problem of semi-supervised learning, and a graph convolution model was built by combining the known miRNA expression profile and the drug structure fingerprint information. There are some other studies that focus on the effect of drug treatment on cell growth. Liu Q. et al. (2020) predicted the therapeutic effect of drugs on cancer cells by constructing a cancer cell information sub-network and a drug structure sub-network. Considering the complexity of cancer factors, Singha et al. (2020) integrated biological network, genomics, inhibitor analysis, and disease–gene association data into a large heterogeneous graph. Multiple graph convolution blocks and attention propagation were used to aggregate network topology information, and a graphic readout framework was constructed for predicting the final result. Hwang et al. (2020) adopted a similar structure with the study of Liu et al. (Huang et al., 2020), but introduced a set of information about the dosage and duration of drug administration for predicting drug-induced liver injury.

Drug–Drug Interaction Prediction

When a drug is taken with another drug, the expected efficacy of drugs may be significantly changed. Therefore, research on drug–drug interaction (DDI) is essential to reduce the occurrence of adverse drug events and maximize the synergistic benefits in the treatment of diseases. Obviously, the most practical way to explore the medicinal properties of drug combinations is computer-aided DDI detection. Zitnik et al. (2018) predicted the side effects between drugs from a multimodal heterogeneous network that was composed of PPI, drug–protein targets, and drug–drug interactions, where each side effect was represented by a different edge. Ma et al. (2018)

proposed a framework of a multi-view drug graph encoder based on the attention mechanism, which was used to measure the drug similarity. To make full use of the heterogeneous correlation between different views, each type of drug feature had been considered as a view, which was associated with a learnable attention weight in the similarity integration. This multi-view method could capture more similar information than the previous single view.

Medical Imaging

Medical images play an extremely important role in clinical disease diagnosis, classification, and treatment. In the field of medical imaging, deep learning methods combined with a computer-aided diagnosis are used for the early detection and evaluation of diseases. Similar to other image-related tasks, segmentation, classification, and recognition are the main tasks of concern in medical imaging. Meanwhile, image data can be represented as a graph structure appropriate for the use of GNNs. Therefore, GNNs have an extensive application space in the field of medical imaging, such as image segmentation (Gopinath et al., 2019; Wang et al., 2019b; Tian et al., 2020a,b), abnormal detection (Wu et al., 2019) of MRI images and pathological images, classification (Shi et al., 2019; Zhou et al., 2019; Adnan et al., 2020) and visualization (Levy et al., 2020; Sureka et al., 2020) of histological images, analysis of surgical images (Zhang et al., 2018), image enhancement (Hu et al., 2020), registration (Hansen et al., 2019), retrieval (Zhai et al., 2019), brain connection (Ktena et al., 2017, 2018; Li X. et al., 2019a; Mirakhorli and Mirakhorli, 2019; Grigis et al., 2020; Li et al., 2020; Zhang and Pierre, 2020; Zhang et al., 2021) and disease prediction (Parisot et al., 2017; Kazi et al., 2018,

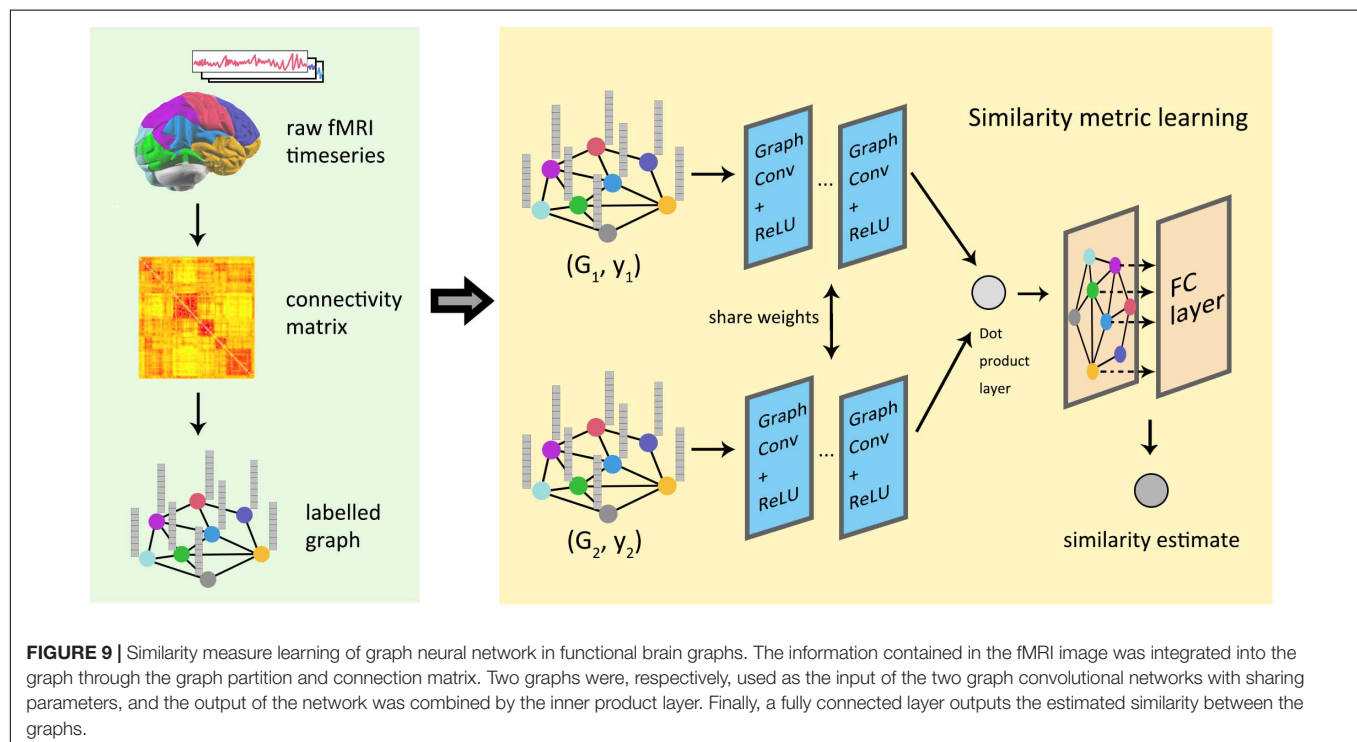
2019a,b,c; Anirudh and Thiagarajan, 2019; Yang et al., 2019; Stankevičiūtė et al., 2020; Zhang and Pierre, 2020; Zhang et al., 2021), etc.

Image Segmentation

Before GNNs were introduced, CNN-based image segmentation technology that was widely used in various studies has been maturing. Ma et al. (2018) proposed a gated graph neural network for segmenting 3D images and utilized directed graph learning to predict the movement of points on the basis of the coarse segmentation, where a second segmentation was performed to obtain a smooth image. For processing surface data (such as MRI), Tian et al. (2020a) utilized spectral convolution for realizing cortical surface parcellation. The traditional spectral embedding can only be realized in orthogonal grid space, but this method was able to directly learn the surface features of the cortex. Gopinath et al. (2019) proposed an automatic and interactive prostate contour prediction method based on GCN blocks, which could consider the multiscale feature, and utilized a contour matching loss training method to preserve the details of the prostate boundary. In most cases, the segmentation model of deep learning adopts the pixel-wise segmentation method with high computational complexity. In contrast, the GNN method only uses object contours to segment, reducing the amount of calculation.

Brain Connectivity Research

Global similarity measurement between graphics represents the structural or functional connections within the brain by labeled maps, which is of great significance in the study of brain



connectivity. At present, most models in deep learning use regular images as the default input data, but the processing of irregular brain connection images remains a problem. The pioneering work of depth graph networks in brain connection research was done by Ktena et al. (2018), and then this work was further extended about the way of cross-validation (Mirakhorli and Mirakhorli, 2019). **Figure 9** shows the process for the conversion and analysis of functional magnetic resonance imaging (fMRI) in the study of Ktena et al. which has been used in some later research of GNNs for brain connection. Zhai et al. (2019) constructed a generative model using graph convolution and VAE to predict the abnormal parts of input graphs. In order to lift the restrictions of fixed graphic structure for a model, Zhang and Pierre (2020) proposed a spatial GCN-based learning model, which could perform the classification of different brain connections and help predict the association between brain connection sub-networks and disease. Parisot et al. (2017) proposed an effective method to annotate human brain activity and then conducted a study on the transferability of brain decoding (Stankevičiūtė et al., 2020). Yang et al. (2019) predicted the state of consciousness of the brain by constructing a dynamic functional connectivity matrix that describes the state of the brain, which proved the effectiveness of graph convolution methods in predicting cortical signals.

Multimodal Fusion

All the studies discussed above are based on the analysis of a single image. Nevertheless, using only single-modal imaging data in disease prediction may lead to a lack of precision in the results. Yang et al. (2019) used the weighted edge-weighted graph attention network model to combine different modalities of medical imaging (brain structural magnetic resonance imaging and fMRI) for identifying bipolar disorder. In addition, more research tends to combine medical imaging data with non-imaging data in disease prediction. Based on multimodal fusion, the lack of some important information in monomodal data can be optimized and complemented to a certain extent. Parisot et al. (2017) introduced GCN into group-level medical applications for the first time. In this work, they proposed a population graph, which modeled people as nodes in the graph, and the feature vector of brain image was used as the feature representation of the node. Phenotypic data were combined for disease prediction in a semi-supervised way, such as gender, age, etc. In 2018, Parisot et al. extended this work with further in-depth analysis methods and modeling options (Stankevičiūtė et al., 2020). Since then, the use of population graph methods for disease prediction has become the choice of many researchers. Kazi et al. (2018) used a multi-level parallel GCN model to optimize the extraction of correlation information between nodes, which introduced an automatic learning layer for weight distribution and the attention mechanism for utilizing the features of each multimodal data (Kazi et al., 2019c). Aiming at the problem of insufficient feature extraction caused by fixed neighborhoods in GCN model, InceptionGCN (Kazi et al., 2019a) was proposed by Kazi et al., which considered the receptive field convolution kernels of different dimensions and utilized two aggregation methods to process all the features obtained by a convolution kernel.

Nevertheless, the performance of this model on various datasets is quite different, so the LSTM-based attention mechanism was introduced in later work to better integrate multi-modal data (Kazi et al., 2019b). For the research of autism spectrum disorder classification problem, Anirudh and Thiagarajan (2019) proposed a more robust method, which was based on the set of weakly trained G-CNN for reducing the model sensitivity to the choice of graph construction. Arya et al. (2020) believed that population phenotypic data is not suitable for defining the relationship between edges; then, the actual similarities between the brain's structures were used to directly extract the variables from brain MRI so as to establish the relationship between nodes. Stankevičiūtė et al. (2020) used a population graph to predict brain age but obtained unsatisfactory results. The topological characteristics between the various regions of the brain make GNNs more suitable for identifying related patterns of brain disease and effectively assisting in exploring the mechanism of the disease.

SUMMARY ON TYPICAL APPLICATIONS

In general, the applications of GNNs in node level, edge level, and graph level are not completely independent. According to the different levels of tasks and biological problems, we give a detailed summary on the above-mentioned typical applications in **Table 1**.

DISCUSSION AND FUTURE RESEARCH DIRECTIONS

The Problem and Trend of Methodology

Current GNNs have room for improvement in the methods of processing biological tasks. This section proposes the methodological problems and future development directions of GNNs for the three application fields of disease prediction, drug discovery, and biomedical imaging.

For disease prediction, improvements need to be made in three areas: the similarity evaluation of new nodes, the introduction of node attribute information, and heterogeneous information processing. First, most of the research in disease prediction adopted broad similarity methods. GNN models are used to extract the in-depth information of a heterogeneous network composed of disease semantic similarity, RNA functional similarity, and multiple association data. However, the construction of various similarity networks would have increased the complexity of the GNN model to a certain extent, and an efficient similarity evaluation paradigm needs to be improved for new diseases or RNA. In addition, more attention should be paid to the introduction of node attribute information into the modeling process, such as disease semantic features and RNA structural features, which can avoid the excessive dependence on associated information. Finally, for heterogeneous networks containing multi-source information, GNNs can deeply integrate their topological information. Nevertheless, current GNNs mainly focus on the processing of isomorphic graphs and cannot sufficiently capture the heterogeneity of nodes and edges in

TABLE 1 | Summary of typical applications.

Classification	Biological problems	Publications	Task of graph neural networks		
			Node level	Edge level	Graph level
Disease association prediction	RNA–disease association prediction	Wang L. et al., 2020	✓		
		Li C. et al., 2019; Zhang J. et al., 2019; Ding et al., 2020; Li J. et al., 2020; Wu et al., 2020; Zheng et al., 2020		✓	
	Inter-cell interaction prediction	Wang et al., 2021		✓	
	scRNA-Seq clustering	Zeng et al., 2020	✓		
	Impute the dropout events in scRNA-Seq data	Jiahua et al., 2020		✓	
	Simulate cell differentiation	Bica et al., 2020			✓
	Disease state prediction	Ravindra et al., 2020	✓		
	Disease genetic prioritization	Rao et al., 2018		✓	
		Li Y. et al., 2019		✓	
		Singh and Lio, 2019		✓	✓
		Wang et al., 2019a		✓	
	Disease–gene association prediction	Han et al., 2019			✓
		Iglehart and Silver, 2009	✓		
		Cai et al., 2020		✓	
		Chereda et al., 2021			✓
	Breast cancer metastasis prediction	Rhee et al., 2018		✓	
	Disease protein judgment	Eyuboglu and Freeman, 2004	✓		
	Discussion on the relationship between drug and disease	Bajaj et al., 2017		✓	
	RNA classification	Rossi et al., 2019			✓
	Predict the binding site of RNA protein	Uhl et al., 2019			✓
		Yan et al., 2020			✓
		Huang et al., 2019			✓
		Zhao et al., 2020			✓
	Biomedical data classification	Wang T. et al., 2020			✓
Drug development and discovery	Protein structure prediction	Zamora-Resendiz and Crivelli, 2019	✓		
	Protein design	Ingraham et al., 2019; Strokach et al., 2020			✓
		Gligorijevic et al., 2019			✓
	Protein function prediction	Fout et al., 2017		✓	
		Toomer, 2020	✓		
	Interface prediction between protein pairs	Yao et al., 2020			✓
	Protein–protein interaction network denoising	Liu X. et al., 2020		✓	
	Prediction of the influence of protein mutation on binding	Cao and Shen, 2020	✓		
	Protein docking model evaluation	Johansson-Åkhe et al., 2020			✓
	Assessment of docked peptide conformations	Feng et al., 2018			✓
	Drug–target interaction prediction	Gao et al., 2018; Miyazaki et al., 2020; Nguyen et al., 2021		✓	
		Neil et al., 2018; Tornø and Altman, 2019; Jiang et al., 2020; Zhong et al., 2020; Zhao et al., 2021			✓
		Duvenaud et al., 2015; Kearnes et al., 2016			✓
		Liu et al., 2019			✓
	Esp surface detection of ligands	Rathi et al., 2019	✓		
	Compound–protein interaction prediction	Li S. et al., 2020		✓	
		Tsubaki et al., 2018			✓
	Compound stability prediction	Li et al., 2019b			✓
	De novo molecule design	Dai et al., 2018; Li et al., 2018a,b; You et al., 2018; Khemchandani et al., 2020			✓
	Prediction of the association between miRNA and drug resistance	Huang et al., 2020		✓	
	Prediction of the effect of drugs on cancer cell growth	Liu Q. et al., 2020; Singha et al., 2020			✓

(Continued)

TABLE 1 | Continued

Classification	Biological problems	Publications	Task of graph neural networks		
			Node level	Edge level	Graph level
Medical image processing	Prediction of drug-induced liver damage	Hwang et al., 2020			✓
	Side effects prediction between drugs	Zitnik et al., 2018		✓	
	Drug similarity prediction	Ma et al., 2018		✓	
	Drug combination design	Long et al., 2020		✓	
	Drug recommendation	Mao et al., 2019		✓	
	Alkaloid classification	Eguchi et al., 2019			✓
	Product prediction of organic reactions	Coley et al., 2019			✓
	Chemical network prediction	Li Q. et al., 2018		✓	
	Image segmentation	Gopinath et al., 2019; Wang et al., 2019b; Tian et al., 2020a,b	✓		
	Image classification	Shi et al., 2019; Zhou et al., 2019; Adnan et al., 2020	✓		
	abnormal detection	Wu et al., 2019	✓		
	Image visualization	Levy et al., 2020; Sureka et al., 2020			✓
	Image enhancement	Hu et al., 2020	✓		
	Image registration	Hansen et al., 2019	✓		
	Image retrieval	Zhai et al., 2019	✓		
	Surgical image analysis	Zhang et al., 2018	✓		
	Disease prediction	Parisot et al., 2017; Kazi et al., 2018, 2019a,b,c; Anirudh and Thiagarajan, 2019; Yang et al., 2019; Arya et al., 2020; Stankevičiūtė et al., 2020		✓	
	Brain connection research	Ktena et al., 2017, 2018; Li X. et al., 2019a; Mirakhorli and Mirakhorli, 2019; Grigis et al., 2020; Zhang and Pierre, 2020; Zhang et al., 2021		✓	
		Li et al., 2020		✓	✓

heterogeneous networks (Zhang C. et al., 2019). So, a new architecture needs to be studied, which can consider the feature of data in heterogeneous biological networks.

In drug discovery, the construction mode of chemical networks and the definition of molecular model structure need to be further explored. In the study of compound interactions, compounds and chemical networks are usually modeled as graphs. These graph-based methods have been successfully applied to the related tasks of chemical networks, but there are few studies that can simultaneously consider these two different types of graphs in an end-to-end manner. Harada et al. (2020) used molecular graphs as nodes in chemical networks and performed internal and outer convolution operations on them. The dual graph convolutional network can capture the feature of the individual molecular graph structure and the molecular relationship network simultaneously, making excellent results in dense networks. In addition, current molecular modeling is based heavily on the 2D graph structure, and the 3D structure that may affect the properties of molecules has rarely been considered. Therefore, the research of GNNs on the molecular 3D structure may be a future direction that has been neglected previously.

Multimodal Fusion

Graph neural networks also have limitations in multimodal data processing in medical imaging. The natural combination of multimodal deep learning (Ngiam et al., 2011) and multi-source omics data accelerates the development of bioinformatics. Some studies of GNNs in the multimodal fusion have been discussed in Section 4.3.3. It is not uncommon to find that the data is relatively

balanced in these studies, but the involvement of unbalanced data cannot be avoided in actual tasks. Therefore, the method of processing unbalanced data in GNNs needs further research.

The Problem and Trend Caused by Biological Data

Existing research on biomolecular networks has proved that many biomolecular networks have the properties of sparseness and scale-free nature. Sparseness is expressed as if the network size is N , then the number of edges is $O(N)$ instead of N^2 , which results from a particular optimization in the long-term evolution of organisms. The scale-free nature is reflected in the degree distribution of these networks that obeys the power-law distribution, where most nodes have a small number of connections and a few nodes have a large number of connections. This characteristic demonstrates that a few nodes represented as biomolecules play a key role in the dynamic changes of biomolecular networks. For the problems of sparseness and scale-free nature of biomolecular networks, two methods—dropout and regularization—can be adopted to alleviate the overfitting caused by them. With a fixed probability, the dropout method randomly sets each dimension of the weight to zero during the training process so that the model only updates part of the parameters each time. As an example, the GCN model proposed by Cai et al. (2020) is based on the methods of fine-grained edge dropout and coarse-grained node dropout to making GCN learn a more stable representation in the process of continuous adaptation. Dropout can alleviate the instability when there is

a fantastic amount of data in the training set; consequently, the dropout method is better suited to large data sets. The regularization method adds a regularization term to the loss function to limit the scale of parameters. In the study of disease–gene association prediction by Han et al. (2019), there were 3,209 diseases and more than 10,000 genes in the data set. Nevertheless, only 3,954 known disease–gene associations and a margin control loss function were defined to reduce sparse impact.

In addition, the collection of negative sample data is often ignored in the most current research, which leads to the fact that the biological data only contains positive samples. The lack of negative samples increases the difficulty of model training. Therefore, in the study of Eyuboglu and Freeman (2004), a simple method was proposed to solve the lack of negative labels, which randomly select k from the set of unlabeled nodes to mark them as negative. This seemingly unreasonable method causes most tags to be randomly selected, but in fact, randomly selected proteins may have a low correlation with disease. Hence, a certain number of random negative labels can be considered reasonable.

Finally, there is a large amount of noise information in the biomolecular network, so noise reduction processing is a very positive step for improving the model's performance. For example, GAT can assign low weight to noisy data or directly eliminate the associations with correlation degrees lower than the threshold in the network. More methods for reducing noise are worth further development.

The Lack of Interpretability

In bioinformatics, simply providing the computing results is just not far enough. The lack of interpretation is a persistent problem of the black box model like deep learning. As the entities and relationships in GNNs usually correspond to various types of objects that exist in the real world, then GNNs have abilities to support more interpretable analysis and visualization (Selsam et al., 2018). Take learning molecular fingerprint (Duvenaud et al., 2015) as an example; the fingerprint encoding method using neural graphs can take into account the similarity between molecular fragments to achieve a more meaningful feature representation, which is also ignored in traditional fingerprint encoding. In the prediction of metastasis for breast cancer patients (Chereda et al., 2021), the graph layer-wise relevance propagation was proposed to explain how GCN generates predictions based on patient-specific PPI sub-network data which could be potentially highly useful for the development of personalized medicine. In histological image analysis, Sureka et al. (2020) modeled histological tissue as a nuclear graph and established a graph convolutional network framework based on attention mechanism and node occlusion for disease diagnosis. This method visualized the relative contribution of each cell nucleus by a whole-slide image. In data analysis, GNNs generate relevant information for each data node, which makes the model more interpretative to some extent. Overall, the further exploration of interpretability in modeling biological networks by GNNs is still an essential direction of future research.

The Exploration of Deep Structure

The deep network structure is more common in deep learning. For instance, a residual network (ResNet) that excels in image

classification has 152 layers (He et al., 2016), but the layers of most networks are below three in the field of GNNs (Zhou et al., 2020). Experiments have shown that, as the number of network layers increases, the characteristics of all nodes will approach the same value, which will reduce network performance (Li Q. et al., 2018). However, deeper networks can provide larger parameter space and stronger representation capabilities, so the feasibility of a deep graph neural network deserves to be explored.

CONCLUSION

Graph neural networks, as a branch of deep learning in non-Euclidean space, perform particularly well in various tasks that process graph structure data. In this paper, a systematic survey of GNNs and their advances in bioinformatics is presented from multiple perspectives. Three representative tasks are especially proposed based on the three levels of structural information that can be learned by GNNs: node classification, link prediction, and graph generation. Meanwhile, according to the specific applications for various omics data, we categorize and discuss the related studies in three aspects: disease prediction, drug discovery, and biomedical imaging. Finally, the limitations and future possibilities of applying GNNs to bioinformatics studies are illustrated.

Although GNN has achieved excellent results in many biological tasks at present, it still faces challenges in terms of low-quality data processing, methodology, and interpretability and has a long road ahead. We believe that GNNs are potentially a wonderful method that solves various biological problems in bioinformatics research. Furthermore, this paper can provide a valuable reference for new researchers joining the studies in this area.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author/s.

AUTHOR CONTRIBUTIONS

LLia and X-MZ contributed to conceptualization, methodology, and writing – original draft preparation. LLiu and M-JT contributed to formal analysis and writing – review and editing. X-MZ contributed to the investigation and data curation. All authors have read and agreed to the published version of the manuscript.

FUNDING

This research was funded by the National Natural Science Foundation of China (No. 61862067) and the Doctor Science Foundation of Yunnan Normal University (No. 01000205020503090).

REFERENCES

- Adnan, M., Kalra, S., and Tizhoosh, H. R. (2020). "Representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, Seattle. 988–989.
- Anfinsen, C. (1973). Principles that govern the folding of protein chains. *Science* 181, 223–230.
- Anirudh, R., and Thiagarajan, J. (2019). "Bootstrapping graph convolutional neural networks for autism spectrum disorder classification," in *Proceedings of the ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (Brighton: IEEE).
- Arya, D., Olij, R., Gupta, D. K., El Gazzar, A., Wingen, G., Worring, M., et al. (2020). "Fusing structural and functional MRIs using graph convolutional networks for autism classification," in *Medical Imaging With Deep Learning*, (Piscataway, NJ: PMLR), 44–61.
- Atwood, J., and Towsley, D. (2015). Diffusion-convolutional neural networks. *NIPS* 2016, 1993–2001.
- Bajaj, P., Heereguppe, S., and Sumanth, C. (2017). Graph convolutional networks to explore drug and disease relationships in biological networks. *Comput. Sci.*
- Barabasi, A., Gulbahce, N., and Loscalzo, J. (2011). Network Medicine: a network-based approach to human disease. *Nat. Rev. Genet.* 12, 56–68. doi: 10.1038/nrg2918
- Bica, I., Andres Terre, H., Cvejic, A., and Lio, P. (2020). Unsupervised generative and graph representation learning for modelling cell differentiation. *Sci. Rep.* 10:9790. doi: 10.1038/s41598-020-66166-8
- Bojchevski, A., and Günnemann, S. (2017). Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. *arXiv [Preprint]*. arXiv:1707.03815.
- Bruna, J., Zaremba, W., Szlam, A., and LeCun, Y. (2013). Spectral networks and locally connected networks on graphs. *arXiv [Preprint]*. arXiv:1312.6203.
- Cai, R., Chen, X., Fang, Y., Wu, M., and Hao, Y. (2020). Dual-dropout graph convolutional network for predicting synthetic lethality in human cancers. *Bioinformatics* 2020:btz211. doi: 10.1093/bioinformatics/btz211
- Cao, Y., and Shen, Y. (2020). Energy-based graph convolutional networks for scoring protein docking models. *Prot. Struct. Funct. Bioinform.* 88, 1091–1099.
- Chereda, H., Bleckmann, A., Menck, K., Perera-Bel, J., Stegmaier, P., Auer, F., et al. (2021). Explaining decisions of graph convolutional neural networks: patient-specific molecular subnetworks responsible for metastasis prediction in breast cancer. *Genome Med.* 13:845. doi: 10.1186/s13073-021-00845-7
- Coley, C., Jin, W., Rogers, L., Jamison, T., Green, W., Jaakkola, T., et al. (2019). A graph-convolutional neural network model for the prediction of chemical reactivity. *Chem. Sci.* 10:4228D. doi: 10.1039/C8SC04228D
- Dai, H., Tian, Y., Dai, B., Skiena, S., and Song, L. (2018). Syntax-directed variational autoencoder for structured data. *arXiv [Preprint]*. arXiv:1802.08786.
- De Cao, N., and Kipf, T. (2018). MolGAN: An implicit generative model for small molecular graphs. *arXiv [Preprint]*. arXiv:1805.11973.
- Defferrard, M., Bresson, X., and Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. *Adv. Neur. Inform. Proces. Syst. Barcelona NIPS* 2016, 3844–3852.
- Ding, Y., Tian, L. P., Lei, X., Liao, B., and Wu, F. X. (2020). Variational graph auto-encoders for miRNA-disease association prediction. *Methods* 2:4. doi: 10.1016/j.jmeth.2020.08.004
- Duvenaud, D., Maclaurin, D., Aguilera-Iparraguirre, J., Gómez-Bombarelli, R., Hirzel, T., Aspuru-Guzik, A., et al. (2015). "Convolutional Networks on graphs for learning molecular fingerprints," in *Advances in Neural Information Processing Systems (NIPS)*, Montreal, QC, 13.
- Eguchi, R., Ono, N., Morita, A., Katsuragi, T., Nakamura, S., Huang, M., et al. (2019). Classification of alkaloids according to the starting substances of their biosynthetic pathways using graph convolutional neural networks. *BMC Bioinform.* 20:2963. doi: 10.1186/s12859-019-2963-6
- Eyuboglu, E. S., and Freeman, P. B. (2004). Disease protein prediction with graph convolutional networks. *Genetics* 5, 101–113.
- Feng, Q., Dueva, E., Cherkasov, A., and Ester, M. (2018). Padme: A deep learning-based framework for drug-target interaction prediction. *arXiv [Preprint]*. arXiv:1807.09741. doi: 10.1007/s12559-021-09840-x
- Fout, A. M., Jonathon, B., Basir, S., and Asa, B. (2017). Protein Interface prediction using graph convolutional networks. *NIPS* 6533–6542.
- Frasca, M. (2017). Gene2DisCo: gene to disease using disease commonalities. *Artif. Intel. Med.* 82:1. doi: 10.1016/j.artmed.2017.08.001
- Gao, K. Y., Fokoue, A., Luo, H., Iyengar, A., Dey, S., and Zhang, P. (2018). Interpretable drug target prediction using deep neural representation. *IJCAI* 2018, 3371–3377.
- Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. (2017). "Neural message passing for quantum chemistry," in *Proceedings of the International Conference on Machine Learning*, (Piscataway, NJ: PMLR), 1263–1272.
- Glorigijevic, V., Renfrew, P., Kosciolk, T., Koehler, J., Cho, K., Vatanen, T., et al. (2019). Structure-based function prediction using graph convolutional networks. *Nat. Commun.* 12, 1–14.
- Goh, K., Cusick, M., Valle, D., Childs, B., Vidal, M., and Barabasi, A. (2007). The human disease network. *Proc. Natl. Acad. Sci. U.S.A.* 104, 8685–8690. doi: 10.1073/pnas.0701361104
- Gopinath, K., Desrosiers, C., and Lombaert, H. (2019). Graph convolutions on spectral embeddings for cortical surface parcellation. *Med. Image Anal.* 54, 297–305. doi: 10.1016/j.media.2019.03.012
- Gori, M., Monfardini, G., and Scarselli, F. (2005). "A new model for learning in graph domains," in *Proceedings of the 2005 IEEE International Joint Conference on Neural Networks*, Vol. 2, (Piscataway, NJ: IEEE), 729–734. doi: 10.1109/IJCNN.2005.1555942
- Grigis, A., Tasserie, J., Frouin, V., Jarraya, B., and Uhrig, L. (2020). Predicting cortical signatures of consciousness using dynamic functional connectivity graph-convolutional neural networks. *bioRxiv [Preprint]*. doi: 10.1101/2020.05.11.078535
- Guan, Y., Gorenshsteyn, D., Burmeister, M., Wong, A. K., Schimenti, J. C., Handel, M. A., et al. (2012). Tissue-specific functional networks for prioritizing phenotype and disease genes. *PLoS Comput. Biol.* 8:e1002694. doi: 10.1371/journal.pcbi.1002694
- Hamilton, W., Ying, R., and Leskovec, J. (2017). Inductive representation learning on large graphs. *NIPS* 2017, 1024–1034.
- Han, P., Yang, P., Zhao, P., Shang, S., Liu, Y., Zhou, J., et al. (2019). "GCN-MF: disease-gene association identification by graph convolutional networks and matrix factorization," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Anchorage. 705–713.
- Hansen, L., Dittmer, D., and Heinrich, M. P. (2019). "Learning deformable point set registration with regularized dynamic graph cnns for large lung motion in copd patients," in *Proceedings of the International Workshop on Graph Learning in Medical Imaging*, (Cham: Springer), 53–61.
- Harada, S., Akita, H., Tsubaki, M., Baba, Y., Takigawa, I., Yamanishi, Y., et al. (2020). Dual graph convolutional neural network for predicting chemical networks. *BMC Bioinform.* 21:3378. doi: 10.1186/s12859-020-3378-0
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, New Orleans. 770–778.
- Hu, X., Yan, Y., Ren, W., Li, H., Zhao, Y., Bayat, A., et al. (2020). Feedback graph attention convolutional network for medical image enhancement. *arXiv [Preprint]*. arXiv:2006.13863.
- Huang, Y., Huang, Z., You, Z., Zhu, Z., Huang, W., Guo, J., et al. (2019). Predicting lncRNA-miRNA interaction via graph convolution auto-encoder. *Front. Genet.* 10:758. doi: 10.3389/fgene.2019.00758
- Huang, Y. A., Hu, P., Chan, K. C., and You, Z. H. (2020). Graph convolution for predicting associations between miRNA and drug resistance. *Bioinformatics* 36, 851–858. doi: 10.1093/bioinformatics/btz621
- Hwang, D., Jeon, M., and Kang, J. (2020). "A drug-induced liver injury prediction model using transcriptional response data with graph neural network," in *Proceedings of the 2020 IEEE International Conference on Big Data and Smart Computing (BigComp)*, (Piscataway, NJ: IEEE), 323–329.
- Iglehart, J. D., and Silver, D. P. (2009). Synthetic lethality – A new direction in cancer drug development. *N. Engl. J. Med.* 361:189. doi: 10.1056/NEJMe0903044
- Ingraham, J., Garg, V., Barzilay, R., and Jaakkola, T. (2019). *Generative Models for Graph-Based Protein Design*. *Advances in Neural Information Processing Systems* 32 (NeurIPS 2019), December 2019, Vancouver, Canada, Neural Information Processing Systems Foundation, 2019. Vancouver, BC: Neural Information Processing Systems Foundation.
- Ioannidis, V. N., Marques, A. G., and Giannakis, G. B. (2019). "Graph neural networks for predicting protein functions," in *Proceedings of the 2019 IEEE 8th*

- International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), (Piscataway, NJ: IEEE), 221–225.
- Jiahua, R., Zhou, X., Lu, Y., Zhao, H., and Yang, Y. (2020). Imputing single-cell RNA-seq data by combining graph convolution and autoencoder neural networks. *iScience* 24:102393. doi: 10.1016/j.isci.2021.102393
- Jiang, L., and Zhu, J. (2020). Review of MiRNA-disease association prediction. *Curr. Prot. Pept. Sci.* 21, 1044–1053.
- Jiang, M., Li, Z., Zhang, S., Wang, S., Wang, X., Yuan, Q., et al. (2020). Drug-target affinity prediction using graph neural network and contact maps. *RSC Adv.* 10, 20701–20712. doi: 10.1039/D0RA02297G
- Jin, W., Barzilay, R., and Jaakkola, T. (2018). “Junction tree variational autoencoder for molecular graph generation,” in *Proceedings of the International Conference on Machine Learning*, (Piscataway, NJ: PMLR), 2323–2332. doi: 10.1186/s13321-019-0396-x
- Johansson-Åkhe, I., Mirabello, C., and Wallner, B. (2020). InterPepRank: assessment of docked peptide conformations by a deep graph network. *bioRxiv* [Preprint]. doi: 10.1101/2020.09.07.285957
- Kazi, A., Albarqouni, S., Kortüm, K., and Navab, N. (2018). Multi layered-parallel graph convolutional network (ML-PGCN) for disease prediction. *arXiv* [Preprint]. arXiv:1804.
- Kazi, A., Shekarforoush, S., Sridhar, A., Burwinkel, H., Vivar, G., Kortüm, K., et al. (2019a). “InceptionGCN: receptive field aware graph convolutional network for disease prediction,” in *Proceedings of the International Conference on Information Processing in Medical Imaging*, (Cham: Springer), 73–85. doi: 10.1007/978-3-030-20351-1_6
- Kazi, A., Shekarforoush, S., Sridhar, A., Burwinkel, H., Wiestler, B., et al. (2019b). “Graph convolution based attention model for personalized disease prediction,” in *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, (Cham: Springer), 122–130. doi: 10.1007/978-3-030-32251-9_14
- Kazi, A., Sridhar, A., Shekarforoush, S., Kortüm, K., Albarqouni, S., and Navab, N. (2019c). “Self-attention equipped graph convolutions for disease prediction,” in *Proceedings of the 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, (Venice: IEEE), 1896–1899. doi: 10.1109/ISBI.2019.8759274
- Kearnes, S., McCloskey, K., Berndl, M., Pande, V., and Riley, P. (2016). Molecular graph convolutions: moving beyond fingerprints. *J. Comput. Aided Mol. Des.* 30, 595–608. doi: 10.1007/s10822-016-9938-8
- Khemchandani, Y., O’Hagan, S., Samanta, S., Swainston, N., Roberts, T. J., Bollegala, D., et al. (2020). DeepGraphMolGen, a multi-objective, computational strategy for generating molecules with desirable properties: a graph convolution and reinforcement learning approach. *J. Cheminform.* 12, 1–17. doi: 10.1186/s13321-020-00454-3
- Kingma, D. P., and Welling, M. (2013). Auto-encoding variational bayes. *arXiv* [Preprint]. arXiv:1312.6114. doi: 10.1093/bioinformatics/btaa169
- Kipf, T. N., and Welling, M. (2016a). Semi-supervised classification with graph convolutional networks. *arXiv* [Preprint]. arXiv:1609.02907.
- Kipf, T. N., and Welling, M. (2016b). Variational graph auto-encoders. *arXiv* [Preprint]. arXiv:1611.07308.
- Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer* 42, 30–37.
- Ktena, S. I., Parisot, S., Ferrante, E., Rajchl, M., Lee, M., Glocker, B., et al. (2018). Metric learning with spectral graph convolutions on brain connectivity networks. *NeuroImage* 169, 431–442. doi: 10.1016/j.neuroimage.2017.12.052
- Ktena, S. I., Parisot, S., Ferrante, E., Rajchl, M., Lee, M., Glocker, B., et al. (2017). “Distance metric learning using graph convolutional networks: Application to functional brain networks,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, (Cham: Springer), 469–477.
- Lee, I., Blom, U. M., Wang, P. I., Shim, J. E., and Marcotte, E. M. (2011). Prioritizing candidate disease genes by network-based boosting of genome-wide association data. *Genome Res.* 21, 1109–1121. doi: 10.1101/gr.118992.110
- Levy, J., Haudenschild, C., Bar, C., Christensen, B., and Vaickus, L. (2020). Topological feature extraction and visualization of whole slide images using graph neural networks. *bioRxiv* [Preprint]. doi: 10.1101/2020.08.01.231639.
- Li, C., Liu, H., Hu, Q., Que, J., and Yao, J. (2019). A novel computational model for predicting microRNA-disease associations based on heterogeneous graph convolutional networks. *Cells* 8:977. doi: 10.3390/cells8090977
- Li, J., Zhang, S., Liu, T., Ning, C., Zhang, Z., and Zhou, W. (2020). Neural inductive matrix completion with graph convolutional networks for miRNA-disease association prediction. *Bioinformatics* 36, 2538–2546. doi: 10.1093/bioinformatics/btz965
- Li, Q., Han, Z., and Wu, X. M. (2018). “Deeper insights into graph convolutional networks for semi-supervised learning,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.
- Li, S., Wan, F., Shu, H., Jiang, T., Zhao, D., and Zeng, J. (2020). MONN: a multi-objective neural network for predicting compound-protein interactions and affinities. *Cell Syst.* 10, 308–322.e11. doi: 10.1016/j.cels.2020.03.002
- Li, X., Dvornek, N. C., Zhou, Y., Zhuang, J., Ventola, P., and Duncan, J. S. (2019a). “Graph neural network for interpreting task-fMRI biomarkers,” in *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, (Cham: New Orleans, Springer), 485–493. doi: 10.1007/978-3-030-32254-0_54
- Li, X., Yan, X., Gu, Q., Zhou, H., Wu, D., and Xu, J. (2019b). DeepChemStable: chemical stability prediction with an attention-based graph convolution network. *J. Chem. Inform. Model.* 59, 1044–1049. doi: 10.1021/acs.jcim.8b00672
- Li, X., Zhou, Y., Gao, S., Dvornek, N., Zhang, M., Zhuang, J., et al. (2020). Braingnn: Interpretable brain graph neural network for fMRI analysis. *bioRxiv* [Preprint]. doi: 10.1101/2020.05.16.100057
- Li, Y., and Li, J. (2012). Disease gene identification by random walk on multigraphs merging heterogeneous genomic and phenotype data. *BMC Genomics* 7(Suppl. 7):S27. doi: 10.1186/1471-2164-13-S7-S27
- Li, Y., Kuwahara, H., Yang, P., Song, L., and Gao, X. (2019). PGCN: Disease Gene Prioritization by Disease and Gene Embedding Through Graph Convolutional Neural Networks. doi: 10.1101/532226
- Li, Y., Vinyals, O., Dyer, C., Pascanu, R., and Battaglia, P. (2018a). Learning deep generative models of graphs. *arXiv* [Preprint]. arXiv:1803.03324.
- Li, Y., Zhang, L., and Liu, Z. (2018b). Multi-objective *de novo* drug design with conditional graph generative model. *J. Cheminform.* 10, 1–24. doi: 10.1186/s13321-018-0287-6
- Liu, K., Sun, X., Jia, L., Ma, J., Xing, H., Wu, J., et al. (2019). Chemi-Net: a molecular graph convolutional network for accurate drug property prediction. *Int. J. Mol. Sci.* 20:3389. doi: 10.3390/ijms20143389
- Liu, Q., Hu, Z., Jiang, R., and Zhou, M. (2020). DeepCDR: a hybrid graph convolutional network for predicting cancer drug response. *Bioinformatics* 36(Suppl. 2), i911–i918. doi: 10.1093/bioinformatics/btaa822
- Liu, X., Luo, Y., Song, S., and Peng, J. (2020). Pre-training of graph neural network for modeling effects of mutations on protein-protein binding affinity. *arXiv* [Preprint]. arXiv:2008.12473.
- Long, Y., Wu, M., Kwok, C., Luo, J., and Li, X. (2020). Predicting human microbe-drug associations via graph convolutional network with conditional random field. *Bioinformatics* 36:598. doi: 10.1093/bioinformatics/btaa598
- Luo, J., Ding, P., Liang, C., Cao, B., and Chen, X. (2016). Collective prediction of disease-associated miRNAs based on transduction learning. *IEEE/ACM Transact. Comput. Biol. Bioinform.* 14, 1468–1475. doi: 10.1109/TCBB.2016.2599866
- Ma, T., Xiao, C., Zhou, J., and Wang, F. (2018). Drug similarity integration through attentive multi-view graph auto-encoders. *arXiv* [Preprint]. arXiv:1804.10850.
- Mao, C., Yao, L., and Luo, Y. (2019). Medgcn: Graph convolutional networks for multiple medical tasks. *arXiv* [Preprint]. arXiv:1904.00326.
- Min, S., Lee, B., and Yoon, S. (2016). Deep learning in bioinformatics. *Brief. Bioinform.* 18, 851–869. doi: 10.1093/bib/bbw068
- Mirakhorli, J., and Mirakhorli, M. (2019). Graph-based method for anomaly detection in functional brain network using variational autoencoder. *bioRxiv* [Preprint]. 616367.
- Miyazaki, Y., Ono, N., Huang, M., Altaf–Ul–Amin, M., and Kanaya, S. (2020). Comprehensive exploration of target-specific ligands using a graph convolutional neural network. *Mol. Inform.* 39:1900095. doi: 10.1002/minf.201900095
- Monti, F., Boscaini, D., Masci, J., Rodolà, E., Svoboda, J., and Bronstein, M. (2017). *Geometric Deep Learning on Graphs and Manifolds Using Mixture Model CNNs*. 5425–5434. doi: 10.1109/CVPR.2017.576
- Navlakha, S., and Kingsford, C. (2010). The power of protein interaction networks for associating genes with diseases. *Bioinformatics (Oxf. Engl.)* 26, 1057–1063. doi: 10.1093/bioinformatics/btq076

- Neil, D., Briody, J., Lacoste, A., Sim, A., Creed, P., and Saffari, A. (2018). Interpretable graph convolutional neural networks for inference on noisy knowledge graphs. *arXiv*. [Preprint]. arXiv:1812.00279.
- Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., and Ng, A. Y. (2011). "Multimodal deep learning," in *ICML*.
- Nguyen, T., Le, H., Quinn, T. P., Nguyen, T., Le, T. D., and Venkatesh, S. (2021). GraphDTA: Predicting drug–target binding affinity with graph neural networks. *Bioinformatics* 37, 1140–1147. doi: 10.1093/bioinformatics/btaa921
- Pan, X., and Shen, H.-B. (2019). Inferring disease-associated MicroRNAs using semi-supervised multi-label graph convolutional networks. *iScience* 20:13. doi: 10.1016/j.isci.2019.09.013
- Pariset, S., Ktena, S. I., Ferrante, E., Lee, M., Moreno, R. G., Glocker, B., et al. (2017). "Spectral graph convolutions for population-based disease prediction," in *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, (Cham: Springer), 177–185.
- Peng, N., Poon, H., Quirk, C., Toutanova, K., and Yih, W. (2017). Cross-sentence N-ary relation extraction with graph LSTMs. *Transact. Associat. Comput. Linguist.* 5:49. doi: 10.1162/tacl_a_00049
- Rao, A., Vg, S., Joseph, T., Kotte, S., Sivadasan, N., and Srinivasan, R. (2018). Phenotype-driven gene prioritization for rare diseases using graph convolution on heterogeneous networks. *BMC Med. Genom.* 11:372. doi: 10.1186/s12920-018-0372-8
- Rathi, P. C., Ludlow, R. F., and Verdonk, M. L. (2019). Practical high-quality electrostatic potential surfaces for drug discovery using a graph-convolutional deep neural network. *J. Med. Chem.* 63, 8778–8790. doi: 10.1021/acs.jmedchem.9b01129
- Ravindra, N., Sehanobish, A., Pappalardo, J., Hafler, D., and Dijk, D. (2020). "Disease state prediction from single-cell data using graph attention networks," in *Proceedings of the ACM Conference on Health, Inference, and Learning (CHIL '20)*, (New York, NY: Association for Computing Machinery), 121–130. doi: 10.1145/3368555.3384449
- Rhee, S., Seo, S., and Kim, S. (2018). Hybrid approach of relation network and localized graph convolutional filtering for breast cancer subtype classification. *arXiv* 3527–3534. [Preprint]. doi: 10.24963/ijcai.2018/490.
- Rossi, E., Monti, F., Bronstein, M., and Lio, P. (2019). ncRNA Classification with Graph Convolutional Networks. *arXiv* [Preprint]. arXiv:1905.06515.
- Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. (2008). The graph neural network model. *IEEE Transact. Neur. Netw.* 20, 61–80. doi: 10.1109/TNN.2008.2005605
- Schulte-Sasse, R., Budach, S., Hnisch, D., and Marsico, A. (2019). "Graph convolutional networks improve the prediction of cancer driver genes," in *Artificial Neural Networks and Machine Learning – ICANN 2019: Workshop and Special Sessions. ICANN 2019. Lecture Notes in Computer Science*, Vol. 11731, eds I. Tetko, V. Kůrková, P. Karpov, and F. Theis (Cham: Springer). doi: 10.1007/978-3-030-30493-5_60
- Selsam, D., Lamm, M., Bünz, B., Liang, P., de Moura, L., and Dill, D. L. (2018). Learning a SAT solver from single-bit supervision. *arXiv* [Preprint]. arXiv:1802.03685.
- Shi, J., Wang, R., Zheng, Y., Jiang, Z., and Yu, L. (2019). Graph convolutional networks for cervical cell classification. *Cervical cell classification with graph convolutional network. Comput. Methods Program. Biomed.* 198, 105807.
- Simonovsky, M., and Komodakis, N. (2018). "Graphvae: towards generation of small graphs using variational autoencoders," in *Proceedings of the International Conference on Artificial Neural Networks*, (Cham: Springer), 412–422. doi: 10.1186/s12868-016-0283-6
- Singh, V., and Lio, P. (2019). Towards probabilistic generative models harnessing graph neural networks for disease-gene prediction. *arXiv* [Preprint]. arXiv:1907.05628.
- Singha, M., Pu, L., Busch, K., Wu, H. C., Ramanujam, J., and Brylinski, M. (2020). GraphGR: a graph neural network to predict the effect of pharmacotherapy on the cancer cell growth. *bioRxiv* [Preprint]. doi: 10.1101/2020.05.20.107458
- Stankevičiūtė, K., Azevedo, T., Campbell, A., Bethlehem, R., and Lio, P. (2020). *Population GNNs for Brain Age Prediction*. doi: 10.1101/2020.06.26.172171
- Strokach, A., Becerra, D., Corbi-Verge, C., Perez-Riba, A., and Kim, P. (2020). Fast and flexible protein design using deep graph neural networks. *Cell Syst.* 11:16 doi: 10.1016/j.cels.2020.08.016
- Sun, J., Shi, H., Wang, Z., Zhang, C., Liu, L., Wang, L., et al. (2014). Inferring novel lncRNA–disease associations based on a random walk model of a lncRNA functional similarity network. *Mol. BioSyst.* 10:70608. doi: 10.1039/c3mb70608g
- Sun, M., Zhao, S., Gilvary, C., Elemento, O., Zhou, J., and Wang, F. (2019). Graph convolutional networks for computational drug development and discovery. *Briefings Bioinform.* 21:bbz042. doi: 10.1093/bib/bbz042
- Sureka, M., Patil, A., Anand, D., and Sethi, A. (2020). "Visualization for Histopathology Images using Graph Convolutional Neural Networks," in *Proceedings of the 2020 IEEE 20th International Conference on Bioinformatics and Bioengineering (BIBE)*, (Piscataway, NJ: IEEE), 331–335.
- Tian, F., Gao, B., Cui, Q., Chen, E., and Liu, T.-Y. (2014). "Learning deep representations for graph clustering," in *Proceedings of the National Conference on Artificial Intelligence*, Piscataway, NJ, Vol. 2, 1293–1299.
- Tian, Z., Li, X., Zheng, Y., Chen, Z., Shi, Z., Liu, L., et al. (2020a). Graph-convolutional-network-based interactive prostate segmentation in MR images. *Med. Phys.* 47, 4164–4176. doi: 10.1002/mp.14327
- Tian, Z., Zheng, Y., Li, X., Du, S., and Xu, X. (2020b). Graph convolutional network based optic disc and cup segmentation on fundus images. *Biomed. Optics Exp.* 11, 3043–3057. doi: 10.1364/BOE.390056
- Toomer, D. (2020). *Predicting Protein Functional Sites Through Deep Graph Convolutional Neural Networks on Atomic Point-Clouds*. Available online at: <https://cs230.stanford.edu/> (accessed November 15, 2020).
- Torng, W., and Altman, R. B. (2019). Graph convolutional neural networks for predicting drug–target interactions. *J. Chem. Inform. Model.* 59, 4131–4149.
- Tsubaki, M., Tomii, K., and Sese, J. (2018). Compound–protein interaction prediction with end-to-end learning of neural networks for graphs and sequences. *Bioinformatics (Oxf. Engl.)* 35:535. doi: 10.1093/bioinformatics/bty535
- Uhl, M., Tran Van, D., Heyl, F., and Backofen, R. (2019). GraphProt2: A novel deep learning-based method for predicting binding sites of RNA-binding proteins. *bioRxiv* [Preprint]. doi: 10.1101/850024
- Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. (2017). Graph attention networks. *arXiv* [Preprint]. arXiv:1710.10903,
- Vohora, D., and Singh, G. (2017). *Pharmaceutical Medicine and Translational Clinical Research*. Cambridge, MA: Academic Press.
- Wang, D., Cui, P., and Zhu, W. (2018). *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY: ACM, 1225–1234.
- Wang, J., Ma, A., Chang, Y., Gong, J., Jiang, Y., Qi, R., et al. (2021). scGNN is a novel graph neural network framework for single-cell RNA-Seq analyses. *Nat. Commun.* 12, 1–11. doi: 10.1038/s41467-021-22197-x
- Wang, L., You, Z.-H., Li, Y.-M., Zheng, K., and Huang, Y.-A. (2020). GCNCDA: a new method for predicting circRNA–disease associations based on graph convolutional network algorithm. *PLoS Comput. Biol.* 16:e1007568. doi: 10.1371/journal.pcbi.1007568
- Wang, Q., Sun, M., Zhan, L., Thompson, P., Ji, S., and Zhou, J. (2017). "Multi-modality disease modeling via collective deep matrix factorization," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Halifax. 1155–1164.
- Wang, T., Shao, W., Huang, Z., Tang, H., Ding, Z., and Huang, K. (2020). MORONET: multi-omics integration via graph convolutional networks for biomedical data classification. *bioRxiv* [Preprint]. doi: 10.1101/2020.07.02.184705
- Wang, X., Gong, Y., Yi, J., and Zhang, W. (2019a). "Predicting gene–disease associations from the heterogeneous network using graph embedding," in *Proceedings of the 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, (Piscataway, NJ: IEEE), 504–511.
- Wang, X., Ye, Y., and Gupta, A. (2018). "Zero-shot recognition via semantic embeddings and knowledge graphs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City. 6857–6866.
- Wang, X., Zhang, L., Roth, H., Xu, D., and Xu, Z. (2019b). "Interactive 3D segmentation editing and refinement via gated graph neural networks," in *Graph Learning in Medical Imaging. GLMI 2019. Lecture Notes in Computer Science*, Vol. 11849, eds D. Zhang, L. Zhou, B. Jie, and M. Liu (Cham: Springer), doi: 10.1007/978-3-030-35817-4_2

- Weininger, D. (1988). SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inform. Comput. Sci.* 28, 31–36.
- Wood, M., and Hirst, J. (2004). “Recent applications of neural networks in bioinformatics. biological and artificial intelligence environments,” in *Proceedings of the 15th Italian Workshop on Neural Nets, WIRN VIETRI 2004, Vietri sul Mare, Italy, 2004*. DBLP, Vietri sul Mare.
- Wu, J., Zhong, J. X., Chen, E. Z., Zhang, J., Jay, J. Y., and Yu, L. (2019). “Weakly- and Semi-supervised Graph CNN for identifying basal cell carcinoma on pathological images,” in *International Workshop on Graph Learning in Medical Imaging*, (Cham: Springer), 112–119.
- Wu, X., Lan, W., Chen, Q., Dong, Y., Liu, J., and Peng, W. (2020). Inferring lncRNA-disease associations based on graph autoencoder matrix completion. *Comput. Biol. Chem.* 87:107282. doi: 10.1016/j.compbiolchem.2020.107282
- Xuan, P., Pan, S., Zhang, T., Liu, Y., and Sun, H. (2019a). Graph Convolutional network and convolutional neural network based method for predicting lncRNA-disease associations. *Cells* 8:1012. doi: 10.3390/cells8091012
- Xuan, P., Sun, H., Wang, X., Zhang, T., and Pan, S. (2019b). Inferring the disease-associated mirnas based on network representation learning and convolutional neural networks. *Int. J. Mol. Sci.* 20:3648. doi: 10.3390/ijms20153648
- Yan, Z., Hamilton, W., and Blanchette, M. (2020). Graph neural representational learning of RNA secondary structures for predicting RNA-protein interactions. *Bioinformatics* 36, i276–i284. doi: 10.1093/bioinformatics/btaa456
- Yang, H., Li, X., Wu, Y., Li, S., Lu, S., Duncan, J., et al. (2019). “Interpretable multimodality embedding of cerebral cortex using attention graph,” in *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, (Cham: Springer), 799–807.
- Yao, H., Guan, J., and Liu, T. (2020). Denoising protein-protein interaction network via variational graph auto-encoder for protein complex detection. *J. Bioinform. Comput. Biol.* 18:2040010. doi: 10.1142/S0219720020400107
- You, J., Liu, B., Ying, R., Pande, V., and Leskovec, J. (2018). Graph convolutional policy network for goal-directed molecular graph generation. *arXiv [Preprint]*. arXiv:1806.02473.
- Zamora-Resendiz, R., and Crivelli, S. (2019). Structural learning of proteins using graph convolutional neural networks structural learning of proteins using graph convolutional neural networks. *bioRxiv [Preprint]*. doi: 10.1101/610444
- Zeng, Y., Zhou, X., Jiahua, R., Lu, Y., and Yang, Y. (2020). “Accurately clustering single-cell rna-seq data by capturing structural relations between cells through graph convolutional network,” in *Proceedings of the 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, Vol. 2020, Piscataway, NJ, 519–522. doi: 10.1109/BIBM49941.2020.9313569
- Zhai, Z., Staring, M., Zhou, X., Xie, Q., Xiao, X., Bakker, M. E., et al. (2019). “Linking convolutional neural networks with graph convolutional networks: application in pulmonary artery-vein separation,” in *Proceedings of the International Workshop on Graph Learning in Medical Imaging*, (Cham: Springer), 36–43.
- Zhang, C., Song, D., Huang, C., Swami, A., and Chawla, N. V. (2019). “Heterogeneous graph neural network” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New Orleans, 793–803.
- Zhang, J., Hu, X., Jiang, Z., Song, B., Quan, W., and Chen, Z. (2019). “Predicting disease-related RNA associations based on graph convolutional attention network,” in *Proceedings of the 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, San Diego, CA, 177–182. doi: 10.1109/BIBM47256.2019.8983191
- Zhang, M., and Chen, Y. (2018). Link prediction based on graph neural networks. *arXiv [Preprint]*. arXiv:1802.09691.
- Zhang, X., He, L., Chen, K., Luo, Y., Zhou, J., and Wang, F. (2018). “Multi-view graph convolutional network and its applications on neuroimage analysis for parkinson’s disease,” in *Proceedings of the AMIA Annual Symposium Proceedings*, Vol. 2018, (Bethesda, MY: American Medical Informatics Association), 1147.
- Zhang, Y., and Pierre, B. (2020). Transferability of brain decoding using graph convolutional networks. *bioRxiv [Preprint]*. doi: 10.1101/2020.06.21.163964 doi: 10.1016/j.compmedimag.2020.101748
- Zhang, Y., Tetrel, L., Thirion, B., and Bellec, P. (2021). Functional annotation of human cognitive states using deep graph convolution. *NeuroImage* 231:117847. doi: 10.1016/j.neuroimage.2021.117847
- Zhao, T., Hu, Y., Peng, J., and Cheng, L. (2020). GCN-CNN: a novel deep learning method for prioritizing lncRNA target genes. *Bioinformatics (Oxf. Engl.)* 36:428. doi: 10.1093/bioinformatics/btaa428
- Zhao, T., Hu, Y., Valsdottir, L. R., Zang, T., and Peng, J. (2021). Identifying drug-target interactions based on graph convolutional network and deep neural network. *Brief. Bioinform.* 22, 2141–2150. doi: 10.1093/bib/bbaa044
- Zheng, K., You, Z. H., Wang, L., Wong, L., and Chen, Z. H. (2020). “Inferring disease-associated Piwi-interacting RNAs via graph attention networks,” in *Proceedings of the International Conference on Intelligent Computing*, (Cham: Springer), 239–250.
- Zhong, F., Wu, X., Li, X., Wan, D., Zunyun, F., Liu, X., et al. (2020). Computational target fishing by mining transcriptional data using a novel Siamese spectral-based graph convolutional network. *bioRxiv [Preprint]*. doi: 10.1101/2020.04.01.019166
- Zhou, H., and Skolnick, J. (2016). A knowledge-based approach for predicting gene-disease associations. *Bioinformatics* 32, 2831–2838. doi: 10.1093/bioinformatics/btw358
- Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., et al. (2020). Graph neural networks: a review of methods and applications. *AI Open* 1, 57–81. doi: 10.1016/j.aiopen.2021.01.001
- Zhou, M., Wang, X., Li, J., Hao, D., Wang, Z., Shi, H., et al. (2015). Prioritizing candidate disease-related long non-coding RNAs by walking on the heterogeneous lncRNA and disease network. *Mol. BioSyst.* 11, 760–769. doi: 10.1039/c4mb00511b
- Zhou, Y., Graham, S., Alemi Koohbanani, N., Shaban, M., Heng, P. A., and Rajpoot, N. (2019). “Cgc-net: Cell graph convolutional network for grading of colorectal cancer histology images,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, Seoul. doi: 10.1109/TMI.2020.2971006
- Zitnik, M., Agrawal, M., and Leskovec, J. (2018). Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics* 34, i457–i466.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher’s Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2021 Zhang, Liang, Liu and Tang. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.



Improving *de novo* Molecule Generation by Embedding LSTM and Attention Mechanism in CycleGAN

Feng Wang^{1,2}, Xiaochen Feng¹, Xiao Guo¹, Lei Xu³, Liangxu Xie^{3*} and Shan Chang^{3*}

¹ Changzhou University Huaide College, Taizhou, China, ² School of Computer Science and Artificial Intelligence, Aliyun School of Big Data, Changzhou University, Changzhou, China, ³ Institute of Bioinformatics and Medical Engineering, Jiangsu University of Technology, Changzhou, China

OPEN ACCESS

Edited by:

Xiangxiang Zeng,
Hunan University, China

Reviewed by:

Leyi Wei,
Shandong University, China
Hui Ding,
University of Electronic Science
and Technology of China, China
Jianmin Wang,
Yonsei University, South Korea

*Correspondence:

Liangxu Xie
xieliangxu@jsut.edu.cn
Shan Chang
schang@jsut.edu.cn

Specialty section:

This article was submitted to
Computational Genomics,
a section of the journal
Frontiers in Genetics

Received: 14 May 2021

Accepted: 19 July 2021

Published: 05 August 2021

Citation:

Wang F, Feng X, Guo X, Xu L,
Xie L and Chang S (2021) Improving
de novo Molecule Generation by
Embedding LSTM and Attention
Mechanism in CycleGAN.
Front. Genet. 12:709500.
doi: 10.3389/fgene.2021.709500

The application of deep learning in the field of drug discovery brings the development and expansion of molecular generative models along with new challenges in this field. One of challenges in *de novo* molecular generation is how to produce new reasonable molecules with desired pharmacological, physical, and chemical properties. To improve the similarity between the generated molecule and the starting molecule, we propose a new molecule generation model by embedding Long Short-Term Memory (LSTM) and Attention mechanism in CycleGAN architecture, LA-CycleGAN. The network layer of the generator in CycleGAN is fused head and tail to improve the similarity of the generated structure. The embedded LSTM and Attention mechanism can overcome long-term dependency problems in treating the normally used SMILES input. From our quantitative evaluation, we present that LA-CycleGAN expands the chemical space of the molecules and improves the ability of structure conversion. The generated molecules are highly similar to the starting compound structures while obtaining expected molecular properties during cycle generative adversarial network learning, which comprehensively improves the performance of the generative model.

Keywords: LSTM, attention mechanism, Mol-CycleGAN, head-to-tail feature fusion, LA-CycleGAN

INTRODUCTION

Computer-aided drug Design (CADD) promotes the speed of drug discovery (Macalino et al., 2015). Beyond the traditional CADD, artificial intelligence (AI) is widely used in the process of new drug screening and optimization. AI is realized by using various kinds of machine learning or deep learning (DL) algorithms (Goodfellow et al., 2016; Laveccchia, 2019). Among different methods, DL method generally trains a large amount of sample data

Abbreviations: GAN, Generative Adversarial Networks; DL, Deep Learning; VAE, variational autoencoder; JT-VAE, Junction Tree Variational Autoencoder; LSTM, Long Short-Term Memory; CBDD, Computer-aided drug Design; RNN, recurrent neural network.

through neural networks to learn the molecular structure of the sample. Different from the traditional similar ligand searching, the DL model obtains the characteristic information and general rules from learning prior knowledge during the training process. In the field of DL, generative models can generate novel compounds effectively with the desired properties, which would reduce the cost of drug discovery (Jing et al., 2018; Wainberg et al., 2018). Drug discovery and design (Muegge et al., 2017; Agrawal, 2018) use the knowledge of available biomedicine (Mamoshina et al., 2016; Tang et al., 2019) to define the parameters and indicators that required by each drug molecule in the following processes: (1) Selection and confirmation of drug targets; (2) Discovery of seed compound (Hit); (3) Discovery and optimization of lead compounds (Lead); (4) Discovery of Candidate. Among the processes, molecule generation (Xu et al., 2019) emerges as new tool in the hit-to-lead and lead optimization phases of drug discovery.

The generation model (Jensen, 2019; Walters and Murcko, 2020) is used to generate new molecules with similar molecular activity to the trained compounds, and to learn the distribution characteristics of data by using unsupervised learning (Radford et al., 2015). The generative models normally use SMILES (Weininger, 1988; Öztürk et al., 2016) grammar based on ASCII characters and molecular graphs based on Graph (Li et al., 2018; Lim et al., 2020) to describe molecules at the atomic level. For example, recurrent neural network (RNN) can generate a larger chemical space than the training set by using SMILES grammar to input a small part of molecular data set (Arús-Pous et al., 2019). Based on deep learning methods, several deep generative models have been proposed. Deep generative models are roughly divided into four categories: (1) Based on the Auto Encoder (AE) model used in semi-supervised learning and unsupervised learning; (2) Generative Adversarial Networks (GAN)(Goodfellow et al., 2014) model, which is composed of two architectures: generator and discriminator, and one-way generation that confronts each other during the training process; (3) Model based on RNN (Méndez-Lucio et al., 2020); (4) Hybrid model based on the combination of deep generative model and reinforcement learning (RL) (Grisoni et al., 2020).

Different architectures are ongoing developed to generate more realistic molecules (Sarmad et al., 2019). Character-level recurrent neural network (CharRNN) (Segler et al., 2018) is trained on the SMILES molecular database. CharRNN is a chemical language model based on SMILES grammar specifications. It uses Maximum Likelihood Estimation (MLE) to optimize model parameters and improve the structural similarity of the generated molecules. CharRNN can generate molecules with new pharmacological properties. Variational Autoencoder (VAE) (Gómez-Bombarelli et al., 2018; Simonovsky and Komodakis, 2018) is a kind of “encoder-decoder” architecture. It proposes a new method based on the continuous coding of molecules to explore the chemical space. This model map high-dimensional data to latent space, and perform a new search based on directional gradients in chemical space. The Adversarial Autoencoder (AAE) (Makhzani et al., 2015; Kadurin et al., 2017) combines the ideas of adversarial training in VAE and GAN for the first time. Junction Tree Variational Autoencoder

(JTN-VAE) (Jin et al., 2018) directly uses molecular graph expression. JTN-VAE alternatively generates linear SMILES strings to complete the task of molecule generation. First, the chemical substructure on the generated tree object is automatically decomposed, and the substructure on the training set is extracted. Then, these substructures are combined into a molecular map. Based on GAN, LatentGAN (Prykhodko et al., 2019) combines an autoencoder and a generative adversarial to carry out new molecular designs. CycleGAN (Zhu et al., 2017) combines two symmetrical GANs into a ring network, which has two generators and two discriminators to perform two data conversions. The Mol-CycleGAN model (Maziarka et al., 2020) extends CycleGAN to the JT-VAE framework to ensure that the generated compounds are always effective. The original molecular data set is entered into the “encoder-decoder” architecture in order to obtain a novel compound that is similar to the original molecular structure with the required pharmacological properties. Mol-CycleGAN does not use SMILES grammar and atomic ordering, which is a heavy training burden. The decoding method selected by the model is combined with the JT-VAE based on the Graph form, so that the generated molecule is always effective.

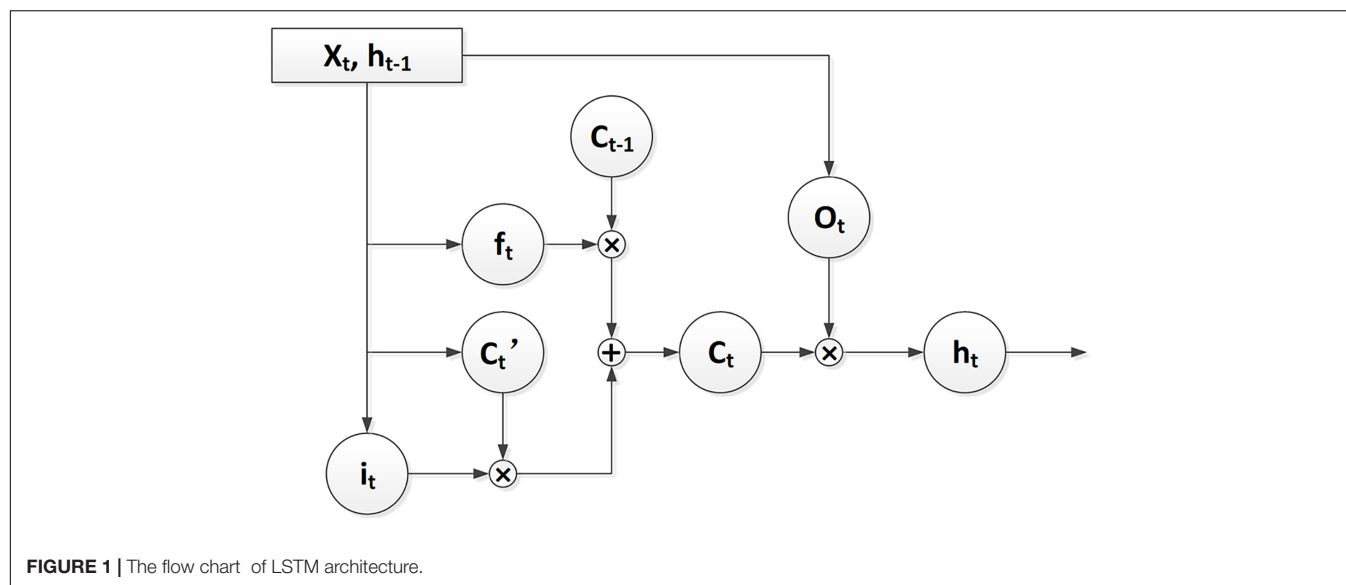
In this article, we proposed LA-CycleGAN by embedding CycleGAN architecture with long and short-term memory module (LSTM) (Mao et al., 2017) and Attention mechanism. We optimized the generator and discriminator of CycleGAN with the aim to improve the training performance of the network model and expand the chemical space. LSTM and Attention mechanism are used to solve long-term dependency problems when treating SMILES grammar inputs. The network layer of the generator of the confrontation network is merged end to end. Least Squares Generative Adversarial Network (LSGAN) (Yasonik, 2020) is used to learn the corresponding transformation by minimizing the loss. The results show that the optimized model improves the ability of structural transformation. The structure similarity between the generating molecule and the starting molecule is increased, which should satisfy the specific requirements for drug production.

MATERIALS AND METHODS

LSTM

Long Short-Term Memory (LSTM) is a network structure extended from the RNN. This structure is mainly used to solve the problem of gradient disappearance and gradient explosion in the long sequence training process of SMILES grammar. The LSTM structure is shown in **Figure 1**.

Long Short-Term Memory (LSTM) can control the transmission state through three gating states. With the gating states, LSTM can process the characterization of various sequence lengths and perform globalization processing. Therefore, the SMILES can still have a better characterization when the SMILES type is longer. f_t stands for forget gate, whose structure purposefully deletes or adds information to the input sequence, and effectively handles the generation of molecular structures described in SMILES form. i_t stands for input gate used to update



the unit state and determine whether the information is stored. C_t represents the output used by the output gate to control. As shown in Eq. (1).

$$h_t = \underbrace{\sigma(W_o \cdot [h_{t-1}, x_t] + b_o)}_{o_t} * \tanh \left(\underbrace{\left(\underbrace{\sigma(W_f \cdot [h_{t-1}, x_t] + b_f)}_{f_t} * C_{t-1} + \underbrace{\sigma(W_i \cdot [h_{t-1}, x_t] + b_i)}_{i_t} * \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)}_{C'_t} \right)}_{C_t} \right) \quad (1)$$

h_{t-1} represents the output of the previous state and x_t represents the input to the current state. σ represents the activation function of sigmoid on the gate and \tanh represents the activation function on the state and output. W represents the weight, b represents the bias, and h_t represents the output of the current state. \otimes is unit times, and \oplus is unit plus.

Attention Mechanism

The Attention mechanism (Vaswani et al., 2017) also handles the chemical structural formula sequence globally. The vector generated by the LSTM module cannot fully represent the information of the entire sequence. The attention mechanism can retain the intermediate output results of the LSTM encoder on the input sequence. It is used to supplement the SMILES information lost by LSTM and perform secondary learning on the features that have not been learned. The process of re-modeling global information can effectively improve the performance of the model. The Attention mechanism is similar to the task mechanism of LSTM, which overcomes the problem of information loss in the feature expression process. It effectively

focuses on the output results of the encoder according to the model target. The Attention mechanism is put at the bottom of the context vector insertion layer. The output vector of the LSTM is used as the input of the Attention mechanism, which is used to calculate the vector probability distribution of the feature. The method can capture the global information of the chemical formula structure. Attention mechanism solves the long-distance dependence of input sequence in RNN. The function of Attention mechanism is as shown in Eq. (2).

$$\text{Attention}(\text{Query}, \text{Key}, \text{Value}) = \text{softmax} \left(\frac{\text{Query} \odot \text{Key}^T}{\sqrt{d}} \right) * \text{Value} \quad (2)$$

Query represents the current molecule fragment, *Key* represents each molecule, and d is the vector dimension of *query* and *key*. *Softmax* is used to make the probability distribution of the result. *Value* represents the current molecule, and re-obtain important information. The attention mechanism can make deep neural network interpretable by capturing the important features. Attention mechanism assigns attention scores for the input features. The attention score can be interpreted as the importance of the feature, which will filter out the useless molecular feature information.

Model Architecture of LA-CycleGAN

During the optimization process of the model generated by Mol-CycleGAN, LSTM neural network and Attention mechanism are embedded in the framework of CycleGAN. The internal network layers of the generator and the discriminator are merged end to end. The first and last layers of the internal network layer are merged for the generator and the discriminator. During the training process, we use LSGAN loss and Batch Normalization (BN) (Ioffe and Szegedy, 2015). Mol-CycleGAN directly uses JT-VAE to generate latent vectors, which is convenient for molecular

graphic expression. The purpose of CycleGAN is to learn the method from the original molecular data domain X to the target molecular data domain Y .

Through training mapping $G: X \rightarrow Y$ and reverse mapping $F: Y \rightarrow X$, $F(G(X)) \approx X$, and $G(F(Y)) \approx Y$ are established at the same time. In order to prevent generators G and F from stopping the conversion function after generating data, we use cycle consistency loss as an incentive, as in Eq. (3).

$$L_{cyc}(G, F) = E_{x \sim P_{data}(x)}[||F(G(x)) - x||_1] + E_{y \sim P_{data}(y)}[||G(F(y)) - y||_1] \quad (3)$$

In order to prevent overfitting between input and output, the identity mapping loss is used to ensure that the generated molecule is close to the starting molecule, as in Eq. (4).

$$L_{identity}(G, F) = E_{y \sim P_{data}(y)}[||F(y) - y||_1] + E_{x \sim P_{data}(x)}[||G(x) - x||_1] \quad (4)$$

In order to ensure that the two generators can achieve mutual inversion, the overall loss function is used, where $\lambda_1 = 0.4$, $\lambda_2 = 0.15$, as in Eq. (5).

$$L(G, F, D_X, D_Y) = L_{GAN}(G, D_Y, Y, X) + L_{GAN}(F, D_X, Y, X) + \lambda_1 L_{GAN}(G, D_Y, Y, X) + \lambda_2 L_{identity}(G, F) \quad (5)$$

Where D_X is used to distinguish between X and $F(Y)$, and D_Y is used to distinguish between Y and $G(X)$. The two

generators simultaneously carry out the reverse process of mutual fight against between optimization reduction and optimization increase. The overall loss function of the state is optimized according to Eq. (6).

$$G^*, F^* = \operatorname{argmin}_{G, F} \max_{D_X, D_Y} L(G, F, D_X, D_Y) \quad (6)$$

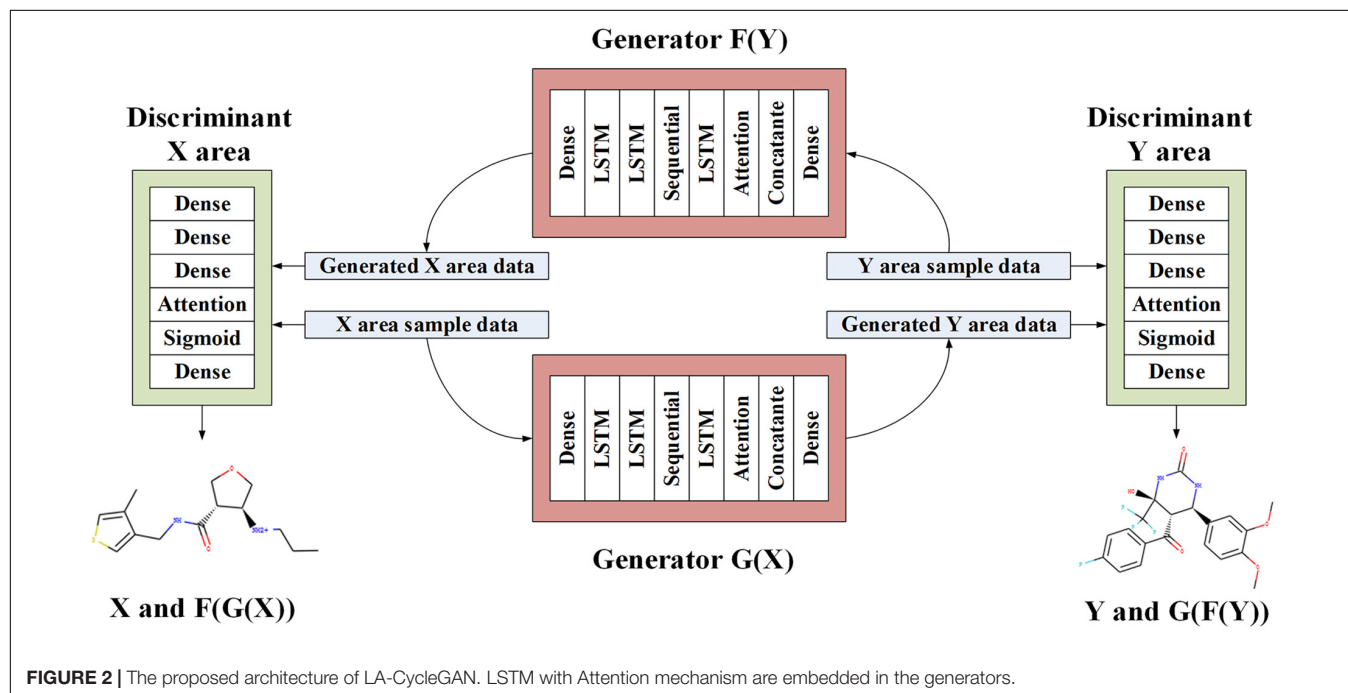
LS-GAN's confrontation loss is introduced, as in Eq. (7).

$$L_{cyc}(G, D_Y, X, Y) = \frac{1}{2} E_{x \sim P_{data}(x)} [(D_Y(G(X)))^2] + \frac{1}{2} E_{y \sim P_{data}(y)} [(D_Y(y) - 1)^2] \quad (7)$$

The main idea of the molecular optimization method is to obtain molecular descriptors, which make the generated chemical molecules easier to synthesize and generate molecules similar to the original molecules.

Workflow

As shown in **Figure 2**, GAN also generates discrete sequences when training potential chemical spaces. Then, the encoder can force the latent space to generate some continuously distributed sequences when training on the molecular data set. We construct a self-encoder in the generator of CycleGAN, and use the input data sequence as the learning target for characterization learning. The generator is composed of three components: encoder, converter, and decoder. The encoder is responsible for converting the SMILES string into a digital representation feature. The converter performs information conversion. The decoder reconstructs the features and obtains a new SMILES string. An encoded SMILES string is input into an LSTM model with 56 hidden neurons, which is constructed as a single-layer



network. The encoder obtains the characteristic information and distribution law of the input original sequence data through LSTM. The vector output by the encoder through multiple feature combination is input to the converter constructed by the linear structure Sequential model. The latent vector output by the encoder is input to the converter. The converter plays a transitional role, converting various molecules from the source domain to the target domain, avoiding information loss. The LSTM module acts as a decoder. The vector of each output vector table of the decoder unit is restored to the size of the starting vector, and regain the characteristic information of low-level molecules. Then, the decoded molecules still retain the structure or characteristics of the original molecular data, and are converted into samples in the target domain.

We map the output vector of the decoder to the Attention mechanism, perform learning mapping on the vector output from the encoder and then calculate the distribution probability. The output feature information is fused into the layer fusion module to ensure that each original molecule input can be directly input to the final fully connected layer. In LA-CycleGAN, the use of head-to-tail feature fusion plays an important role in improving the similarity of molecular generation. In the process of generating, the low-level feature information at the head is accepted by the high-level feature information at the tail. By using the feature fusion, a thicker feature can be obtained from splicing the channel dimensions of the features. Therefore, the generated molecular dataset is closer to the molecules of the original data domain. The original feature information is retained, and the deviation between the generated sample and the original sample is reduced. The original molecular features remain in the generated molecular features, resulting in an effective molecular structure.

The discriminator inputs the original SMILES data set and the generated data together. The discriminator is composed of multiple dense layers to extract feature vectors from the data. In order to determine whether these features belong to a specific category. The output feature vector of the Attention mechanism is processed by the feedforward network layer with sigmoid activation, and the probability of sampling each character of the known character set in the data set is feedback. The last layer of the discriminator network is the dense layer and produce one-dimensional output. It is used to realize the judgment of the similarity difference of the generated molecules to continue the training of the discriminator. The network structure parameters of the new model are shown in **Table 1**.

Data Set

ZINC database (Sterling and Irwin, 2015) is a molecule structure database. The deep generative model requires training on a large amount of data to be able to learn patterns that can generalize and generate new molecules. Therefore, a molecular data set is extracted from the ZINC database containing 250,000 drugs. The chemical characteristics of drug molecules are generally defined by FeatureType and FeatureFamily. Each FeatureType uses smarts expressions to describe the mode and attributes of the molecule, and summarize and characterize the molecular structure from different degrees.

TABLE 1 | LA-CycleGAN network layer structure parameters.

Network layer structure (generator)	Units	Network layer structure (Discriminator)	Units
Input	56	Input	56
Dense	56	Dense	56
LSTM	56	Dense	28
LSTM	28	Dense	56
LSTM	56	Attention	56, 64, 1
Attention	56, 64, 1	activation	Sigmoid
concatenate	57	Dense	1
Dense	56	–	–

Molecules with the same activity generally have common chemical characteristics. FeatureFamily classifies features as a whole to achieve the matching effect of pharmacophore (Nastase et al., 2019), which is an effective way to determine whether a molecule has a certain type of pharmacodynamic characteristics. It can assist the structural design of drug molecules. The pharmacophore model is a model based on the characteristic elements of pharmacodynamics. Pharmacophore includes Aliphatic Rings, Aromatic Rings, Hydrogen-Bonding Acceptor (HBA), Hydrogen-Bond Donor (HBD), and other characteristic elements, which can test the ability of molecular structure transformation. The characteristic elements of the pharmacophore are incorporated into one of the standards for the generation of compound molecules, which can effectively avoid the generation of molecular structures with large errors in similar compounds. The new type of compound generated is contrary to the design requirements. Then, we select X and Y with different structure distributions, and test whether our model can learn transformation rules and apply them to molecules that the model has not seen before. According to the characteristics of the pharmacophore elements, the datasets with different features are divided as follows:

- **Aliphatic Rings:** Aliphatic Ring compounds refer to the hydrocarbon group in the molecule containing a carbocyclic ring, and this carbocyclic ring can be saturated and unsaturated. The molecule in X has exactly 1 alicyclic ring, while the molecule in Y has 2 or 3 Aliphatic Rings.
- **Aromatic Rings:** The molecule in X has exactly 2 Aromatic Rings, while the molecule in Y has 1, 3, or 4 Aromatic Rings.
- **Hydrogen bond acceptor (HBA):** The electronegative atom is the hydrogen acceptor. The molecule in X has 1 hydrogen bond, and the molecule in Y has 2–3 hydrogen bond acceptors.
- **Hydrogen bond donor (HBD):** The molecule in X is a hydrogen bond, and the molecule in Y is 2, 3, or 4 hydrogen bond donors.

Performance Evaluation Index

We use the evaluation indicators provided by MOSES (Polykovskiy et al., 2020) to evaluate the generated molecules. The generative model is evaluated by comparing the fragment similarity, scaffold similarity, nearest neighbor similarity, Tanimoto coefficient, Fréchet ChemNet Distance and internal diversity of the generated set *G*. **Valid** judges the generated

SMILES string and checks the consistency of the valence and chemical bond of the generated molecule. **Filters** are part of the generated molecules, which pass the filters applied during the construction of the dataset to remove molecules containing charged atoms. **Novelty** is the proportion of generated molecules that do not appear in the training set, and low novelty represents overfitting. **Success rate** represents the success rate of molecular structure transformation. **Uniqueness** checks whether the model will collapse and only a few typical molecules are generated. **Non-identity** is the score when the generated molecule is different from the starting molecule.

Fragment similarity (Frag) is defined as the cosine distance between fragment frequency vectors, such as Eq. (8).

$$\text{Frag}(G, R) = 1 - \cos(F_G, F_R) \quad (8)$$

For the fragment frequency vectors F_G and F_R of molecule G and molecule R , the size is equal to the vocabulary of all chemical fragments in the data set. The corresponding molecular element indicates the frequency of the corresponding fragment in the molecular set. This metric shows the similarity of the two groups of molecules at the chemical fragment level Degree and distance definition.

Nearest neighbor similarity (SNN) is the average Tanimoto similarity between the generated molecule and the nearest molecule in the test set, as in the Eq. (9).

$$\text{SNN}(G, R) = \frac{1}{|G|} \sum_{m_G \in G} \max_{m_R \in R} T(m_G, m_R) \quad (9)$$

The chemical structure encoded in the fingerprint, the nearest neighbor molecule m_R from the test set R and the molecule m_G in the generating set G .

Internal Diversity (IntDiv_p) is the average pair-wise similarity of generated molecules, which is used to evaluate the chemical diversity in the generating set, as shown in Eq. (10).

$$\text{IntDiv}_p(G) = 1 - \sqrt[p]{\frac{1}{|G|^2} \sum_{m_1, m_2} T(m_1, m_2)^p} \quad (10)$$

FréchetChemNet Distance (FCD) can predict the biological activity of drugs, as shown in Eq. (11).

$$\text{FCD}(G, R) = \|\mu_G - \mu_R\|^2 + \text{Tr}(\sum G + \sum R - 2(\sum G \sum R)^{\frac{1}{2}}) \quad (11)$$

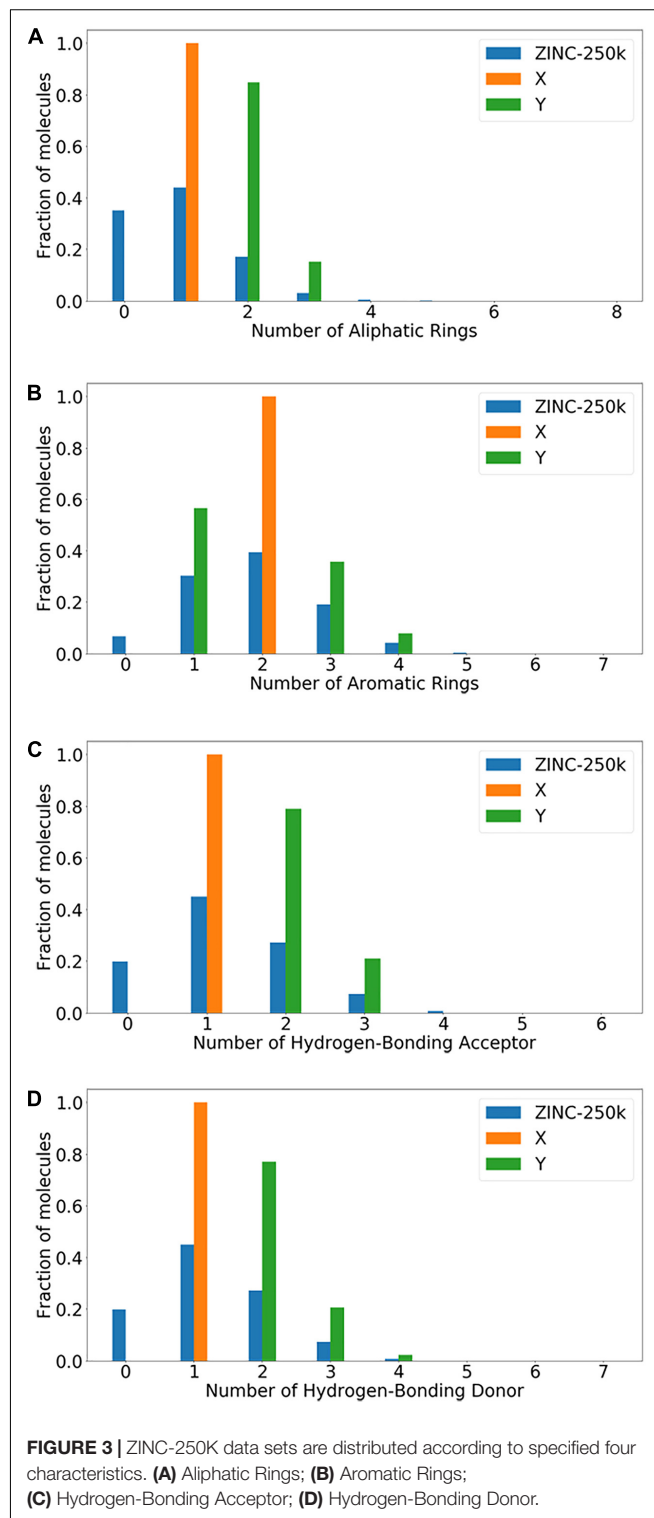
μ_G and μ_R are mean vectors, $\sum G$ and $\sum R$ are the covariance matrix of the penultimate layer activity on sets G and R .

TABLE 2 | Structural transformations and dataset sizes.

Dataset	Aliphatic Rings	Aromatic Rings	HBA	HBD
X_{train}	40,000	80,000	75,000	75,000
X_{test}	69,682	18,220	37,149	37,149
Y_{train}	40,000	80,000	75,000	75,000
Y_{test}	10,329	53,717	10,764	12,785

Tanimoto coefficient based on molecular fingerprints is used to judge the degree of correlation between two data, as shown in Eq. (12).

$$T(G, R) = \frac{\sum_i G_i \cap R_i}{\sum_i G_i \cup R_i} \quad (12)$$



Scaffold similarity (Scaff) is the cosine distance between the frequency vectors of the molecular scaffold as in Eq. (13).

$$Scaff(G, R) = 1 - \cos(S_G, S_R)$$

(13)

S_G and S_R represent the frequency of the scaffold in molecule G and molecule R .

The above six indicators comprehensively examine the characteristics of the generated molecules. In order to quantitatively compare the distribution of the generated set and the test set, we use the following three auxiliary indicators:

Molecular weight (MW): It is the sum of the atomic weights in the molecule.

logP: It reflects the distribution of a substance in oil and water. This value is the logarithmic value of the ratio

TABLE 3 | Structure conversion assessment of generated molecules.

Model test		X → G(X)			Y → F(Y)		
Data	Model	Success rate	Uniqueness	Non-identity	Success rate	Uniqueness	Non-identity
Aliphatic Rings	Mol-CycleGAN	0.534	0.976	0.908	0.422	0.990	0.890
	LA-CycleGAN	0.617	0.982	0.998	0.494	0.991	0.999
Aromatic Rings	Mol-CycleGAN	0.535	0.986	0.908	0.421	0.995	0.889
	LA-CycleGAN	0.536	0.990	0.997	0.427	0.999	0.997
HBA	Mol-CycleGAN	0.608	0.987	0.697	0.378	0.987	0.684
	LA-CycleGAN	0.612	0.996	0.995	0.382	0.993	0.999
HBD	Mol-CycleGAN	0.602	0.985	0.658	0.380	0.988	0.648
	LA-CycleGAN	0.612	0.991	0.996	0.420	0.994	0.998

The item with the highest data value in the column is indicated in bold.

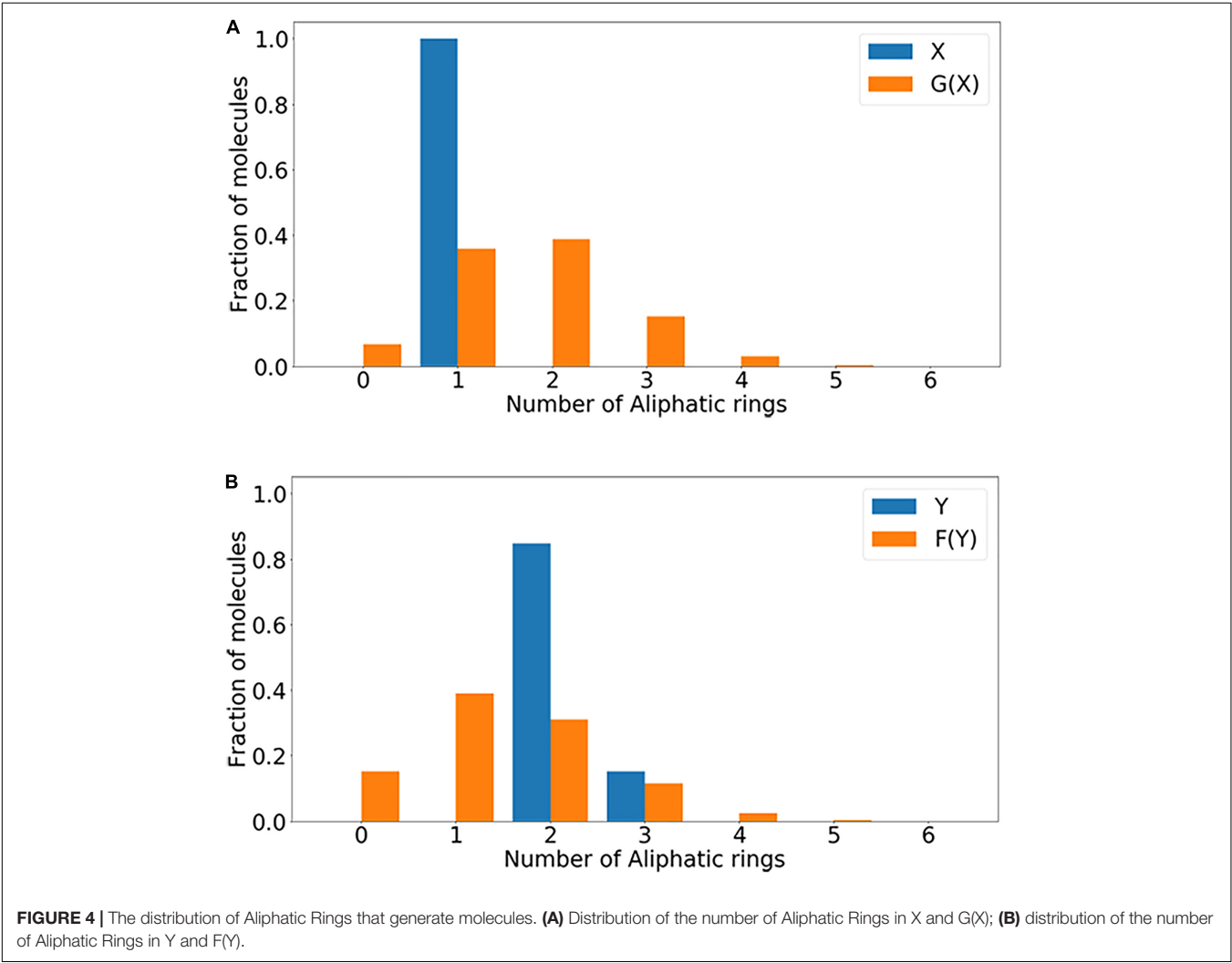


FIGURE 4 | The distribution of Aliphatic Rings that generate molecules. (A) Distribution of the number of Aliphatic Rings in X and G(X); (B) distribution of the number of Aliphatic Rings in Y and F(Y).

of the partition coefficient between n-octanol and water. The larger the value of $\log P$ is, the more fat-soluble the substance is.

Synthetics Accessibility Score (SA): It used to evaluate the difficulty of compound synthesis.

RESULTS AND DISCUSSION

Composition of Datasets

The data set is divided based on Aliphatic Rings, Aromatic Rings, HBA, and HBD. The data set size is shown in **Table 2**, which

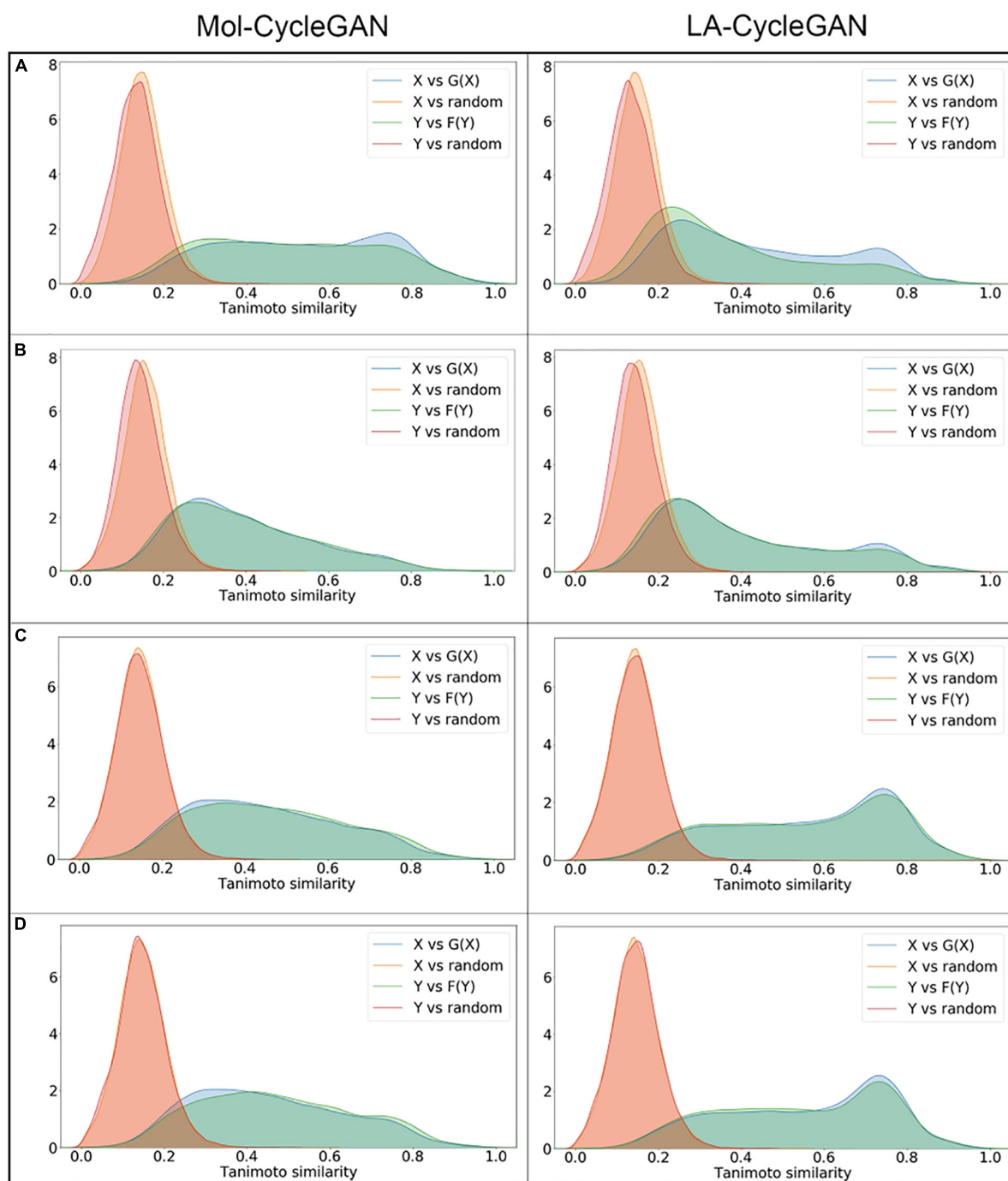


FIGURE 5 | Density map of Tanimoto similarity between corresponding molecules. **(A)** Tanimoto similarity for the Aliphatic Rings subset; **(B)** Tanimoto similarity for the Aromatic Rings subset; **(C)** Tanimoto similarity for the HBA subset; **(D)** Tanimoto similarity for the HBD subset. The left panel is from Mol-CycleGAN and the right panel is from LA-CycleGAN.

TABLE 4 | Performance evaluation of model on generating molecules.

Data	Model	Valid (↑)	Novelty (↑)	IntDiv ₁ (↑)	IntDiv ₂ (↑)	Filters (↑)
Aliphatic	Mol-CycleGAN	0.998	0.975	0.861	0.856	0.588
Rings	LA-CycleGAN	0.998	0.992^a	0.869	0.863	0.644
Aromatic	Mol-CycleGAN	0.996	0.988	0.867	0.861	0.615
Rings	LA-CycleGAN	0.998	0.988	0.868	0.872	0.682
HBA	Mol-CycleGAN	0.997	0.977	0.871	0.865	0.606
	LA-CycleGAN	0.998	0.987	0.883	0.877	0.674
HBD	Mol-CycleGAN	0.996	0.975	0.872	0.866	0.603
	LA-CycleGAN	0.998	0.987	0.883	0.866	0.673

^aThe bolded data in **Table 4** represents the item with the highest data value in the column.

shows the train size and test size of the molecules in the data set. In all experiments, we use the training set (X_{train} and Y_{train}) to train and the test set (X_{test} and Y_{test}) to evaluate the model.

Figure 3 shows the experimental data set ratio distribution map in **Table 2**. The population of molecules with Aliphatic Rings, Aromatic Rings, HBA, and HBD and the distribution of X and Y are shown in **Figure 3**. In each histogram, the blue columns represent the distribution of the ZINC-250K data set according to the four characteristics. The orange bars represent the distribution of the X data set. The green bars represent the distribution of the Y data set.

Structure Attributes

In the process of model training, due to the special structure of CycleGAN, the model will transform and reconstruct the structure of the molecular data on the two opposite regions X and Y. In **Table 3**, the structural performance and structural attributes of the model in the distribution of different characteristics are quantitatively computed, including Aliphatic Rings, Aromatic Rings, HBA, and HBD. A fully symmetrical structure is used in the model. When the model is constructing, we partly consider the problem of structural transformation. LA-CycleGAN model has been improved in terms of Success rate, Uniqueness and Non-identity as shown as the bolded data in **Table 3**. Aromatic Rings has the lowest conversion success rate and is the most

difficult to convert. After optimization, Aliphatic Rings has the highest success rate for substructure conversion tasks. Under the distribution of HBA and HBD characteristics, the two data sets tend to be consistent in the direction of structural transformation. The success rate of HBA and HBD is only slightly different. Uniqueness has improved significantly in all distributions. This result shows that the LA-CycleGAN model can reduce the probability of repeated generation of molecules and avoid an increase in the overlap rate of molecules at the same region. During model verification, the Non-identity of HBA and HBD has been significantly improved after optimization. This result shows that the similarity of molecules has been improved with the conserved Aliphatic Rings. In summary, the LA-CycleGAN model presents the best ability to obtain data conversion over Mol-CycleGAN when training on a data set distributed according to Aliphatic Rings. It also proves that the data set is easy to change. It is easier to use in tests of drug production.

As shown in **Figure 4**, in the distribution of Aliphatic Rings, without Aliphatic Ring molecules, there is only a difference in the number of rings between data sets X and Y. Therefore, the number of Aliphatic Rings in the newly generated molecular structure is significantly reduced, and the success rate of obtaining Aliphatic Ring conversion is higher. In the above-mentioned molecular data conversion processes based on the distribution of the characteristic element, molecules that do not contain the above-mentioned characteristic element are eliminated, which effectively improves the success rate of molecular transformation. At the same time, the number of molecules produced by the 3-ring Aliphatic Ring has increased significantly, and the number of Aliphatic Rings produced by the 1-ring and 2-ring Aliphatic Rings is still the largest as shown in **Figure 4**.

Similarity Evaluation

In the chemical space, the similarity of molecular structure will directly affect the biological activity of similar molecules. In **Figure 5**, the Tanimoto similarity evaluation is performed on the data set of the distribution of various elements. It visually shows the similarity between each compound vector. Compared with Mol-CycleGAN, LA-CycleGAN achieve better Tanimoto

TABLE 5 | Evaluation of the similarity of the generated molecules.

Model test		FCD (↓)		SNN (↑)		Frag (↑)		Scaff (↑)	
Data	Model	Test	TestSF	Test	TestSF	Test	TestSF	Test	TestSF
Aliphatic Rings	Mol-CycleGAN	0.574	0.600	0.498	0.486	0.954	0.475	0.277	0.090
	LA-CycleGAN	0.568	0.600	0.502^a	0.493	0.963	0.664	0.374	0.108
Aromatic Rings	Mol-CycleGAN	0.391	0.047	0.142	0.467	0.142	0.109	0.563	0.157
	LA-CycleGAN	0.361	0.046	0.481	0.471	0.166	0.471	0.577	0.199
HBA	Mol-CycleGAN	0.389	0.442	0.479	0.464	0.331	0.473	0.759	0.134
	LA-CycleGAN	0.382	0.417	0.479	0.468	0.347	0.514	0.799	0.141
HBD	Mol-CycleGAN	0.392	0.444	0.480	0.464	0.318	0.456	0.660	0.137
	LA-CycleGAN	0.384	0.330	0.480	0.468	0.328	0.486	0.689	0.137

^aThe bolded data in **Table 5** represents the best value for the column.

similarity distribution between the generated molecules and the starting molecules. The Tanimoto similarity distribution also holds true for random molecules in the ZINC-250K data set. It can be seen that the molecular similarity of the LA-CycleGAN model is significantly increased. The Attention mechanism assigns different degrees of attention to different parts of the input data or feature map. Focused learning makes the weight distribution obtained in the process of molecule generation more concentrated, and can retain useful information. Finally, it will generate highly similar molecules. The distribution of similarity between the newly generated molecule and the starting molecule in the LA-CycleGAN model gradually tends to be consistent in the X and Y regions. Compared with the original model, the consistency of the similarity distribution of the optimized model has been adjusted and improved. To improve the similarity between two molecules, it is necessary to enhance the discrimination ability of the discriminator. Extract features from the input, and then add a dense layer of one-dimensional output to determine whether the extracted features belong to a specific category. The problem of judging true and false molecules is transformed into a binary classification problem. After convergence, the discriminator is used as a classifier for judging the true and false of molecular data.

Table 4 shows the prediction and verification of the model from the data structure of the four different feature distributions to determine the effectiveness of the model's molecular generation. The Valid of the HBA distribution is the highest in the four different feature distributions. The Valid of the Aliphatic Rings distribution has not been changed. The effectiveness of the distribution of Aromatic Rings has increased the most. Novelty reveals the ability of the optimized model to generate new molecules. Aliphatic Rings generates the highest proportion of new molecules and the highest improvement. The LA-CycleGAN model has the highest IntDiv1 and IntDiv2 scores on the HBA distribution. The HBA molecule has the best performance in terms of generating internal diversity. The internal diversity of HBD distribution structure is slightly lower than that of HBA. It can be seen that the data structure based on the hydrogen bond system is easy to modify and reorganize molecules. Aromatic Rings has the most obvious performance in the Filters evaluation. As part of generating molecules, it can filter charged particles to satisfy the effectiveness of generated molecular compounds. At the same time, inhibiting the generation or deleting molecular fragments do not meet the designed expectations. In our evaluation, we divide the dataset according to the characteristics of the pharmacophore elements without considering the diversity of molecular skeletons. It would be an interesting study to compare the detailed effect of diversity of starting molecules on the diversity of generated molecule by finely tuning the degree of scaffold similarity in two sets X and Y (Benhenda, 2017; Gui et al., 2020).

Table 5 shows the four evaluation indicators of FCD, SNN, Frag, and Scaff for the evaluation of the similarity of generated molecules. The chemical fragments generated by the model design have higher similarity and the optimized

model has improved on these four evaluation indicators. Aliphatic Rings performs well in Fra, Scaff, and SNN. The fragments of the generated molecule and the starting molecule have the highest similarity ratio, and have a high degree of similarity in the direction of the chemical structure, so they

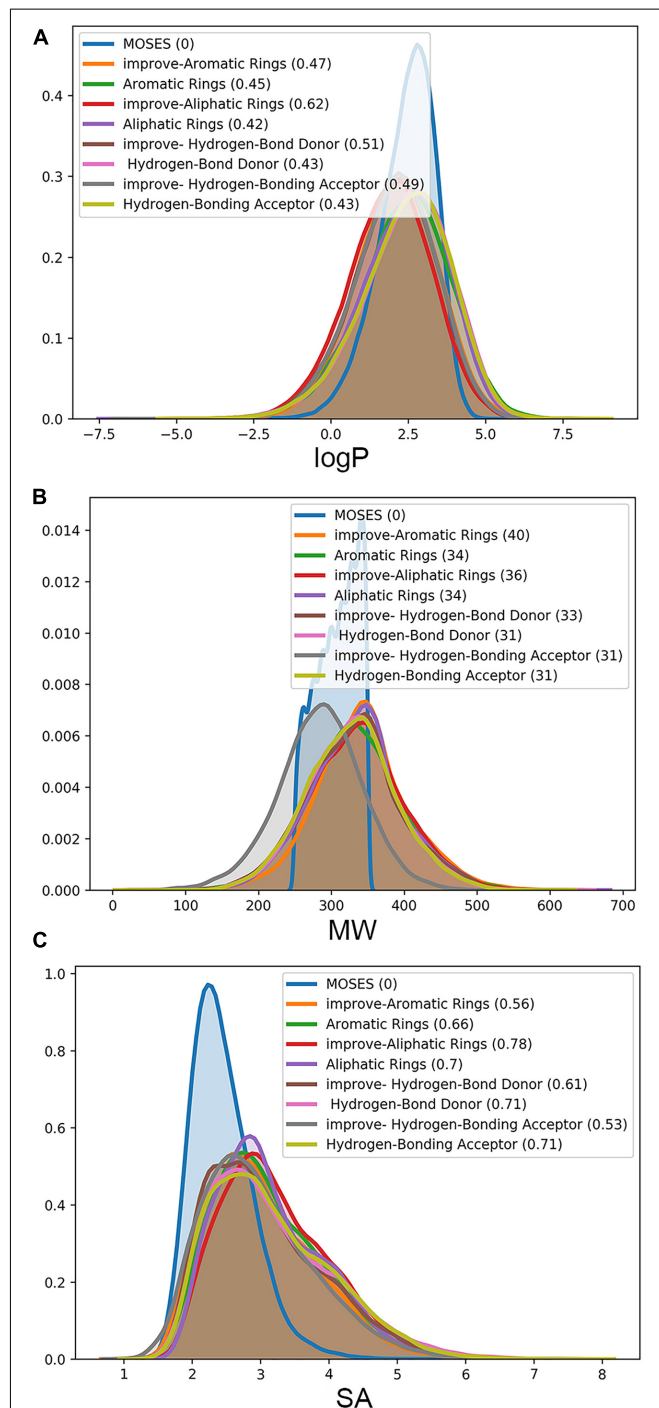


FIGURE 6 | Attribute distribution of generated molecules. LA-CycleGAN model provides (A) the improved logP, (B) molecule weight (MW) and (C) Synthetics Accessibility (SA) Score.

have the same biological response. From an overall point of view, the performance of the LA-CycleGAN model has been comprehensively improved. LSTM solves the problem of gradient disappearance and improves the accuracy of molecule generation. The generated molecular data has a high proportion of scaffolds that are the same as the test set. Among the four data sets distributed according to characteristic elements, HBA has the most significant improvement and the highest proportion. FCD is the error function used to measure the activity of the resulting compound. In the optimized model, the result of Aromatic Rings achieves the smallest error and its biological activity is most similar to the starting molecule. In summary, from the four different chemical structure distributions, the model can generate pharmacophores with excellent biological activity.

We evaluate the generated molecules by using the evaluation indicators provided by MOSES. In **Figure 6**, the Mol-CycleGAN model and the LA-CycleGAN model are compared from LogP, MW, and SA. LogP shows that the concentration ratio of octanol

to water in the LA-CycleGAN model has increased significantly. Compared with Mol-CycleGAN, our LA-CycleGAN model has reproduced the best LogP distribution for the set of Aliphatic Rings, reaching 0.62. The overall Mol-Weight has been improved, and the overall molecular weight has been increased. Based on the molecular weight of the generated and tested sets, it can be judged whether the generated set is biased toward lighter or heavier molecules. In MW plots, the MW of the generation set of the improved model has been increased, which means that the generated set of the new model is biased toward heavier molecules. SA reveals the difficulty of drug synthesis and is used to evaluate learning models. Aliphatic Rings is the most difficult to synthesize drugs. The difficulty of synthesizing has decreased for the other three groups of compounds. For example, the SA decreases from 0.71 to 0.53 for the set of HBA.

The visualization process of the Mol-CycleGAN and LA-CycleGAN model to generate molecules in the four data distributions is shown in **Figures 7, 8**. As shown in molecular

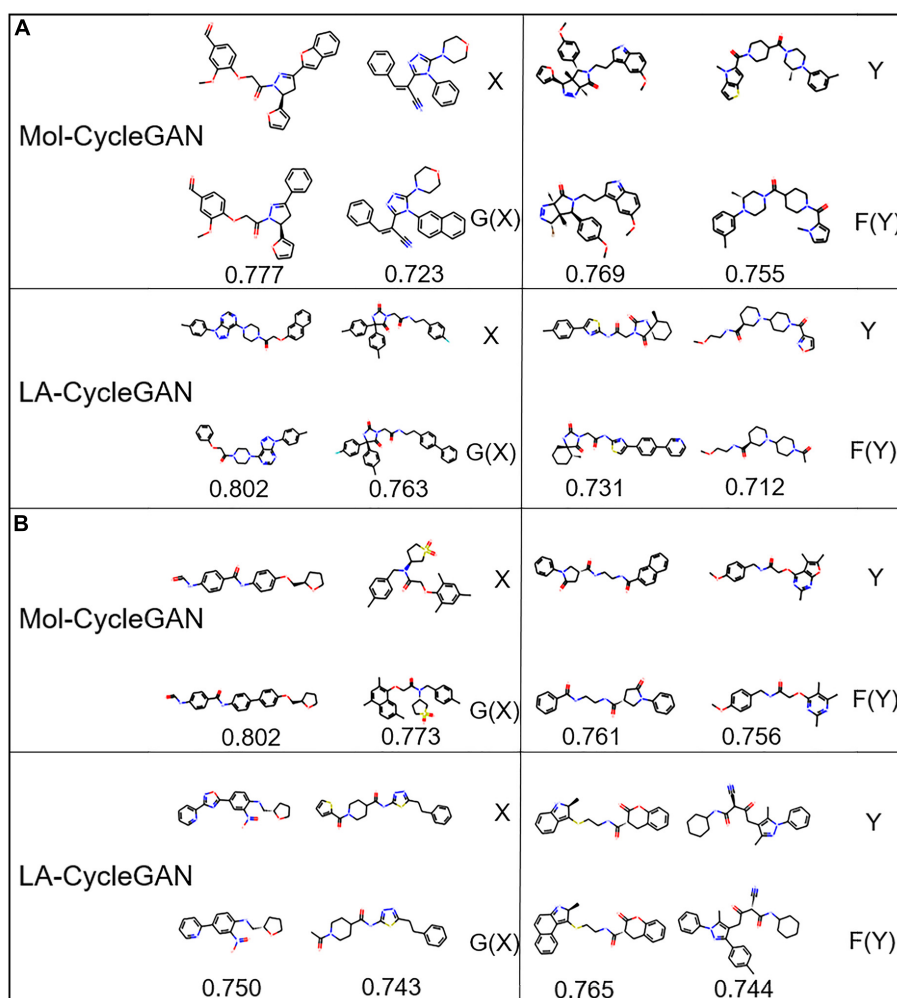


FIGURE 7 | Molecular structure diagram. **(A)** Molecules from the aliphatic ring subset; **(B)** molecules from aromatic ring subset. In each sub-figure, the upper layer shows the starting molecules, the middle layer shows the generated molecules, and the bottom layer shows the similarity score.

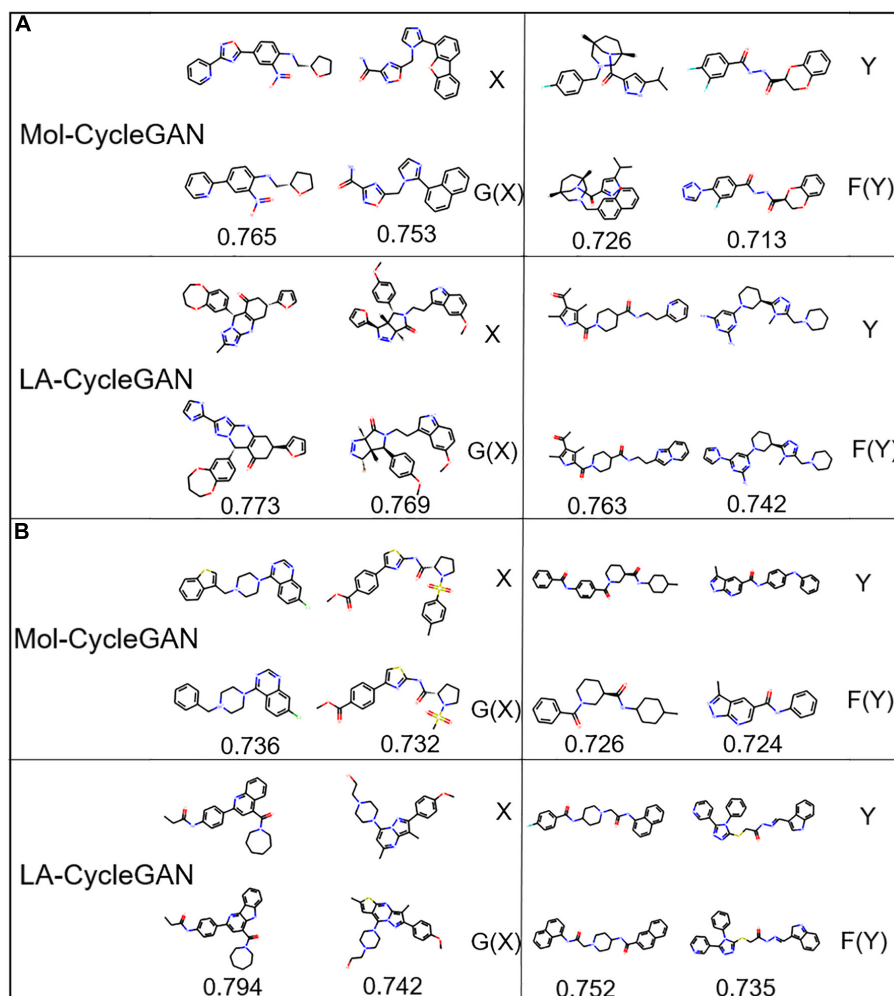


FIGURE 8 | Molecular structure diagram. **(A)** Molecules from the HBA subset; **(B)** molecules from HBD subset. In each sub-figure, the upper layer shows the starting molecules, the middle layer shows the generated molecules, and the bottom layer shows the similarity score.

diagram, the obtained molecules are all effective molecules. For both models, the similarity of X and G(X) is generally higher than that of Y and F(Y). LA-CycleGAN displays comparative performance as the Mol-CycleGAN. Especially in the task of reproducing molecules of HBA and HBD, LA-CycleGAN produces higher similarity score than the Mol-CycleGAN as shown in **Figure 8**. After optimizing the model, HBD obtained the highest similarity score, indicating that the similarity has been improved significantly.

CONCLUSION

We proposed LA-CycleGAN as a new method of molecule generation by embedding LSTM and Attention mechanism in the CycleGAN model architecture. LSTM and Attention mechanisms are introduced to solve the problem of gradient disappearance and improve the accuracy of molecule generation. The generator in the CycleGAN uses Autoencoder to map the molecular

structure into the latent chemical space. The decoder returns the sampled potential chemical space to the original space. Finally, the molecular map is obtained from the potential space of JT-VAE. The proposed model is evaluated by four subsets with different feature distributions extracted from the ZINC-250K dataset. The generated molecules between the Mol-CycleGAN model and the LA-CycleGAN model are quantitatively evaluated. The experimental results show that the similarity and success rate of molecules generated by the LA-CycleGAN model have been significantly improved over Mol-CycleGAN. The generated molecules have similar or even the same biological activity as the starting molecules. LA-CycleGAN model can act as one of molecule generation method to generate molecules with similar drug-like compounds. The attention-based deep neural network can be interpreted by further analyzing the relationship between the attention scores of features and the expected generated molecules. It would be an attractive work to make *de novo* molecular generation model interpretable in future study.

DATA AVAILABILITY STATEMENT

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found below: https://osf.io/42x7d/?view_only=c2cf708215804280bdf24c2bfb4f690b, Open Science Framework.

AUTHOR CONTRIBUTIONS

FW, LiX, and SC derived the concept. FW and XF wrote relevant code and performed the experimental work and analysis. XF, XG, and LeX wrote the original

manuscript. FW, XF, XG, and LeX provided the feedback and critical input. LiX and SC prepared the revised version of the manuscript. All authors read and approved the final manuscript.

FUNDING

This work was supported by the fund of the Natural Science Foundation of Jiangsu Province (BE2019650 and BK20191032), the National Natural Science Foundation of China (22003020 and 81803430), the Changzhou Sci&Tech Program (CJ20200045), and the Jiangsu “Mass Innovation and Entrepreneurship” Talent Programme.

REFERENCES

- Agrawal, P. (2018). Artificial intelligence in drug discovery and development. *J. Pharmacovigil* 6:e173.
- Arús-Pous, J., Blaschke, T., Ulander, S., Reymond, J. L., Chen, H., and Engkvist, O. (2019). Exploring the GDB-13 chemical space using deep generative models. *J. Cheminform* 11, 1–14.
- Benhenda, M. (2017). ChemGAN challenge for drug discovery: can AI reproduce natural chemical diversity? *arXiv [Preprint]*.arXiv:1708.08227.
- Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., et al. (2018). Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Sci.* 4, 268–276. doi: 10.1021/acscentsci.7b00572
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. Cambridge: MIT press.
- Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). Generative adversarial networks. *Adv. Neural Inform. Process. Syst.* 3, 2672–2680.
- Grisoni, F., Moret, M., Lingwood, R., and Schneider, G. (2020). Bidirectional molecule generation with recurrent neural networks. *J. Chem. Inform. Model* 60, 1175–1183. doi: 10.1021/acs.jcim.9b00943
- Gui, J., Sun, Z., Wen, Y., Tao, D., and Ye, J. (2020). A review on generative adversarial networks: algorithms, theory, and applications. *arXiv [Preprint]*.arXiv:2001.06937
- Ioffe, S., and Szegedy, C. (2015). “Batch normalization: accelerating deep network training by reducing internal covariate shift,” in *Proceedings of the 32nd International Conference on Machine Learning*, Vol. 37 (Lille: PMLR), 448–456.
- Jensen, J. H. (2019). A graph-based genetic algorithm and generative model/Monte Carlo tree search for the exploration of chemical space. *Chem. Sci.* 10, 3567–3572. doi: 10.1039/c8sc05372c
- Jin, W., Barzilay, R., and Jaakkola, T. (2018). “Junction tree variational autoencoder for molecular graph generation,” in *Proceedings of the International Conference on Machine Learning*, Vol. 80 (Stockholm: PMLR), 2323–2332.
- Jing, Y., Bian, Y., Hu, Z., Wang, L., and Xie, X. Q. (2018). Deep learning for drug design: an artificial intelligence paradigm for drug discovery in the big data era. *AAPS J.* 20, 1–10. doi: 10.1016/b978-0-12-820045-2.00002-7
- Kadurin, A., Aliper, A., Kazennov, A., Mamoshina, P., Vanhaelen, Q., Khrabrov, K., et al. (2017). The cornucopia of meaningful leads: applying deep adversarial autoencoders for new molecule development in oncology. *Oncotarget* 8:10883. doi: 10.18632/oncotarget.14073
- Lavecchia, A. (2019). Deep learning in drug discovery: opportunities, challenges and future prospects. *Drug Discov. Today* 24, 2017–2032. doi: 10.1016/j.drudis.2019.07.006
- Li, Y., Zhang, L., and Liu, Z. (2018). Multi-objective de novo drug design with conditional graph generative model. *J. Cheminform* 10, 1–24.
- Lim, J., Hwang, S. Y., Moon, S., Kim, S., and Kim, W. Y. (2020). Scaffold-based molecular design with a graph generative model. *Chem. Sci.* 11, 1153–1164. doi: 10.1039/c9sc04503a
- Macalino, S. J., Gosu, V., Hong, S., and Choi, S. (2015). Role of computer-aided drug design in modern drug discovery. *Arch. Pharm. Res.* 38, 1686–1701. doi: 10.1007/s12272-015-0640-5
- Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., and Frey, B. (2015). Adversarial autoencoders. *arXiv [preprint]*.arXiv:1511.05644.
- Mamoshina, P., Vieira, A., Putin, E., and Zhavoronkov, A. (2016). Applications of deep learning in biomedicine. *Mol. Pharm.* 13, 1445–1454. doi: 10.1021/acs.molpharmaceut.5b00982
- Mao, X., Li, Q., Xie, H., Lau, R. Y. K., Wang, Z., and Smolley, S. P. (2017). “Least squares generative adversarial networks,” in *Proceedings of the IEEE International Conference on Computer Vision* (New York, NY:IEEE), 2794–2802.
- Maziarka, Ł., Pocha, A., Kaczmarczyk, J., Rataj, K., Danel, T., and Warchol, M. (2020). Mol-CycleGAN: a generative model for molecular optimization. *J. Cheminform* 12:2.
- Méndez-Lucio, O., Baillif, B., Clevert, D. A., Rouquié, D., and Wichard, J. (2020). De novo generation of hit-like molecules from gene expression signatures using artificial intelligence. *Nat. Commun.* 11:10.
- Muegge, I., Bergner, A., and Kriegl, J. M. (2017). Computer-aided drug design at Boehringer Ingelheim. *J. Comp. Aid. Mol. Design* 31, 275–285. doi: 10.1007/s10822-016-9975-3
- Nastase, A. F., Anand, J. P., Bender, A. M., Montgomery, D., Griggs, N. W., Fernandez, T. J., et al. (2019). Dual pharmacophores explored via structure–activity relationship (SAR) matrix: insights into potent, bifunctional opioid ligand design. *J. Med. Chem.* 62, 4193–4203. doi: 10.1021/acs.jmedchem.9b00378
- Öztürk, H., Ozkirimli, E., and Özgür, A. (2016). A comparative study of SMILES-based compound similarity functions for drug-target interaction prediction. *BMC Bioinformatics* 17:128. doi: 10.1186/s12859-016-0977-x
- Polykovskiy, D., Zhebrak, A., Sanchez-Lengeling, B., Golovanov, S., Tatanov, O., Belyaev, S., et al. (2020). Molecular sets (MOSES): a benchmarking platform for molecular generation models. *Front. Pharmacol.* 11:565644. doi: 10.3389/fphar.2020.565644
- Prykhodko, O., Johansson, S. V., Kotsias, P. C., Bjerrum, E. J., Engkvist, O., and Chen, H. (2019). A de novo molecular generation method using latent vector based generative adversarial network. *J. Cheminform* 11:74.
- Radford, A., Metz, L., and Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv [Preprint]*.arXiv:1511.06434.
- Sarmad, M., Lee, H. J., and Kim, Y. M. (2019). “RL-GAN-NET: a reinforcement learning agent controlled GAN network for real-time point cloud shape completion,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA. 5898–5907.
- Segler, M. H. S., Kogej, T., Tyrchan, C., and Waller, M. P. (2018). Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Central Sci.* 4, 120–131. doi: 10.1021/acscentsci.7b00512
- Simonovsky, M., and Komodakis, N. (2018). “Graphvae: towards generation of small graphs using variational autoencoders,” in *Proceedings of the*

- 27th International Conference on Artificial Neural Networks (Cham:Springer), 412–422. doi: 10.1007/978-3-030-01418-6_41
- Sterling, T., and Irwin, J. J. (2015). ZINC 15—ligand discovery for everyone. *J. Chem. Inf. Model.* 55, 2324–2337. doi: 10.1021/acs.jcim.5b00559
- Tang, B., Pan, Z., Yin, K., and Khateeb, A. (2019). Recent advances of deep learning in bioinformatics and computational biology. *Front. Genet.* 10:214. doi: 10.3389/fgene.2019.00214
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. *arXiv [Preprint]*. arXiv:1706.03762.
- Wainberg, M., Merico, D., Delong, A., and Frey, B. J. (2018). Deep learning in biomedicine. *Nat. Biotechnol.* 36, 829–838.
- Walters, W. P., and Murcko, M. (2020). Assessing the impact of generative AI on medicinal chemistry. *Nat. Biotechnol.* 38, 143–145. doi: 10.1038/s41587-020-0418-2
- Weininger, D. (1988). SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.* 28, 31–36. doi: 10.1021/ci00057a005
- Xu, Y., Lin, K., Wang, S., Wang, L., Cai, C., Song, C., et al. (2019). Deep learning for molecular generation. *Future Med. Chem.* 11, 567–597.
- Yasonik, J. (2020). Multiobjective de novo drug design with recurrent neural networks and nondominated sorting. *J. Cheminf.* 12:14.
- Zhu, J. Y., Park, T., Isola, P., and Efros, A. A. (2017). “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *Proceedings of the IEEE International Conference on Computer Vision*, Venice, 2223–2232.
- Conflict of Interest:** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.
- Publisher’s Note:** All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.
- Copyright © 2021 Wang, Feng, Guo, Xu, Xie and Chang. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.



Locally Adjust Networks Based on Connectivity and Semantic Similarities for Disease Module Detection

Jia Liu¹, Huole Zhu^{2,3} and Jianfeng Qiu^{2,3*}

¹State Key Laboratory of Media Convergence and Communication, Communication University of China, Beijing, China, ²Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, School of Artificial Intelligence, Anhui University, Hefei, China, ³Information Materials and Intelligent Sensing Laboratory of Anhui Province, School of Artificial Intelligence, Anhui University, Hefei, China

OPEN ACCESS

Edited by:

Xiangxiang Zeng,
Hunan University, China

Reviewed by:

Jinting Guan,
Xiamen University, China
Huaming Chen,
University of Adelaide, Australia

*Correspondence:

Jianfeng Qiu
qiu Jianfeng@ahu.edu.cn

Specialty section:

This article was submitted to
Computational Genomics,
a section of the journal
Frontiers in Genetics

Received: 17 June 2021

Accepted: 22 September 2021

Published: 25 October 2021

Citation:

Liu J, Zhu H and Qiu J (2021) Locally
Adjust Networks Based on
Connectivity and Semantic Similarities
for Disease Module Detection.
Front. Genet. 12:726596.
doi: 10.3389/fgene.2021.726596

For studying the pathogenesis of complex diseases, it is important to identify the disease modules in the system level. Since the protein-protein interaction (PPI) networks contain a number of incomplete and incorrect interactome, most existing methods often lead to many disease proteins isolating from disease modules. In this paper, we propose an effective disease module identification method IDMCSS, where the used human PPI networks are obtained by adding some potential missing interactions from existing PPI networks, as well as removing some potential incorrect interactions. In IDMCSS, a network adjustment strategy is developed to add or remove links around disease proteins based on both topological and semantic information. Next, neighboring proteins of disease proteins are prioritized according to a suggested similarity between each of them and disease proteins, and the protein with the largest similarity with disease proteins is added into a candidate disease protein set one by one. The stopping criterion is set to the boundary of the disease proteins. Finally, the connected subnetwork having the largest number of disease proteins is selected as a disease module. Experimental results on asthma demonstrate the effectiveness of the method in comparison to existing algorithms for disease module identification. It is also shown that the proposed IDMCSS can obtain the disease modules having crucial biological processes of asthma and 12 targets for drug intervention can be predicted.

Keywords: complex disease, module identification, protein-protein interaction network, locally adjust networks, connectivity and semantic similarities

1 INTRODUCTION

There exist a number of complex diseases, which are not caused by the malfunction of an individual gene product, but the dysfunction of biological systems formed by several disease-related genes (Zheng et al., 2006; Zheng et al., 2008; Schadt, 2009; Zanzoni et al., 2009; Albert-László et al., 2011; Su et al. 2019). These disease-related genes and their products (e.g., proteins) are not randomly distributed on a molecular network, but they prefer to work together as a group for similar biological functions Sol et al., 2010. The above evidence suggests the existence of disease modules, which were firstly defined by Barabasi et al. as the connected subgraphs formed by proteins associated with a disease (Menche et al., 2015). The disease modules can be considered as the characteristic of a particular disease phenotype (Susan Dina et al., 2015). It becomes quite important to identify the

disease modules, which is helpful for understanding the molecular mechanisms of disease origin and progression, and thus aiding the identification of synergistic drug combinations (Cheng et al., 2019).

With the rapid accumulation of protein-protein interactions, the investigation of interactions between proteins in the human protein-protein interaction (PPI) networks has become one of the primary approaches for detecting disease modules of complex diseases (Igor et al., 2008; Sebastian et al., 2008; Wang et al., 2011). These approaches usually are performed by using the connectivity information in the PPI network, and can be roughly classified into four categories, i.e., neighborhood scoring methods (Krauthammer et al., 2004; Jonsson and Bates, 2006; Tu et al., 2006; Xu and Li, 2006), seed expanding-based methods (Sharma et al., 2012; Susan Dina et al., 2015; Zhang et al., 2017b), diffusion-based methods (Sebastian et al., 2008) and representation learning methods (Härtner et al., 2018). However, the disease modules achieved by these connectivity-based approaches usually show insufficient reliability to illustrate a specific disease phenotype, since nearly 80% of actual associations between proteins are not included in the existing PPI network and these missing associations leave many disease proteins isolated from their disease modules (Menche et al., 2015). Besides, high throughput experiments often produce a large number of interactions with noise, which makes several irrelevant proteins included in the disease module (Cho and Montanez, 2013).

To obtain better detection results, several studies have been performed by combining the protein-protein interaction data with other types of biological data, such as sequence-based features, epigenomic data, gene ontology (GO) annotation and expression patterns (Csaba and Mauno, 2009; Franke et al., 2006b; Liu et al., 2015). Among these biological data, GO annotation has shown to be an effective semantic resource which usually serves as a complement to protein-protein interactions to reflect functional information, where the semantic information of a gene is defined as the molecular function of genes and the biological processes in which the genes are involved (Franke et al., 2006b; Liu et al., 2015). Disease modules achieved by existing approaches have shown the ability to combine the connectivity information with the semantic information for the prioritizing of candidate disease genes (Franke et al., 2006b; Liu et al., 2015). For example, in Franke et al. (2006b), a gene network is developed by the intergration of the GO annotation information, interactions between proteins and microarray coexpressions, and genes are ranked based on the network. In Liu et al. (2015), Liu et al. proposed a method combining the topological similarity in the PPI network with the semantic similarity to select the candidate disease genes. However, the detection results of existing methods need to be further improved, since several unreliable interactions will hinder the detection effectiveness.

Recent studies on complex networks show that an ambiguous community structure can be converted into a structure much clearer than the original one by adding and reducing several links in the network (Su et al., 2021). It is known that about 80% of the

disease proteins are disconnected from disease modules because of the incomplete biological network, where these proteins tend to be localized in the neighborhood of the disease modules (Menche et al., 2015). This means that the implementing of removing associations from the PPI network and adding into associations around the known disease proteins can compensate for the incomplete and incorrect interactions between the proteins in the PPI network, which will facilitate the detection of disease modules. For this reason, we proposed a connectivity and semantic similarities based method (termed as IDMCSS) to identify disease modules by locally adjusting a given PPI network in the detection process in a conference paper (Su et al., 2020). The connectivity similarity reflects the closeness of proteins based on protein-protein interactions and the semantic similarity represents functional similarities of proteins based on GO annotation information. In Su et al. (2020), due to the page limitation, the IDMCSS was only briefly presented and some simple experiments demonstrated the effectiveness of the algorithm for disease module identification. In this paper, we give an extended version of the paper in Su et al. (2020) by adding more analysis and discussions on the algorithm. Specifically, we present a detailed description of the strategies used in the IDMCSS and a series of experimental results are reported with detailed discussions to illustrate the competitiveness of the IDMCSS. We also add the related work section to highlight the difference between the IDMCSS and existing algorithms, as well as the complexity analysis of the IDMCSS. To sum up, the IDMCSS algorithm contains the following two main contributions:

- 1) A strategy of network structure adjustment is proposed to locally change the structure of the existing PPI network by adding several missing links which are likely to be related to disease proteins and removing some existing links which have an extremely weak correlation to disease proteins. To this end, the strong-linked or weak-linked proteins are firstly selected from the neighbors of disease proteins, where the strong-linked proteins and the weak-linked proteins have large and small connective similarities with disease proteins, respectively. Then, two key operators, i.e., adding link operator and removing link operator, are designed to add several links between strong-linked proteins and disease proteins, and remove some links between strong-linked proteins and disease proteins.
- 2) A disease module detection method IDMCSS is proposed by using the strategy of network structure adjustment based on both connective and semantic similarity. In the proposed method, a strategy to expand the set of disease proteins is tailored for the disease module identification. The proposed IDMCSS is verified to be superior over some representative disease module identification approaches.

The rest of the paper is organized as follows. **Section 2** presents the disease module detection problem and reviews the related methods for disease module identification. Then, we describe the details of the proposed algorithm in **Section 3**.

Section 4 shows the experimental results and **Section 5** concludes the paper and gives the future work.

2 RELATED WORK

Recently, the PPI network has become a popular resource for disease module identification (Cagney et al., 2000; Navlakha and Kingsford, 2010). Several disease protein prioritization strategies have been developed to detect disease modules by taking advantage of the existing PPI networks (Agrawal et al., 2017; Cui et al., 2018; Tian et al., 2020). Due to the unreliability of the connective information, there exist some disease modules that are not observable in the PPI networks (Wu et al., 2013). There are also some approaches which are performed by combining connective information and other information such as GO annotation information and expression patterns, to change the structure of the PPI networks (Liu et al., 2015; Franke et al., 2006a; Luo and Liang, 2015; Zhang et al., 2017a). In what follows, we only recall several approaches based on changing network structure, which can be roughly divided into two groups.

The first group changes the network structure by adding several potential missing links to make the network more reliable or adding extra nodes to connect disassociated disease proteins. In order to achieve a reliable network, Franke et al. (2006a) collected a set of validated protein-protein interactions and made use of GO annotation, coexpression data to predict interactions of the remaining protein pairs by a Bayesian classifier. The achieved network was applied to detect candidate disease proteins. To avoid spurious interactions in the PPI networks, a network was reconstructed by connecting pairs of disconnected proteins in the PPI network whose higher-order topological similarities were larger than a certain threshold, where the higher-order topological similarity between two proteins was measured by a link prediction algorithm. Then, candidate inherited disease proteins were prioritized by a random walk-based algorithm on the reconstructed network (Luo and Liang, 2015). Based on a similar idea, Liu et al. developed an algorithm (CTSS) to detect disease proteins by adding the weak interactions between genes which were not connected in the existing network based on the semantic similarity between them (Liu et al., 2015). Experimental results indicated that the PPI network became more perfect by involving reliable associations. In order to connect known disease proteins to be a coherent network module, a seed connector algorithm was developed to detect disease modules by adding as few extra hidden proteins to the set of known proteins as possible (Wang and Loscalzo, 2018). The newly added proteins have been demonstrated useful, since they show significant biological relevance in terms of their functional similarity to known disease proteins and their enrichment of drug targets.

The second group focuses on eliminating potential incorrect associations in the existing networks to achieve a more reliable network or removing several links which are not related to a particular disease phenotype to obtain a disease-specific network. For instance, in order to eliminate potential incorrect associations, the structure of the human PPI network is

adjusted by measuring the correlation coefficient between a pair of connected proteins and removing those with a low correlation coefficient (<0.75) in gene expression data (Liu et al., 2011). In Zhang et al., (2017a), a gene co-expression network was constructed according to the expression patterns of genes, and the links which were not included in the gene co-expression network were removed from the existing PPI network to improve the prediction accuracy of disease proteins. As for a disease-specific network, only the interactions between the immunome proteins in the PPI network were taken into account for the construction of primary immunodeficiencies network, where no new nodes were added, and proteins without interactions were removed (Ortutay and Vihinen, 2008). Similarly, in Bragina et al. (2016), an associative network, which represents molecular interactions between proteins and genes associated with Tuberculosis, was reconstructed and analyzed, and new candidate genes for TB susceptibility were discovered.

Although various network structure based techniques have been developed for the identification of disease modules, traditional approaches are still far from satisfactory, since little approaches focus on dealing with the missing and incorrect links simultaneously. In this paper, we propose a disease module identification method, which is achieved by both adding several potential missing interactions and removing several potential incorrect interactions from the existing PPI networks, based on two types of data, i.e., connective information and semantic information of proteins.

3 THE IDMCSS METHOD

In this section, we give the details of the proposed IDMCSS algorithm. Firstly, the general framework of IDMCSS is presented, and then the network adjustment strategy as well as the way to identify disease proteins which are the main components of IDMCSS are elaborated.

3.1 Framework of IDMCSS

The proposed IDMCSS is a network-based disease module detection method, where the keypoint is to expand a seed module based on an adjusted PPI network. To be specific, let a biological network be G and let the set of known disease proteins be S_0 , the IDMCSS performs seven main steps to detect a disease module. First, we initialize the disease protein set S to be the set of known disease proteins S_0 , and let the candidate disease protein set C be empty. Then, we select all the neighbors of known disease proteins, i.e., $NS = (b_1, \dots, b_\alpha)$, based on the current network G , where b_i ($i = 1, \dots, \alpha$) is a neighbor of a certain node in S . Third, the structure of the current network is locally changed into a new network, G_{new} , by the suggested network adjustment strategy, which focuses on removing the potential incorrect links and adding the potential missing links around the nodes in S . Fourth, the neighbors of the nodes in S , i.e., NS , are updated according to the adjusted network G_{new} . Fifth, we select the protein b from NS which is most likely to be a disease protein by the suggested similarity, and add the node b into the set S and the candidate

disease protein set C . The above the second to the fifth steps are repeated until a certain disease-related information (gene ontology, differential expression genes, pathways) is not significantly enriched in the set C , where the significance estimation used in Wen et al. (2013) is adopted here for enrichment analysis. Sixth, the subnetwork G_s is extracted from the adjusted network G_{new} , where the node set of the subnetwork is S . Note that, G_s may be disconnected. Finally, the connected network with the largest number of nodes in G_s is selected as a disease module, denoted as G_{cs} . **Algorithm 1** presents the pseudo code of the framework of IDMCSS.

Algorithm 1. Framework of the IDMCSS.

Input: A network G , the set of known disease proteins S_0 .

Output: A disease module G_{cs} .

```

1:  $S \leftarrow S_0, C \leftarrow \emptyset$ ;
2: while  $\max(\text{Sig}_{go}(C), \text{Sig}_{de}(C), \text{Sig}_{pa}(C)) < 0.01$  do
3:    $NS \leftarrow \text{disease\_protein\_neighbor}(G, S)$ ;
4:    $G_{new} \leftarrow \text{network\_adjustment}(G, S, NS)$ ;
5:    $NS \leftarrow \text{disease\_protein\_neighbor}(G_{new}, S)$ ;
6:    $(S, C) \leftarrow \text{disease\_protein\_identification}(NS, S, C)$ ;
7: end while
8:  $G_s \leftarrow \text{subnetwork\_selection}(G_{new}, S)$ ;
9:  $G_{cs} \leftarrow \text{disease\_module\_selection}(G_s)$ .
```

3.2 Network Adjustment Strategy

For the network $G = (V, E)$ and the disease protein set S , the IDMCSS starts to locally change the network structure of the original network G around the nodes in S , in order to discard several potential incorrect links and retrieve several missing links in G . To this end, a network adjustment strategy is developed to focus on removing several potential incorrect links associated to the nodes in S and adding potential missing links between a node S and its neighbors. **Algorithm 2** details the procedure of network adjustment strategy, which is performed as follows.

Algorithm 2. Network-adjustment (G, S, NS).

Input: A network $G = (V, E)$, a disease protein set S and a set of disease proteins' neighbors NS .

Output: A network $G_{new} = (V, E')$.

```

1:  $(CS, SS) \leftarrow \text{cal\_connective\_semantic}(NS)$ ;
2:  $(SN, WN) \leftarrow \text{select\_strong\_weak\_nodes}(CS)$ ;
3: for each  $p' \in SN$  do
4:    $S'_1 \leftarrow \text{select\_connected\_nodes}(S, p')$ ;
5:    $S'_2 \leftarrow \text{select\_unconnected\_nodes}(S, p')$ ;
6:    $\varphi_1 \leftarrow \text{mean}(SS, S'_1)$ ;
7:   for each  $p_{i_c} \in S'_2$  do
8:     if  $ss(p', p_{i_c}) > \varphi_1$  then
9:        $E' \leftarrow E \cup \{e_{p'p_{i_c}}\}$ ;
10:    end if
11:  end for
12: end for
13: for each  $p'' \in WN$  do
14:    $S''_1 \leftarrow \text{select\_unconnected\_nodes}(S, p'')$ ;
15:    $S''_2 \leftarrow \text{select\_connected\_nodes}(S, p'')$ ;
16:    $\varphi_2 \leftarrow \text{mean}(SS, S''_1)$ ;
17:   for each  $p_{j_c} \in S''_2$  do
18:     if  $ss(p'', p_{j_c}) < \varphi_2$  then
19:        $E' \leftarrow E / \{e_{p''p_{j_c}}\}$ ;
20:    end if
21:  end for
22: end for
```

First, we calculate both the connective similarity and the semantic similarity between each protein in NS and the

diseases proteins in $S = (p_1, \dots, p_n)$. For a node $b \in NS$, it is supposed that the node b has the degree k and connects to k_s nodes in S . The connective similarity between node b and the nodes in S is calculated by a hypergeometric test as **Eq. 1**, which represents how closely protein b connects to disease proteins in S (Susan Dina et al., 2015).

$$cs(b, S) = 1 - \sum_{t=k_s}^k \frac{C_n^t C_{N-n}^{k-t}}{C_N^k}, \quad (1)$$

where n is the number of nodes in S , and N is the number of nodes in G .

Then, we can calculate the semantic similarity between protein b and disease proteins S . Assume that the set $T = \{t_i | i = 1, \dots, M\}$ consists of all of the terms annotating N proteins in network G .

$$ss(b, S) = \sum_{i=1}^n \frac{\sum_{t_i \in (A_b \cap A_{p_i})} I(t_i)}{I_{\max}(S)}, \quad (2)$$

where $A_b = \{t_{x_k} | k = 1, \dots, m\}$ and $A_{p_i} = \{t_{y_j} | j = 1, \dots, m'\}$ are the sets of terms used to annotate the proteins b and p_i , and t_i represents a term in T . $I(t_i) = -\log[\text{pro}(t_i)]$ is the information of the term t_i , where $\text{pro}(t_i)$ denotes the probability of the presence of the term t_i and its descendants in the term set T . The information of protein p is $I(p) = \sum_{k=1}^m I(t_{x_k})$. $I_{\max}(S) = \max[I(p_1), \dots, I(p_n)]$ denotes the largest value of the information of proteins in S .

Second, the strong-linked nodes (SN) and the weak-linked nodes (WN) are selected from NS , denoting proteins in NS closely and weakly related with disease proteins in S , where a strong-linked node is defined as the protein having a connective similarity with S larger than 0.99, and a weak-linked node is defined as the protein when it has a connective similarity with S smaller than the average value in NS . Note that, the connective similarity ranges from 0 to 1, and the average value of connective similarity is always smaller than 0.99. Thus, there is no intersection between the strong-linked nodes (SN) and the weak-linked nodes (WN). Third, the network G is changed to G_{new} by adding or removing several links associated with the strong-linked or weak-linked nodes, according to the suggested network adjustment strategy. The network adjustment strategy includes two key operators, i.e., adding and removing links, which are designed as follows.

1) Adding link operator: For a strong-linked node $p' \in SN$, we check whether a link needs to be added between p' and the node in S which is not connected with p' in the current network. Let $S'_1 = \{p_{i_1}, \dots, p_{i_r}\} \subseteq S$ and $S'_2 = S/S'_1$ be the two sets of nodes which are connected and not connected to node p' . For each node $p_{i_c} \in S'_2$, a link between node p' and node p_{i_c} is added into the current network when $ss(p', p_{i_c}) > \varphi_1$. This means that a link is added if the semantic similarity $ss(p', p_{i_c})$ between p' and node p_{i_c} is larger than φ_1 , where φ_1 is the mean semantic similarity between p' and each node in S'_1 .

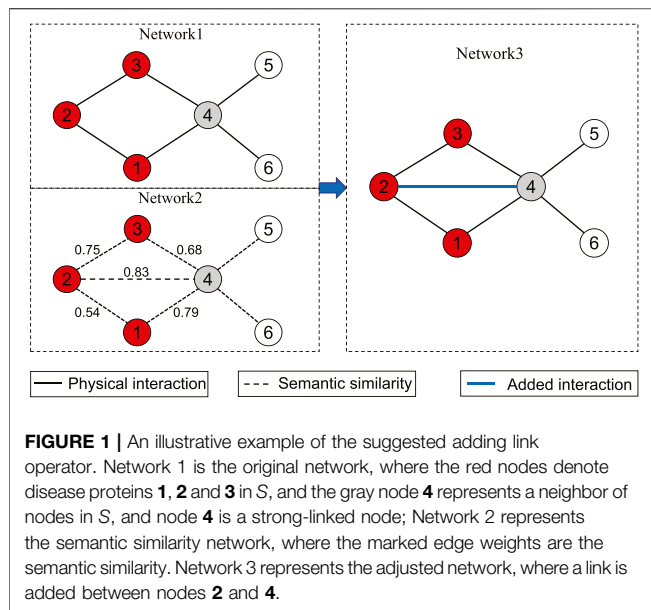
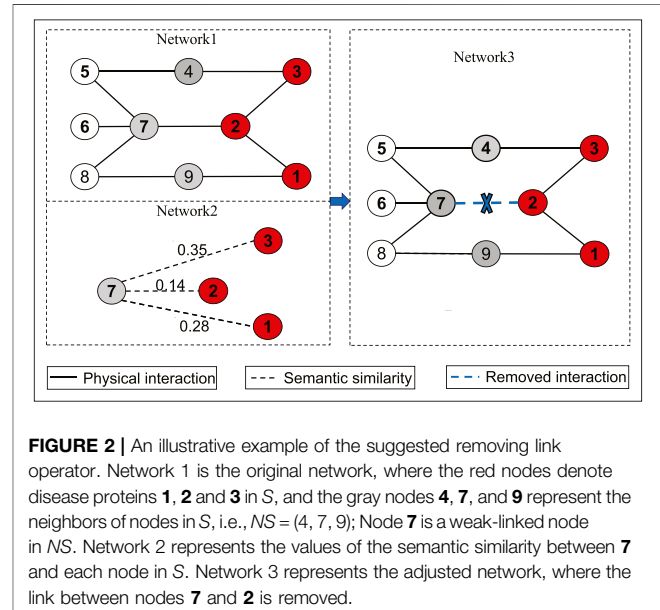


Figure 1 presents an example to show how the suggested adding link operator works. As shown in this figure, the set of disease proteins S contains three nodes 1, 2 and 3, and $NS = \{4\}$. For node 4, $S'_1 = \{1, 3\}$ includes the nodes in S which are connected with 4, and $S'_2 = \{2\}$ contains node 2 which is not connected with node 4. Node 4 is a strong-linked node in NS , since the connective similarity between node 4 and $S = \{1, 2, 3\}$ is 0.9964 according to Eq. 1, which is larger than the threshold 0.99. Further, the link between node 2 and node 4 is added, since the semantic similarity between them is 0.83 which is larger than the threshold $\varphi_1 = \frac{0.68+0.79}{2}$.

2) Removing link operator: For a weak-linked node $p'' \in WN$, the network adjustment strategy checks whether some links deserve to be removed to ensure that the weak-linked node p'' is not connected to any node in S . Let $S'_1 = \{p_{j_1}, \dots, p_{j_s}\} \subseteq S$ and $S'_2 = S/S'_1$ be the two sets of nodes which are not connected and connected to node p'' . For each node $p_{j_c} \in S'_2$, a link between p'' and p_{j_c} is removed when the semantic similarity between p'' and p_{j_c} is smaller than φ_2 , where φ_2 denotes the mean semantic similarity between node p'' and each node in S'_1 .

Figure 2 presents an illustrative example of the removing link operator. In this example, $S = \{1, 2, 3\}$ represents the set of disease proteins and $NS = \{4, 7, 9\}$ consists of all neighbors of nodes in S . For node 7, there are two nodes 1 and 3 which are not connected with it ($S'_1 = \{1, 3\}$), and one node 2 which is connected with it ($S'_2 = \{2\}$). By simple calculation, we can obtain that the connective similarity between node 7 and set S is 0.9964 and the average connective similarity of the nodes in NS is 0.9984. Since the connective similarity is smaller than the average value, the node 7 is weak-linked. Hence, we need to remove the link between nodes 7 and 2 from the network, due to the fact that the threshold $\frac{0.35+0.28}{2}$ is larger than the semantic similarity between nodes 7 and 2 in S'_2 (i.e., 0.14).



3.3 The Similarity Between a Protein and Disease Proteins

In the IDMCSS, the protein having the largest similarity with the nodes in S is selected as a disease protein, where the similarity is measured based on both connective similarity and semantic similarity. Specifically, considering a protein p and a set of disease proteins $S = (p_1, \dots, p_l)$, the similarity between the protein p and the set of disease proteins S , denoted as $sv(p, S)$, is the normalization of the sum of the connective similarity and the semantic similarity, which is defined as Eq. 3.

$$sv(p, S) = \frac{cs(p, S) + ss(p, S)}{2}, \quad (3)$$

where $cs(p, S)$ represents the connective similarity between p and S , and $ss(p, S)$ represents the semantic similarity between p and S .

3.4 Complexity Analysis

Here, an upper bound of the time complexity of the IDMCSS is presented. As described above, the main complexity of IDMCSS lies in the following five steps: 1) the identification of NS , 2) the network adjustment, 3) the selection of disease protein, 4) extracting the subnetwork G_s from the adjusted network, 5) selecting a disease module G_{cs} . Note that, the first three steps are in a while loop.

The complexity for the identification of NS is $O(d_{max} \times n)$, where $|S| = n$, the largest degree of nodes in S is d_{max} . Suppose the number of nodes in NS is n' , a complexity of $O(4 \times n' + n'^2)$ is needed for the network adjustment, since the complexity for calculating connective and semantic similarity as well as selecting strong and weak nodes is $O(4 \times n')$, and the maximum complexity for adding and removing links is $O(n'^2)$. The maximum complexity for the selection of disease protein is $O(n')$. The first three steps holds a time complexity of $O(d_{max} \times n + n'^2)$, since $O(d_{max} \times n + n'^2) \approx O(d_{max} \times n + 4 \times n' + n'^2 + n')$. After the iteration of $maxgen$ times, it needs a complexity of O

$((d_{max} \times n + n'^2) \times maxgen)$ for identifying the disease proteins. The fourth step needs a time complexity of $O(M)$ to extract the subnetwork G_s from the adjusted network G_{new} , where M is the number of links in G_{new} . Finally, it holds a time complexity of $O(M')$ to select a disease module, where M' is the number of links in G_s . Therefore, the IDMCSS holds a computational complexity of $O(d_{max} \times n \times maxgen + n'^2 \times maxgen + M)$, since $O((d_{max} \times n + n'^2) \times maxgen + M + M') \approx O(d_{max} \times n \times maxgen + n'^2 \times maxgen + M)$.

4 EXPERIMENTAL RESULTS

In this section, we first analyze the module of asthma obtained by the proposed IDMCSS, and then compare the performance of the IDMCSS with that of four existing algorithms for disease module detection.

4.1 Datasets

The IDMCSS performs the detection of asthma-related modules based on the protein-protein interaction network. The stopping criterion of the algorithm is set according to the information of gene ontology, differential expression genes and pathways which are related to the asthma. Specifically, the protein-protein interactions, microarray expression data, asthma-related genes and pathways are presented as follows.

First, the protein-protein interaction network is obtained by considering seven kinds of physical interactions simultaneously, which yields a network having 13,460 proteins and 141,296 physical interactions. The seven physical interactions considered here are regulatory interactions (Matys et al., 2003), biophysical interactions Aranda et al. (2009), Ceol et al. (2007), literature curated interactions Prasad et al. (2009), metabolic enzyme-coupled interactions Lee et al. (2008), protein complexes Ruepp et al. (2010), kinase network Hornbeck et al. (2012) and signaling interactions Vinayagam et al. (2011) in human interactome. From the gene ontology annotation database (GOA) Huntley et al. (2015), we extract 19,707 genes annotated with GO terms and hence the obtained network consists of 12,562 proteins and 130,390 physical interactions.

Next, we adopt nine asthma-related microarray expression data sets consisting of the gene expression values for the differential expression analysis. The nine data sets are GSE470, GSE2125, GSE3004, GSE4302, GSE16032, GSE31773, GSE35571, GSE41649 and GSE43696, which can be available from the NCBI Gene Expression Omnibus database (GEO)¹. It is worth noting that we use 107 known asthma-related genes in the protein-protein interaction network for experimental analysis in this paper, which are compiled from pervious literature Vercelli (2008) and several datasets². In addition, 23 asthma-related pathways collected from the literature (Song and Lee, 2013; Sharma et al., 2012) are used in this paper (Supplementary Appendix S1).

¹<http://www.ncbi.nlm.nih.gov/geo/>

²<http://gene2mesh.ncibi.org>

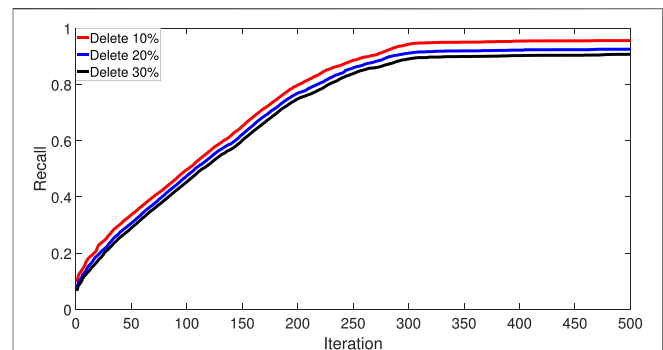


FIGURE 3 | The recall rate of disease module.

4.2 Identification of Disease Modules

We use the IDMCSS to identify disease modules based on an adjusted network, where the final disease module of asthma is achieved by running the proposed IDMCSS 217 iterations. The reason for the iterations for 217 times is that “differential expression genes” is not significantly enriched in current disease proteins earlier than “GO annotation information” and “pathway information”, and the enrichment of the differential expression genes included in the disease proteins is smaller than 0.05 when the algorithm iterates 218 times.

For the disease module of asthma obtained by the suggested IDMCSS, it consists of 279 nodes and 2,819 links. Among the 279 nodes, 62 nodes are known asthma-related proteins and the other 217 nodes are newly discovered relating to asthma-related proteins. In the 2,819 links found in the disease module, 489 links are newly added and 19 links are removed from the original network by the proposed IDMCSS. It is worth noting that some known disease proteins associated with asthma are not included in the obtained disease module of asthma and hence they may be included in other connected subgraphs.

Finally, we take a close look at the closeness of the obtained disease module. We here use the ratio of the number of inner-links to that of external-links as the closeness of the disease module. The module has 2,819 inner-links and 47,657 external-links, and thus the closeness of the disease module is 0.0592. This confirms that the disease module is not a locally dense community as stated by Susan Dina et al. (2015). It can also be found that the obtained disease module has statistically larger closeness than the subnetworks randomly selected from the adjusted protein-protein interaction network according to the Student’s t-test.

4.3 Asthma-Related Pathways and Genes in the Disease Module

In this subsection, we analyze the asthma-related pathways and genes in the disease module. To this end, from 304 human pathways in the Biocarta database given in Supplementary Appendix S2, we extract the 72 candidate pathways which has at least half of genes in the disease module obtained by the algorithm. It can be found that the 72 pathways are possible asthma-related pathways as shown in Supplementary Appendix

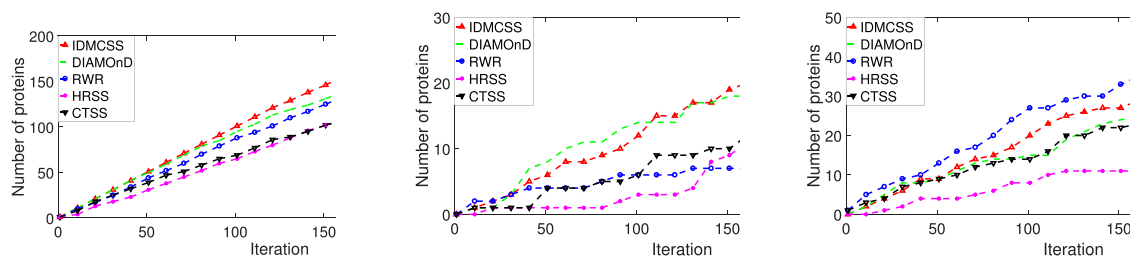


FIGURE 4 | Performance of IDMCSS, CTSS, DIAMOnD, RWR and HRSS on the asthma dataset. GO annotations: the number of asthma-related GO annotations enriched in the disease module; DifferExpre: the number of differential expressed genes in the disease module; Pathways: the number of asthma-related pathways enriched in the disease module.

S3, since they are statistically significantly enriched in the disease module. Among the 72 pathways, two are included in the 23 known asthma-related pathways and the rest 70 are the newly asthma-related pathways predicted by the algorithm. For the 70 pathways, five pathways, “h-il7Pathway”, “h-pkcPathway”, “h-melanocytepathway”, “h-ngfPathway”, and “h-trkaPathway”, are considered to be associated with asthma in previous literature (Kelly and et al., 2009; Hou and et al., 2017; Raap et al., 2003; Abram, 2008).

Next, we will predict several targets of glucocorticoid based on the disease module of asthma, since they are an effective anti-inflammatory drug for asthma. The genes will be considered as the targets of glucocorticoid in asthma if they are differentially expressed between asthmatic fibroblasts untreated and asthmatic fibroblast cells treated with glucocorticoid, but not between normal untreated fibroblast cells and normal fibroblasts treated with glucocorticoid. For this reason, in this paper the 12 genes, *acvr11*, *ar*, *cdk1*, *ctgf*, *ddit3*, *icam1*, *jak1*, *rora*, *smad1*, *snca*, *tgfb2*, and *tlr4*, are considered to be targets of glucocorticoid. To verify the effectiveness of the targets, we use the enrichment analysis of the differential expression genes before and after the treatment of glucocorticoid. For 217 expanded proteins, 23 and 17 expanded proteins are differentially expressed in normal and asthmatic samples, respectively. As for the 62 known asthma-related proteins, 10 and 8 known asthma-related proteins are differentially expressed in normal and asthmatic samples, respectively. Based on the Fisher's exact test, in normal and asthmatic samples the expanded proteins have the enrichment of differential expression genes 6.0324×10^{-4} and 2.70×10^{-3} , and the known asthma-related proteins have the enrichment of differential expression genes 4.32×10^{-2} and 4.30×10^{-2} . This means that the expanded proteins has significantly higher enrichment of differential expression genes than the known asthma-related proteins. Thus, we can conclude that the algorithm can provide effective targets for therapeutic intervention.

4.4 Robustness of IDMCSS

To show the robustness of IDMCSS, Figure 3 gives the recall rate of the disease module when 10, 20, and 30% of the known asthma disease genes are randomly deleted, averaging over 30 times experiments (Warren et al., 2002). It can be found that the removal of the known disease genes has little influence on the performance of the suggested IDMCSS, and it always detect

similar disease modules in the 217 iterations. Hence, we can conclude that the suggested IDMCSS shows a good robustness in detecting disease modules of asthma.

4.5 Performance Comparison

The IDMCSS is compared to four state-of-the-art disease module identification approaches, including a network structure change-based algorithm (CTSS) (Liu et al., 2015) and three traditional approaches without changing network structures (DIAMOnD Susan Dina et al. (2015), RWR Sebastian et al. (2008) and HRSS Wu et al. (2013)), where DIAMOnD and RWR are connective-based algorithms and HRSS is a semantic-based algorithm. Specifically, CTSS identifies disease genes by adding weak interactions between unconnected genes in the existing network based on the semantic similarity between them. The DIAMOnD algorithm is a seed-expanding method which identifies a disease module around a set of known disease proteins in the PPI network. RWR uses random walk analysis, which is a global network distance measure, to measure similarities among proteins in the PPI network. HRSS ranks all nodes by calculating the relative specificity similarity of each node in the network to known disease nodes, where the relative specificity similarity is calculated by taking the global position of relevant gene ontology terms into account. For the above comparison algorithms, the best parameters recommended in their original references are adopted.

Figure 4 presents performance (the number of proteins annotated by asthma-related GO terms, the number of differential expression genes, and the number of proteins in asthma-related pathways) obtained by five approaches on the asthma dataset. To be specific, the left one in Figure 4 draws the number of proteins which are significantly annotated by 940 asthma-related GO terms for different iterations, where the 940 asthma-related GO terms are those enriched in the 107 known asthma proteins (Supplementary Appendix S4). From the figure, it can be found that IDMCSS achieves the largest number of proteins annotated by asthma-related GO terms.

The middle one in Figure 4 plots the number of differential expression genes included in the disease module achieved by IDMCSS and those by four compared algorithms when the iteration ranges from 1 to 217. As can be seen from the figure, the algorithm IDMCSS gains the largest number of differential

expression genes when the iteration is larger than 111. The main reason may be attributed to the fact that by enhancing the structure of PPI, it becomes relatively easy to detect the differential expression genes, thus the IDMCSS can achieve a competitive performance in detecting disease modules. The right one in **Figure 4** presents the number of proteins which belong to the 23 known asthma-related pathways. It is found that the IDMCSS is slightly worse than RWR, but it is better than other algorithms. The main reason for the phenomenon is that the proteins linked by physical interactions tend to collaborate with each other in the same pathway (Venkatesan et al., 2008). The proteins obtained by RWR are always the known disease proteins' neighbors which are connected to the known disease proteins by physical interactions in the PPI network, while those obtained by IDMCSS may be the nodes which are not linked with the known disease proteins. Therefore, we can conclude that the IDMCSS is a competitive disease module detection algorithm in terms of detection quality.

5 CONCLUSION AND FUTURE WORK

³In this paper, we have developed a disease module identification method IDMCSS by modifying the existing PPI networks. In the suggested IDMCSS, some potential interactions are added in the existing PPI network and some incorrect interactions are removed based on the connective and semantic similarities between the given disease proteins and their neighboring proteins. The basic idea of modifying the existing PPI network is that the incorrect links and the missing links are in the original PPI network, and we want to eliminate interference of the incorrect links and missing links for detecting disease module. However, due to the lack of the knowledge about the accurate protein-protein interactions, it is hard to analyze the validity of the modified PPI network, which may be verified in the future. The protein having the best connective and semantic similarities in the neighborhood of known disease proteins is extended into the set of disease proteins on the adjusted PPI network step by step until a stopping criterion is reached. Further, the connected subgraphs which include the disease proteins, as well as the interactions between them, are extracted from the adjusted network. Finally, the connected subgraph which contains the largest number of disease proteins is selected as a disease module.

We have performed a series of experiments on a particular disease, i.e., asthma to show the effectiveness of the IDMCSS. First, the disease module detected by the IDMCSS was not a dense community which is in accordance with traditionary discovery, and it was also significantly different from the random subgraphs. Then, several pathways and genes discovered in the disease

module have been verified to be related to asthma. Further, IDMCSS has little sensitivity to the number of known disease proteins. Finally, IDMCSS was superior to state-of-the-art approaches for disease module identification, since the disease module achieved by IDMCSS includes more proteins which are enriched in asthma-related GO terms, pathways and differential expression genes than those achieved by other approaches. From the above, the experiments have extensively demonstrated the superiority of IDMCSS in disease module identification.

In this work, we have locally adjusted the network structure by the suggested network adjustment strategy to deal with the PPI network which suffers from both high false positive and false negative rates. The IDMCSS performs based on the assumption that the detection results will become better if the PPI network becomes more perfect. Future attention can be given to combing connective information with other kinds of information, such as pathway information and phenotypic similarity information, to further improve the IDMCSS.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found here: In this paper, we adopt nine asthma-related microarray expression data sets consisting of the gene expression 252 values for the differential expression analysis. The nine data sets are GSE470, GSE2125, GSE3004, GSE4302, 253 GSE16032, GSE31773, GSE35571, GSE41649 and GSE43696, which can be available from the NCBI Gene Expression Omnibus database (GEO) <http://www.ncbi.nlm.nih.gov/geo/>.

AUTHOR CONTRIBUTIONS

JL: Software, Original draft preparation HZ: Data process, Experiments JQ: Methodology, Investigation, Reviewing.

FUNDING

The National Natural Science Foundation of China (Grant No. U1804262, 61822301, 61976001, and 61876184), the Ministry of Science and Technology of China Key Project of Science and Technology Innovation 2030 (Grant No. 2018AAA0101302 and 2018AAA0100105), and the Anhui Provincial Key Research and Development Plan (Grand No. 202004j07020005), the Fundamental Research Funds for the Central Universities (CUC210B001). Key Program of Natural Science Project of educational Commission of Anhui Province (KJ2019A0029), the Natural Science Foundation of Anhui Province (2008085QF294, 1908085MF218).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/fgene.2021.726596/full#supplementary-material>

³This paper is an extended version of a paper of our published in the 14th International Conference on Bio-inspired Computing: Theories and Applications (BIC-TA 2019).

REFERENCES

- Abram, M. (2008). Ngf increases cell viability of isolated plasma cells from inflamed airways via trka signalling in a mouse model of allergic asthma. *J. Allergy Clin. Immunol.* 121, S200. doi:10.1016/j.jaci.2007.12.745
- Agrawal, M., Zitnik, M., and Leskovec, J. (2017). Large-scale analysis of disease pathways in the human interactome. *Pac. Symp. Biocomputing* 23, 111–122. doi:10.1142/9789813235533_0011
- Albert-László, B., Natali, G., and Joseph, L. (2011). Network medicine: a network-based approach to human disease. *Nat. Rev. Genet.* 12, 56–68.
- Aranda, B., Achuthan, P., Alam-Faruque, Y., Armean, I., Bridge, A., Derow, C., et al. (2009). The IntAct molecular interaction database in 2010. *Nucleic Acids Res.* 38, D525–D531. doi:10.1093/nar/gkp878
- Bragina, E. Y., Tiys, E. S., Rudko, A. A., Ivanisenko, V. A., and Freidin, M. B. (2016). Novel tuberculosis susceptibility candidate genes revealed by the reconstruction and analysis of associative networks. *Infect. Genet. Evol.* 46, 118–123. doi:10.1016/j.meegid.2016.10.030
- Cagney, G., Uetz, P., and Fields, S. (2000). [1] High-throughput screening for protein-protein interactions using two-hybrid assay. *Methods Enzymol.* 328, 3–14. doi:10.1016/S0076-6879(00)28386-9
- Ceol, A., Aryamontri, A. C., Licata, L., Peluso, D., Briganti, L., Perfetto, L., et al. (2007). Mint, the molecular interaction database: 2009 update. *Nucleic Acids Res.* 35, 572–574. doi:10.1093/nar/gkl961
- Cheng, F., Lu, W., Liu, C., Fang, J., Hou, Y., Handy, D. E., et al. (2019). A genome-wide positioning systems network algorithm for in silico drug repurposing. *Nat. Commun.* 10, 3476. doi:10.1038/s41467-019-10744-6
- Cho, Y., and Montanez, G. (2013). Predicting false positives of protein-protein interaction data by semantic similarity measures. *Curr. Bioinformatics* 8, 339–346.
- Csaba, O., and Mauno, V. (2009). Identification of candidate disease genes by integrating gene ontologies and protein-interaction networks: case study of primary immunodeficiencies. *Nucleic Acids Res.* 37, 622–628.
- Cui, Y., Cai, M., and Stanley, H. E. (2018). Discovering disease-associated genes in weighted protein-protein interaction networks. *Physica A: Stat. Mech. its Appl.* 496, 53–61. doi:10.1016/j.physa.2017.12.080
- del Sol, A., Balling, R., Hood, L., and Galas, D. (2010). Diseases as network perturbations. *Curr. Opin. Biotechnol.* 21, 566–571. doi:10.1016/j.copbio.2010.07.010
- Franke, L., Bakel, H. V., Fokkens, L., de Jong, E. D., Egmont-Petersen, M., and Wijmenga, C. (2006a). Reconstruction of a functional human gene network, with an application for prioritizing positional candidate genes. *Am. J. Hum. Genet.* 78, 1011–1025. doi:10.1086/504300
- Franke, L., Bakel, H. v., Fokkens, L., de Jong, E. D., Egmont-Petersen, M., and Wijmenga, C. (2006b). Reconstruction of a functional human gene network, with an application for prioritizing positional candidate genes. *Am. J. Hum. Genet.* 78, 1011–1025. doi:10.1086/504300
- Härtner, F., Andrade-Navarro, M. A., and Alanis-Lobato, G. (2018). Geometric characterisation of disease modules. *Appl. Netw. Sci.* 3, 10. doi:10.1007/s41109-018-0066-3
- Hornbeck, P. V., Kornhauser, J. M., Tkachev, S., Zhang, B., Skrzypek, E., Murray, B., et al. (2012). Phosphositeplus: a comprehensive resource for investigating the structure and function of experimentally determined post-translational modifications in man and mouse. *Nucleic Acids Res.* 40, D261–D270. doi:10.1093/nar/gkr1122
- Hou, L., Zhu, L., Zhang, M., Zhang, X., Zhang, G., Liu, Z., et al. (2017). Participation of antidiuretic hormone (adh) in asthma exacerbations induced by psychological stress via pka/pkc signal pathway in airway-related vagal preganglionic neurons (avpns). *Cell Physiol Biochem.* 41, 2230–2241. doi:10.1159/000475638
- Huntley, R. P., Sawford, T., Mutowo-Meulenet, P., Shypitsyna, A., Bonilla, C., Martin, M. J., et al. (2015). The goa database: gene ontology annotation updates for 2015. *Nucleic Acids Res.* 43, D1057–D1063. doi:10.1093/nar/gku1113
- Igor, F., Andrey, R., and Dennis, V. (2008). Network properties of genes harboring inherited disease mutations. *Proc. Natl. Acad. Sci. United States America* 105, 4323–4328.
- Jonsson, P. F., and Bates, P. A. (2006). Global topological features of cancer proteins in the human interactome. *Bioinformatics* 22, 2291–2297. doi:10.1093/bioinformatics/btl390
- Kelly, E. A. B., Koziol-White, C. J., Clay, K. J., Liu, L. Y., Bates, M. E., Bertics, P. J., et al. (2009). Potential contribution of il-7 to allergen-induced eosinophilic airway inflammation in asthma. *J. Immunol.* 182, 1404–1410. doi:10.4049/jimmunol.182.3.1404
- Keshava Prasad, T. S., Goel, R., Kandasamy, K., Keerthikumar, S., Kumar, S., Mathivanan, S., et al. (2009). Human Protein Reference Database--2009 update. *Nucleic Acids Res.* 37, D767–D772. doi:10.1093/nar/gkn892
- Krauthammer, M., Kaufmann, C. A., Gilliam, T. C., and Rzhetsky, A. (2004). Molecular triangulation: Bridging linkage and molecular-network information for identifying candidate genes in Alzheimer's disease. *Proc. Natl. Acad. Sci.* 101, 15148–15153. doi:10.1073/pnas.0404315101
- Lee, D. S., Park, J., Kay, K. A., Christakis, N. A., Oltvai, Z. N., and Barabasi, A. L. (2008). The implications of human metabolic network topology for disease comorbidity. *Proc. Natl. Acad. Sci.* 105, 9880–9885. doi:10.1073/pnas.0802208105
- Liu, B., Jin, M., and Zeng, P. (2015). Prioritization of candidate disease genes by combining topological similarity and semantic similarity. *J. Biomed. Inform.* 57, 1–5. doi:10.1016/j.jbi.2015.07.005
- Liu, T.-Y., Liu, Z.-P., Zhao, X.-M., and Chen, L. (2011). Future Work. *J. Am. Med. Inform. Assoc.* 19, 241–248. doi:10.1007/978-3-642-14267-3_20
- Luo, J., and Liang, S. (2015). Prioritization of potential candidate disease genes by topological similarity of protein-protein interaction network and phenotype data. *J. Biomed. Inform.* 53, 229–236. doi:10.1016/j.jbi.2014.11.004
- Matys, V., Fricke, E., Geffers, R., Gößling, E., Haubrock, M., Hehl, R., et al. (2003). TRANSFAC(R): transcriptional regulation, from patterns to profiles. *Nucleic Acids Res.* 31, 374–378. doi:10.1093/nar/gkg108
- Menche, J., Sharma, A., Kitsak, M., Ghiassian, S. D., Vidal, M., Loscalzo, J., et al. (2015). Uncovering disease-disease relationships through the incomplete interactome. *Science* 347, 1257601. doi:10.1126/science.1257601
- Navlakha, S., and Kingsford, C. (2010). The power of protein interaction networks for associating genes with diseases. *Bioinformatics* 26, 1057–1063. doi:10.1093/bioinformatics/btq076
- Ortutay, C., and Vihinen, M. (2008). Identification of candidate disease genes by integrating gene ontologies and protein-interaction networks: case study of primary immunodeficiencies. *Nucleic Acids Res.* 37, 622–628. doi:10.1093/nar/gkn982
- Raap, U., Brzoska, T., Sohl, S., Pärth, G., Emmel, J., Herz, U., et al. (2003). α -Melanocyte-Stimulating Hormone Inhibits Allergic Airway Inflammation. *J. Immunol.* 171, 353–359. doi:10.4049/jimmunol.171.1.353
- Ruepp, A., Waegele, B., Lechner, M., Brauner, B., Dunger-Kaltenbach, I., Fobo, G., et al. (2010). CORUM: the comprehensive resource of mammalian protein complexes--2009. *Nucleic Acids Res.* 38, D497–D501. doi:10.1093/nar/gkp914
- Schadt, E. E. (2009). Molecular networks as sensors and drivers of common human diseases. *Nature* 461, 218–223. doi:10.1038/nature08454
- Sebastian, K., Sebastian, B., Denise, H., and Peter N, R. (2008). Walking the interactome for prioritization of candidate disease genes. *Am. J. Hum. Genet.* 82, 949–958.
- Sharma, A., Menche, J., Huang, C. C., Ort, T., Zhou, X., Kitsak, M., et al. (2012). A disease module in the interactome explains disease heterogeneity, drug response and captures novel pathways and genes in asthma. *Hum. Mol. Genet.* 46, 957–961.
- Song, G. G., and Lee, Y. H. (2013). Pathway analysis of genome-wide association study on asthma. *Hum. Immunol.* 74, 256–260. doi:10.1016/j.humimm.2012.11.003
- Su, Y., Li, S., Zheng, C., and Zhang, X. (2019). A heuristic algorithm for identifying molecular signatures in cancer. *IEEE Trans. Nanobioscience* 19, 132–141. doi:10.1109/TNB.2019.2930647
- Su, Y., Liu, C., Niu, Y., Cheng, F., and Zhang, X. (2021). A community structure enhancement-based community detection algorithm for complex networks. *IEEE Trans. Syst. Man. Cybern. Syst.* 51, 2833–2846. doi:10.1109/tsmc.2019.2917215
- Su, Y., Zhu, H., Zhang, L., and Zhang, X. (2020). "Identifying disease modules based on connectivity and semantic similarities," in *Proceedings of 14th International Conference on Bio-inspired Computing: Theories and Applications*, 1–8. doi:10.1007/978-981-15-3415-7_3
- Susan Dina, G., Jorg, M., and Albert-Laszlo, B. (2015). A disease module detection algorithm derived from a systematic analysis of connectivity patterns of disease proteins in the human interactome. *Plos Comput. Biol.* 11, e1004120.
- Tian, Y., Su, X., Su, Y., and Zhang, X. (2020). EMODMI: A multi-objective optimization based method to identify disease modules. *IEEE Trans. Emerging Top. Comput. Intelligence* 1, 13. doi:10.1209/TETCI.2020.3325117

- Tu, Z., Wang, L., Xu, M., Zhou, X., Chen, T., and Sun, F. (2006). Further understanding human disease genes by comparing with housekeeping genes and other genes. *BMC Genomics* 7, 31–13. doi:10.1186/1471-2164-7-31
- Venkatesan, K., Rual, J.-F., Vazquez, A., Stelzl, U., Lemmens, I., Hirozane-Kishikawa, T., et al. (2008). An empirical framework for binary interactome mapping. *Nat. Methods* 6, 83–90. doi:10.1038/nmeth.1280
- Vercelli, D. (2008). Discovering susceptibility genes for asthma and allergy. *Nat. Rev. Immunol.* 8, 169–182. doi:10.1038/nri2257
- Vinayagam, A., Stelzl, U., Foulle, R., Plassmann, S., Zenkner, M., Timm, J., et al. (2011). A directed protein interaction network for investigating intracellular signal transduction. *Sci. Signaling* 4, rs8. doi:10.1126/scisignal.2001699
- Wang, R.-S., and Loscalzo, J. (2018). Network-based disease module discovery by a novel seed connector algorithm with pathobiological implications. *J. Mol. Biol.* 430, 2939–2950. doi:10.1016/j.jmb.2018.05.016
- Wang, X., Gulbahce, N., and Yu, H. (2011). Network-based methods for human disease gene prediction. *Brief. Funct. Genomics* 10, 280–293. doi:10.1093/bfpgp/elt024
- Warren, R. M. L., Pointon, L., Caines, R., Hayes, C., Thompson, D., and Leach, M. O. (2002). What is the recall rate of breast mri when used for screening asymptomatic women at high risk? *Magn. Reson. Imaging* 20, 557–565. doi:10.1016/s0730-725x(02)00535-0
- Wen, Z., Liu, Z. P., Liu, Z., Zhang, Y., and Chen, L. (2013). An integrated approach to identify causal network modules of complex diseases with application to colorectal cancer. *J. Am. Med. Inform. Assoc.* 20, 659–667. doi:10.1136/amiajnl-2012-001168
- Wu, X., Pang, E., Lin, K., and Pei, Z.-M. (2013). Improving the measurement of semantic similarity between gene ontology terms and gene products: insights from an edge- and ic-based hybrid method. *Plos One* 8, e66745. doi:10.1371/journal.pone.0066745
- Xu, J., and Li, Y. (2006). Discovering disease-genes by topological features in human protein-protein interaction network. *Bioinformatics* 22, 2800–2805. doi:10.1093/bioinformatics/btl467
- Zanzoni, A., Soler-López, M., and Aloy, P. (2009). A network medicine approach to human disease. *Febs Lett.* 583, 1759–1765. doi:10.1016/j.febslet.2009.03.001
- Zhang, T., Wang, X., and Yue, Z. (2017a). Identification of candidate genes related to pancreatic cancer based on analysis of gene co-expression and protein-protein interaction network. *Oncotarget* 8, 71105–71116. doi:10.18632/oncotarget.20537
- Zhang, X., Wang, C., Su, Y., Pan, L., and Zhang, H.-F. (2017b). A fast overlapping community detection algorithm based on weak cliques for large-scale networks. *IEEE Trans. Comput. Soc. Syst.* 4, 218–230. doi:10.1109/tcss.2017.2749282
- Zheng, C.-H., Huang, D.-S., Kong, X.-Z., and Zhao, X.-M. (2008). Gene expression data classification using consensus independent component analysis. *Genomics, Proteomics & Bioinformatics* 6, 74–82. doi:10.1016/s1672-0229(08)60022-4
- Zheng, C.-H., Huang, D.-S., and Shang, L. (2006). Feature selection in independent component subspace for microarray data classification. *Neurocomputing* 69, 2407–2410. doi:10.1016/j.neucom.2006.02.006

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2021 Liu, Zhu and Qiu. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.



Deep Learning Algorithms Achieved Satisfactory Predictions When Trained on a Novel Collection of Anticoronavirus Molecules

Emna Harigua-Souiai^{1*}, Mohamed Mahmoud Heinhane¹, Yosser Zina Abdelkrim¹, Oussama Souiai², Ines Abdeljaoued-Tej^{2,3} and Ikram Guizani¹

¹Laboratory of Molecular Epidemiology and Experimental Pathology-LR16IPT04, Institut Pasteur de Tunis, Université de Tunis El Manar, Tunis, Tunisia, ²Laboratory of Bioinformatics BioMathematics and BioStatistics (BIMS)-LR20IPT09, Institut Pasteur de Tunis, University of Tunis El Manar, Tunis, Tunisia, ³Engineering School of Statistics and Information Analysis, University of Carthage, Ariana, Tunisia

OPEN ACCESS

Edited by:

Xiangxiang Zeng,
Hunan University, China

Reviewed by:

Laurent Emmanuel Dardenne,
National Laboratory for Scientific
Computing (LNCC), Brazil
Khanh N. Q. Le,
Taipei Medical University, Taiwan

*Correspondence:

Emna Harigua-Souiai
harigua.emna@gmail.com

Specialty section:

This article was submitted to
Computational Genomics,
a section of the journal
Frontiers in Genetics

Received: 19 July 2021

Accepted: 30 September 2021

Published: 29 November 2021

Citation:

Harigua-Souiai E, Heinhane MM,
Abdelkrim YZ, Souiai O,
Abdeljaoued-Tej I and Guizani I (2021)
Deep Learning Algorithms Achieved
Satisfactory Predictions When Trained
on a Novel Collection of
Anticoronavirus Molecules.
Front. Genet. 12:744170.
doi: 10.3389/fgene.2021.744170

Drug discovery and repurposing against COVID-19 is a highly relevant topic with huge efforts dedicated to delivering novel therapeutics targeting SARS-CoV-2. In this context, computer-aided drug discovery is of interest in orienting the early high throughput screenings and in optimizing the hit identification rate. We herein propose a pipeline for Ligand-Based Drug Discovery (LBDD) against SARS-CoV-2. Through an extensive search of the literature and multiple steps of filtering, we integrated information on 2,610 molecules having a validated effect against SARS-CoV and/or SARS-CoV-2. The chemical structures of these molecules were encoded through multiple systems to be readily useful as input to conventional machine learning (ML) algorithms or deep learning (DL) architectures. We assessed the performances of seven ML algorithms and four DL algorithms in achieving molecule classification into two classes: active and inactive. The Random Forests (RF), Graph Convolutional Network (GCN), and Directed Acyclic Graph (DAG) models achieved the best performances. These models were further optimized through hyperparameter tuning and achieved ROC-AUC scores through cross-validation of 85, 83, and 79% for RF, GCN, and DAG models, respectively. An external validation step on the FDA-approved drugs collection revealed a superior potential of DL algorithms to achieve drug repurposing against SARS-CoV-2 based on the dataset herein presented. Namely, GCN and DAG achieved more than 50% of the true positive rate assessed on the confirmed hits of a PubChem bioassay.

Keywords: deep learning, artificial neural network, SARS-CoV-2, machine learning, graph convolutional networks, drug discovery and repurposing

1 INTRODUCTION

Discovery and design of effective treatments against COVID-19 is actually an active research field. Tremendous efforts have been deployed worldwide to find new molecules with therapeutic potential against its pathogenic agent SARS-CoV-2 (Song et al., 2021). The most forerunner achievements mainly consisted in drug repurposing attempts of previously described drugs able to affect the SARS-CoV such as chloroquine and its derivatives (Vincent et al., 2005; Pastick et al., 2020; Yao et al., 2020;

Galan et al., 2021; Moiseev et al., 2021). Other antivirals or antibiotics were also assessed for their potential as COVID-19 therapeutics (Pillaiyar et al., 2020; Kelleni, 2021). Still, as of today, no candidates have been yet retained as a universal COVID-19 treatment (Hoffmann et al., 2020; Dragojevic Simic et al., 2021). Various approaches were adopted, including computational methods toward a faster discovery of drugs, given the urge of the global sanitary situation.

Computational approaches may be split into two subcategories: Structure-Based Drug Discovery (SBDD) and Ligand-Based Drug Discovery (LBDD). For SBDD, the structure of a molecular target is used to perform virtual screenings of large chemical libraries. The most popular targets are the Spike protein, known as the S protein, the 3-Chymotrypsin-Like cysteine protease (3CLpro), also called the main protease (Mpro), and the Papain-Like protease (PLpro) (Chellapandi and Saranya, 2020; Trezza et al., 2020; Zhang et al., 2020; Zhai et al., 2021). These approaches rely on the availability of structural data of SARS-CoV-2 proteins, which are noticeably abundant as compared to other organisms. In fact, as of July 14, 2021, the RCSB PDB database accounted for 446 structures of the S protein and its binding domains, 360 crystal structures of the 3CLpro, 35 for the PLpro, and 505 structures corresponding to other SARS-CoV-2 proteins (RCSB).

On the other hand, LBDD is more likely dependent on the availability of data on the biological activity of molecules. Machine learning (ML) approaches demonstrated their ability to predict the activity of novel molecules based on such data (Altae-Tran et al., 2017; Lo et al., 2018; Vamathevan et al., 2019; Zeng et al., 2019; Korkmaz, 2020). The underlying assumption is that chemically and topologically similar compounds may have similar bioactivities and targets (Gfeller et al., 2014; Shi et al., 2015; Perualila-Tan et al., 2016). These approaches were extensively used in novel drug discovery (DD) and repurposing against COVID-19 (Keshavarzi Arshadi et al., 2020; Bung et al., 2021; Yang et al., 2021). In fact, dedicated resources have been developed to facilitate and enhance international efforts toward DD against COVID-19. Namely, the COVID-19 Moonshot Consortium has deployed international efforts in tackling data collection and curation of molecules targeting the 3CLpro of SARS-CoV-2. Their approach allied with SBDD and LBDD techniques (Achdout et al., 2020). In fact, data availability is a cornerstone in building reliable ML models. This being said, data in DD is often sparse, heterogeneous, noisy, or too few. Multiple efforts have been made to build ML algorithms able to deal with such limitations and achieve satisfactory predictions (Duran-Frigola et al., 2019; Irwin et al., 2020; Yang et al., 2020).

Beyond COVID-19 research, ML and deep learning (DL) were applied to a variety of DD projects. Applications can be split into two types: 1) activity prediction through regression and 2) classification of molecules into classes, mostly active vs. inactive (Rifaoglu et al., 2019; Vamathevan et al., 2019). ML algorithms are implemented and trained on binary or float values descriptors of a fixed length, generated using a chemical structure encoding system. The most popular encoding systems are either the physicochemical descriptors

(molecular weight, H-bond donors, H-bond acceptors, rotatable bonds, etc.) or molecular fingerprints (Jing et al., 2018). The latter correspond to a variety of algorithms that are able to capture topological features and properties within chemical structures. Most of them calculate a series of binary digits that encode the presence or the absence of particular substructures in the molecule. More recently, there was a rising interest in graph convolution networks as chemical structure encoding systems in the frame of DL applications in LBDD (Micheli, 2009; Lusci et al., 2013; Duvenaud et al., 2015; Kearnes et al., 2016; Altae-Tran et al., 2017). A molecular graph is the most common machine-readable representation (David et al., 2020). Chemical representations in these schemes lie in mapping the atoms and bonds of a molecule into sets of nodes and edges. Spatial relationships between the nodes are then encoded through network embedding. This leads to a low-dimensional vector representation of the molecular graph, preserving both network topology structure and node content information (Wu et al., 2020). Graph convolutional networks (GCN) apply then a series of convolution layers to construct the whole molecule encoder. Graphs have irregular designs and sizes; there is no spatial order attached to the nodes. As a result, traditional convolution on regular grid-like structures cannot be applied directly to graphs. In the literature, efforts have been made to generalize the convolution operator to non-Euclidean structured data, resulting in convolutional graph networks (CGNs). GCNs have emerged as the state-of-the-art encoding when it comes to DD (Sun et al., 2020), especially when one seeks to extract features with respect to the data structure. This extraction is done automatically from raw inputs (Lavecchia, 2019). Duvenaud et al. presented a graph convolution method to encode molecule structures using a differentiable neural network (NN) that generalizes fingerprint-based features *via* backpropagation on an undirected graph representation of the molecule (Duvenaud et al., 2015). The authors demonstrated that applying graph convolution enhances property predictions as compared to conventional circular fingerprints. Kearnes et al. also described a graph convolution approach that learns from a graph representation of the molecule while taking into account its structure and composition (Kearnes et al., 2016).

Here, we present a dataset of molecules validated for their effects on SARS-CoV-2 and/or SARS-CoV through viral growth inhibition, cell-based, or enzymatic experiments. Data were collected through an extensive search of the literature and databases, curated and formatted for cheminformatics simulations toward LBDD against COVID-19. Chemical structures of the molecules were then encoded through multiple systems to be readily useful as input to conventional ML algorithms or for GCN. We run an extensive set of simulations under different splitting and formatting conditions of the data to identify the ML and DL algorithms that could achieve satisfactory results. Most promising models were then optimized, and their performances were validated through cross-validation. An external validation step was performed to assess

the potential of these algorithms to achieve drug repurposing using experimental data on the FDA-approved drugs collection.

2 MATERIALS AND METHODS

2.1 Data Collection

The data collection process included three distinct approaches. The first consisted in literature mining. We collected data on molecules described in peer-reviewed papers as anticoronavirus effectors. Two beta-coronavirus species were considered: SARS-CoV (Severe Acute Respiratory Syndrome Coronavirus, 2003) and SARS-CoV-2 (Severe Acute Respiratory Syndrome Coronavirus, 2019). The second approach consisted in retrieving data on molecules deposited in the RCSB PDB as cocrystals with SARS-CoV and SARS-CoV-2 proteins, mainly the 3CL-protease and the Papain-Like protease. When available, activity data on these cocrystallized inhibitors were fetched from corresponding scientific publications. The third approach consisted in retrieving data from bioassays deposited in the PubChem database (Kim et al., 2020). Priority was given to bioassays targeting SARS-CoV-2 or related molecular targets, with a special interest in large bioassays on other coronaviruses. Data collected from these bioassays correspond to viral growth inhibition or cell-based tests targeting a given viral enzyme. In total, data from 10 COVID-19 bioassays were included. These were complemented by four bioassays targeting SARS-CoV. PubChem IDs of the bioassays, their types, and sizes are listed in **Supplementary Table S1**. Bioassay datasets were then formatted to be merged with the literature dataset previously collected. Data collected on each molecule included chemical structure, name, chemical name if indicated, activity, target virus, and any additional information such as identifiers in the PubChem database, *in vitro* IC₅₀ values, cellulo IC₅₀ values, and any other valuable biological data (*in vivo* EC₅₀, inhibition rate at a given concentration, etc.). The chemical structure of the molecules was encoded using the Simplified Molecular Input Line Entry System (SMILES). For compounds with a graphical description of their structure in the literature, we used the Optical Structure Recognition Application (OSRA) tool (Filippov and Nicklaus, 2009) to correctly infer the corresponding SMILES. For compounds referred to in the literature using a common name, SMILES were directly retrieved from the PubChem database. Duplicates were removed using a similarity threshold of 97% based on the Tanimoto coefficient. Each molecule was assigned an activity status that can be “active,” “inactive,” or “inconclusive.” For molecules retrieved from PubChem bioassays, these status values were provided from the experimentalists’ data. For molecules fetched in the literature, these status values were deduced from the authors’ conclusions. For the molecules retrieved from the PDB records, these status values were assigned to “active” by default. In fact, we considered that the ability of a molecule to bind to a given protein receptor encloses valuable information on potential active moieties, although no biological activity is reported for these molecules. Any data point with inconclusive or blurry value was discarded for robustness sake.

2.2 Datasets Construction

The benchmark datasets used herein were split using two different approaches. First, a random split with no consideration for chemical equilibration among the training, validation, and test sets was applied. Then, a scaffold split (Ramsundar et al., 2019) was applied. The scaffold split method would cluster molecules based on the Murcko scaffold calculated using RDkit. Compounds with different scaffolds are placed into different sets (Ramsundar et al., 2019). This significantly reduces the overlap of chemical scaffolds between the training and the test sets (Ramsundar et al., 2019).

In addition, we tested how the size of the validation and test sets would affect the algorithms’ performances. Thus, we tested two scenarios: 80/10/10 and 60/20/20 split. An additional splitting method of the original dataset that permitted the generation of category-specific subsets for validation purposes was applied. Undersampling and oversampling were applied in order to obtain equilibrated datasets in each case. Undersampling consisted in reducing the inactive molecules subset to achieve equilibrated classes. Oversampling consisted in artificially generating additional SMILES of the active molecules in order to reach the inactive subset size.

2.3 Molecular Structure Embedding

Based on the SMILES, we calculated either molecular fingerprints or graph convolution-based features that consist in binary or float values vectors to be used as input to the ML and DL algorithms, respectively. As fingerprints, we chose the extended-connectivity fingerprints with a radius of two atoms (ECFP4), also known as the circular Morgan fingerprints (Rogers and Hahn, 2010), to encode the molecule structures for ML algorithms. We used the RDkit library to generate 2,048-bit length ECFP4. Molecules with erroneous SMILES or chemistry were removed at this stage. We used these fingerprints to calculate the Tanimoto coefficient of similarity in a pairwise fashion. This metric consists of the fraction of the intersection over the union of the set of chemical substructures between two molecules. It is one of the most used to assess the chemical similarity between molecules (Chung et al., 2019). As for the graph convolution-based features, depending on the DL architecture requirement, two featurizers were used:

- The ConvMolFeat featurizer (Duvenaud et al., 2015) to generate input for the Graph Convolutional (GraphConv) (Duvenaud et al., 2015) and the Directed Acyclic Graph (DAG) (Lusci et al., 2013) models.
- The MolGraphConvFeat (Kearnes et al., 2016) to generate input for the GAT (Velickovic et al., 2018) and the GCN (Kipf and Welling, 2016) models.

Graphical convolutional models map molecules as undirected graphs whose vertices and edges represent individual atoms and bonds, respectively. Graphical convolutions extract meaningful patterns from basic descriptions of graph structure (atom and bond properties and graph distances) to form molecule-level representations. They are considered fully integrated approaches to virtual screening. The output of the model is

invariant to the order in which the atom and bond information is encoded in the input. The graph represents class similarity information and is fed into DL classification models.

2.4 ML and DL Algorithms

We implemented multiple artificial intelligence (AI) algorithms to develop classification models: ML, ensemble learning methods (EL), and DL. We implemented seven ML algorithms, out of which two are simple ML algorithms, namely, Logistic Regression (LR) and Support Vector Machine (SVM). Five additional EL algorithms were implemented, namely, Random Forests (RF), Multitask Classifier (MTC), IRV-MTC, Robust MTC, and Gradient Boosting (XGBoost). EL are learning algorithms that construct the first set of classifiers and then construct a new one by taking a weighted vote of data predictions from the previous classifiers (Dietterich, 2000). These algorithms were implemented under Scikit-learn, an open-source python library (Pedregosa et al., 2011). LR measures the relationship between a categorical dependent variable and one or more explanatory variables. This is performed by estimating probabilities using a logistic function, which is the cumulative logistic distribution, thus predicting the probability of certain outcomes. The SVM is one of the most popular supervised ML algorithms. It is effective in high-dimensional spaces. The hyperplane learning in the SVM algorithm can be performed using different kernel functions for the decision function. The RF method is an ensemble method, based on decision trees. The model fits on various subsamples of the dataset and uses averaging to improve predictive accuracy and control overfitting. The Gradient Boosting model implemented herein is called XGBoost (Paul et al., 2020). It is an extremely gradient boosting algorithm and a decision tree-based boosting integration algorithm (Ericksen et al., 2017). Further ensemble methods have been tested: Multitask Classifier (MTC), IRV-MTC, and Robust MTC. These are fully connected NN, where various hyperparameters are optimized. They operate like EL algorithms, where they integrate data from different tasks to achieve classification. When used on a single task data, they are a nonlinear classifier that performs repeated linear and nonlinear transformations on one single task (Ramsundar et al., 2017).

Then, four DL architectures were implemented under the DeepChem library (Ramsundar et al., 2019): the Graph Convolutional Model (GraphConv), the DAG model, the Graph Attention Networks model (GAT), and the GCN model. The GraphConv Model (Duvenaud et al., 2015) learns a vector representing the compound from the graph-based representation of the molecule. It predicts the target value directly through graph convolution operations. Convolutional networks operate the same operation locally and globally and combine the information in a common pooling step. Feature extraction involves computing an initial feature vector and a list of neighbors for each atom. The feature vector summarizes the local chemical environment of the atom, including atomic types, hybridization types, and valence structures. The neighbor lists map the connectivity of the entire molecule and are then

processed in each model to generate graph structures (Wu et al., 2018). The DAG model is an ensemble of recursive NN that associate all vertex-centered acyclic orientations of the graph representation of the molecule. It is slightly dependent on the molecular descriptors since suitable representations are learned from the DAG representation (Lusci et al., 2013). The graph attentional layer (GAT) model (Velickovic et al., 2018) is a convolutional NN that operates on graph-structured data, taking advantage of self-attention hidden layers. The attention mechanism is applied in a shared manner to all edges of the graph and thus does not depend on prior access to the overall structure of the graph or to (characteristics of) all its nodes. It allows assigning (implicitly) different importance to the nodes of the same neighborhood. GCN is an implementation of graph convolutional NN (Kipf and Welling, 2016). It learns hidden layer representations that are able to encode both individual features of nodes and their respective environments. It computes a weighted sum of the node representations in the graph, where the weights are computed by applying a gating function to the node representations, and then applies a max pooling of the node representations. It perform the final prediction using a multilayer perceptron (MLP) over a concatenation of the last convolution layer output. It differs from the GraphConv model by the fact that, for each graph convolution, the learnable weight in this model is shared across all nodes. The GraphConv model computes separate learnable weights for nodes.

Under the DeepChem library, both the GraphConv Model and the DAG model were implemented to learn from MolConv featurizer (Duvenaud et al., 2015) that corresponds to GCN that learns from circular morgan fingerprints-like representation of the molecule. On the other hand, the GAT and GCN models have been implemented in a way that they can learn from the MolGraphConv featurizer (Kearnes et al., 2016). Data were split into training, validation, and test sets. The hyperparameters of the DL models were tuned using the loss of the validation sets.

2.5 Model Evaluation and Selection

We performed the first comparison of all models' performances with hyperparameters set to the optimal values obtained through the MoleculeNet benchmarks (Wu et al., 2018). To better evaluate the different models, we calculated multiple performance metrics, including the ROC-AUC, accuracy, F1-score, Matthews correlation coefficient (MCC) (Matthews, 1975), and Cohen's Kappa coefficient (κ). Then, we performed a cross evaluation of the model performances when trained and tested on stratified subsets of the data based on the different categories of targets. Accuracy, F1-score, Recall, and specificity were used as evaluation metrics for these simulations.

For the metric definitions, the following abbreviations are used: the number of true positives (TP), the number of false positives (FP), the number of true negatives (TN), and the number of false negatives (FN). Specificity, also called the False Positive Rate (FPR), is the model's ability to correctly reject an inactive molecule. Specificity of a test is the proportion of molecules that are truly inactive, which are classified as is. It is defined as follows:

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \quad (1)$$

Model Recall can be thought of as the percentage of true class labels correctly identified by the model as true. It is equal to the model sensitivity in binary classification and is also called the True Positive Rate (TPR). It is defined as follows:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (2)$$

The F1-score is the harmonic mean of the Recall and precision:

$$\text{F1-score} = 2 * \frac{\text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} \quad (3)$$

where precision is the probability of a predicted true label is predicted as true and is defined as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (4)$$

Accuracy is the percentage of correctly identified labels out of the entire population.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \quad (5)$$

The ROC-AUC score tells how much the model is capable of distinguishing between classes. It varies between 0 and 1, where 1 means a perfect prediction. The MMC is a correlation coefficient between the observed and predicted binary classifications. It is between -1 and +1, where +1 indicates a perfect prediction, 0 indicates no better than random, and -1 indicates prediction and observation are totally different.

$$\text{MMC} = \frac{(\text{TP} * \text{TN}) - (\text{FP} * \text{FN})}{\sqrt{(\text{TP} + \text{FP}) * (\text{TP} + \text{FN}) * (\text{TN} + \text{FP}) * (\text{TN} + \text{FN})}} \quad (6)$$

Cohen's Kappa method measures interclassifier agreement in qualitative classification tasks. It evaluates the agreement between two classifiers and takes into account the random occurrence of the agreement. A value close to one denotes better agreement between the results and ground truth.

$$\kappa = \frac{2 * (\text{TP} * \text{TN} - \text{FN} * \text{FP})}{(\text{TP} + \text{FP}) * (\text{FP} + \text{TN}) + (\text{TP} + \text{FN}) * (\text{FN} + \text{TN})} \quad (7)$$

The best performers were then selected for hyperparameter optimization on the particular anticoronavirus dataset collected through the present study. Their performances were mainly assessed through ROC-AUC, F1-score, Recall, Accuracy, MCC, and Cohen's Kappa scores, which are a set of popular metrics in evaluating ML algorithms in a variety of applications (Le et al., 2019; Le and Huynh, 2019; Le and Nguyen, 2019). ML algorithm optimization included all optimizable parameters for the respective model. For DL architectures, the number of epochs, the batch size, the learning rate, the dropout, or the number of graph features when they apply were optimized. We selected the configuration that maximizes the ROC-AUC of the model on the validation set. The accuracy, the F1-score, the MCC, and Cohen's Kappa coefficient were also calculated for all combinations.

Tenfold cross-validation was performed, and the mean ROC-AUC, F1-score, and Recall values were reported. A stratified validation was also applied in order to assess the ability of the algorithms trained on the heterogeneous dataset to correctly predict active molecules from different categories of experiments. The sensitivity (Recall) and specificity were herein used as performance indicators. The optimized models were then subject to an external validation using an unseen set of molecules. We used a PubChem bioassay that consisted in a primary screen of 1,518 FDA-approved molecules against SARS-CoV-2-infected cells (AID_1409594). A total number of 17 hits were retained as potentially active molecules, and their antiviral efficacy was further confirmed through a second assay (AID_1409595). We performed a prediction of these 1,518 FDA-approved drugs as anti-SARS-CoV-2 inhibitors using the best performing algorithms.

3 RESULTS

3.1 Integration Efforts Led to a Curated Dataset of Anticoronavirus Molecules

We collected data on molecules with anticoronavirus effects, out of which 533 were retrieved from literature. All remaining compounds were collected from 14 PubChem bioassays. Since activity types were different from one source to the other, we considered the activity as a binary variable. Initially, four classes of activity status were listed: active, inactive, unspecified, and inconclusive. Only molecules within the first two classes were retained in the frame of the present work. The combined set of active and inactive molecules was subject to redundancy check, and duplicates were removed. The number of active molecules was equal to 1,305 at this stage. We then looked to obtain an equal number (1,305) of inactive molecules, which were in larger numbers, namely, within large bioassays. Thus, from some SARS-CoV bioassays, only a subset of inactive molecules was randomly selected (see **Supplementary Table S1**). Ultimately, 2,610 nonredundant compounds were obtained. We performed a structural similarity analysis to assess the chemical diversity of the dataset (**Figure 1**). Based on the circular Morgan fingerprints, we calculated the pairwise distance between all compounds using the Tanimoto similarity coefficient. The similarity distribution demonstrated too few values higher than 60%. This indicates a high chemical diversity within the dataset. Also, experiments that revealed these molecules included enzymatic activity assays against one of the viruses proteases 3CLpro and PLpro, inhibition assays targeting the whole virus, and cell-based assays. We defined most relevant experiment categories as follows: 3CLpro_cov, 3CLpro_cov2, PLpro-cov, PLpro_cov2, and viral_cov2. Each category presents a specific count in terms of active and inactive molecules (**Figure 1**), revealing unbalanced and insufficient data within some categories. Within the molecules with known molecular targets, only 0.7% were targeting the PLpro of SARS-CoV-2, while 40.6% were targeting PLpro of SARS-CoV (**Figure 1**). The remaining molecules were targeting the 3CLpro of SARS-CoV-2 (6.3%)

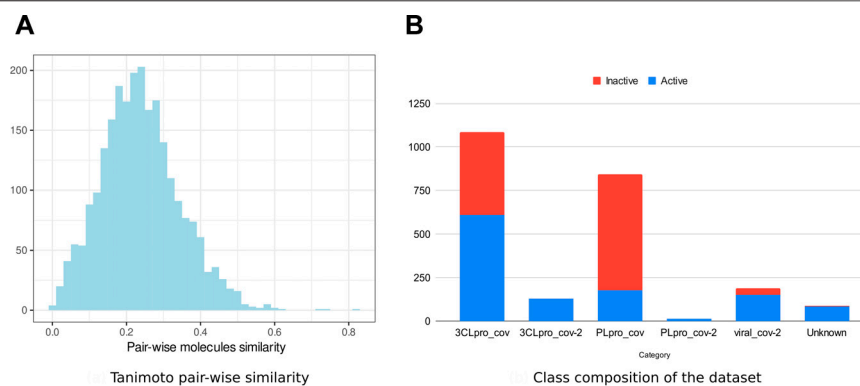


FIGURE 1 | Anticoronavirus dataset composition. **(A)** Distribution of the pairwise chemical similarity among the molecules based on the Tanimoto coefficient. **(B)** Proportions of “active” and “inactive” molecules within each experimental category.

and 3CLpro SARS-CoV (52.4%). This bias reflects the higher interest toward the 3CLpro as a therapeutic target against coronaviruses (Yang et al., 2021; Zhai et al., 2021).

3.2 Graph Convolution-Based Models Compete With Baseline ML Algorithms

At this stage, we disposed of 2,610 anticoronavirus molecules. We used a random and a scaffold split of the dataset using two splitting proportions of the training, validation, and test sets as follows: 80/10/10 and 60/20/20. We seek to identify which scenario is overall optimal. The final datasets, ready for the upcoming experiments, are available on GitHub.

We first run preliminary simulations of seven ML algorithms and four DL algorithms using the hyperparameter values released by the MoleculeNet authors (Wu et al., 2018). These optimized values were tuned on multiple types of datasets related to DD tasks. Test set representing 10% of the dataset derived significantly better results than test sets of size 20% (Supplementary Table S2). This highlighted the need to keep the training set at its higher size in order to reach satisfying levels of training. Such proportions also demonstrated the highest scores of few-shot learning algorithms (Liu et al., 2021).

To better understand to which extent the heterogeneity of our dataset may be influential, we considered a subset of homogeneous data from the largest PubChem bioassay on SARS-CoV within our dataset: AID_1706. It is a biochemical assay targeting the enzymatic activity of the 3CLpro of SARS-CoV, through which 290,893 compounds were tested. A total of 405 molecules showed an inhibitory effect on the 3CLpro-mediated peptide cleavage. Based on this bioassay, we generated one undersampled (810 molecules) and one oversampled (2,430 molecules) homogeneous datasets. On randomly split data, ROC-AUC scores on the heterogeneous dataset were the most stable across the different algorithms. The best results were exhibited by RF and SVM on the oversampled homogeneous dataset. For the DL algorithms, GraphConv model, DAG, and GCN demonstrated satisfying performances (> 80%)

on the oversampled and heterogeneous datasets, with comparable values. Overall, six out of eleven presented similar ROC-AUC scores between the heterogeneous and the oversampled homogeneous datasets (Figure 2A). Noticeably, these datasets had comparable sizes and were larger than the undersampled homogeneous dataset. This confirms the sensitivity of the AI models' performances to the dataset's size (Yang et al., 2020).

On scaffold-based split datasets, ROC-AUC scores were lower than those obtained with the randomly split data (Figure 2B). Moreover, the lowest values were observed for the oversampled homogeneous data, while the highest were obtained with the undersampled homogeneous data. The heterogeneous dataset achieved scores comparable to the undersampled dataset varying between 61 and 80%. This scheme was observed in overall simulations (Figures 2C,D). The difference between scores obtained with the oversampled and the heterogeneous datasets, at equal sizes, indicated a lower chemical diversity (number of scaffolds) within the homogeneous dataset. Thus, scaffold splitting induced lower diversity across the train and the test sets, which points out the interest of using a random split of the heterogeneous dataset in building performing ML/DL models. For the upcoming simulations, we will report results on the heterogeneous dataset using an 80/10/10 random split.

The scores of the training, the validation, and the test sets obtained with all splitting combinations showed little to no overfitting, as no significant differences were observed between these sets' scores overall (Table 1). According to the ROC-AUC scores on the test set, RF and SVM were the best classifiers within the ML/EL algorithms (Figure 2). Although the Multitask Classifier (MTC) and its variants IRV and Robust MTC exhibited higher Recall, they exhibited lower values of ROC-AUC and F1-score. We concluded that RF and SVM were the most likely to correctly predict the active molecules as being active. In the set of DL architectures, the DAG and the GCN models were the best performers. They both achieved ROC-AUC scores of 87%,

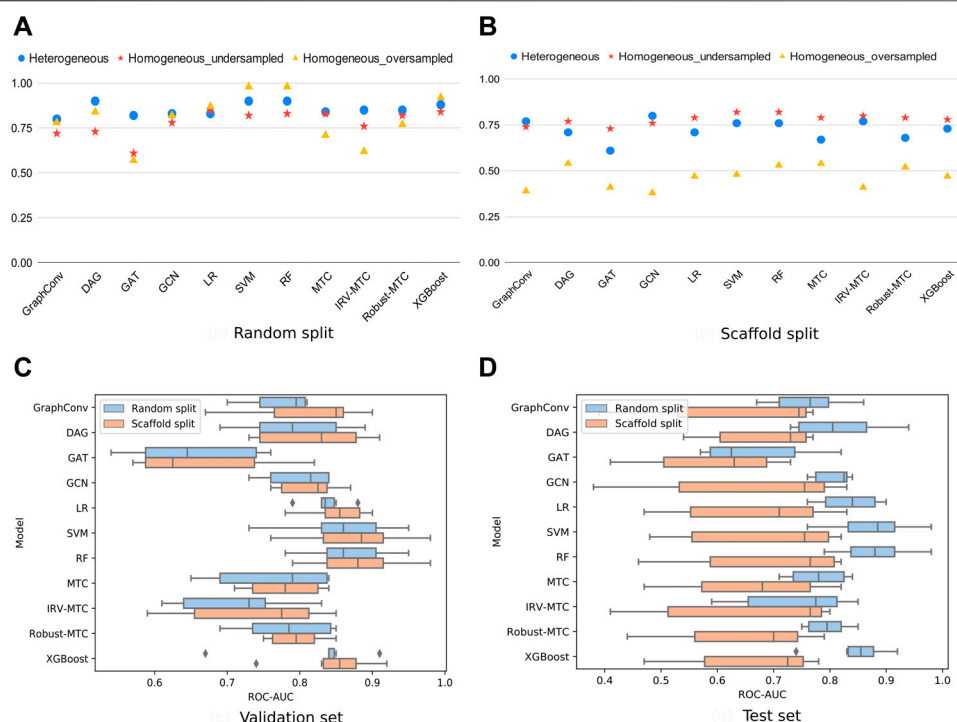


FIGURE 2 | ROC-AUC scores of all models for three different datasets (heterogeneous, undersampled homogeneous, and oversampled homogeneous). **(A)** ROC-AUC scores achieved by all models under the random 80/10/10 split. **(B)** ROC-AUC scores achieved by all models under the scaffold 80/10/10 split. **(C)** Boxplots of the ROC-AUC scores achieved by each model on all validation subsets (heterogeneous, undersampled homogeneous included) and with both splitting proportions (80/10/10; 60/20/20). **(D)** Boxplots of the ROC-AUC scores achieved by each model on all test subsets (heterogeneous, undersampled homogeneous, and oversampled homogeneous included) and with both splitting proportions (80/10/10; 60/20/20).

TABLE 1 | Performances of 11 algorithms in predicting activity class of the anticoronavirus dataset. Optimized settings based on the MoleculeNet benchmarks were considered for all models.

Model	Train ROC-AUC	Validation ROC-AUC	Test ROC-AUC	Train F1-score	Validation F1-score	Test F1-score	Train Recall	Validation Recall	Test Recall
GraphConv	0.99	0.80	0.86	0.98	0.75	0.79	0.98	0.75	0.80
DAG	0.99	0.82	0.87	0.99	0.72	0.73	0.98	0.68	0.68
GAT	0.75	0.77	0.82	0.62	0.65	0.69	0.54	0.55	0.61
GCN	0.94	0.82	0.87	0.86	0.75	0.79	0.88	0.75	0.82
LR	0.99	0.81	0.89	0.97	0.76	0.82	0.97	0.77	0.82
SVM	0.99	0.86	0.90	0.97	0.80	0.82	0.97	0.79	0.82
RF	0.99	0.86	0.90	0.99	0.78	0.81	0.99	0.80	0.81
MTC	0.81	0.77	0.84	0.67	0.71	0.68	0.99	0.99	0.99
IRV-MTC	0.82	0.82	0.85	0.75	0.78	0.76	0.88	0.89	0.90
Robust MTC	0.83	0.80	0.85	0.71	0.73	0.71	0.97	0.96	0.99
XGBoost	0.93	0.84	0.88	0.85	0.76	0.80	0.82	0.73	0.84

F1-scores of 73 and 79%, and Recall values equal to 68 and 82%, respectively (Table 1). Noticeably, the DAG model had quite higher performances on the train set (99% for all metrics). This was not the case for the GCN. This indicated that the herein used hyperparameters for the DAG model were close to the optimal configuration for our case study. We may expect better results for the GCN algorithm after the optimization step. For the upcoming steps, we will consider

the RF, the DAG, and the GCN models for hyperparameters tuning and optimization.

3.3 Optimization Led to Comparable Performances of all Models

Hyperparameters tuning of the selected models led to the identification of the combination of parameters that

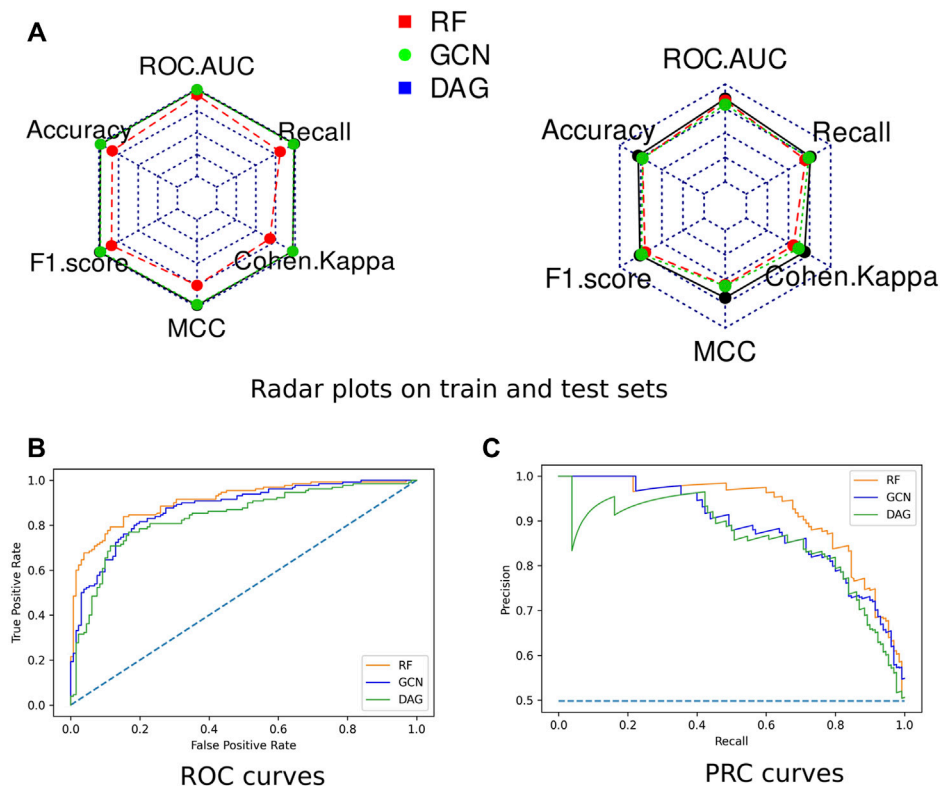


FIGURE 3 | Performances of the optimized models. **(A)** Radar plots of the models' performances assessed on the train set (**left**) and the test set (**right**) through ROC-AUC, F1-score, Accuracy, Cohen's Kappa, MCC, and Recall. **(B)** The ROC curve of all three models. **(C)** The Precision-Recall (PR) curve of all three models.

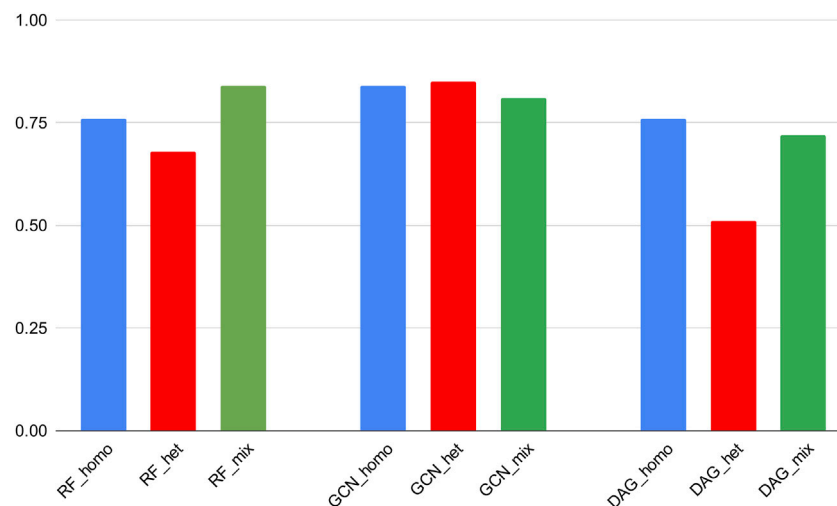


FIGURE 4 | ROC-AUC scores of the best classifiers tested on stratified subsets of the data (homogeneous, heterogeneous, and mixed).

maximizes the model's ROC-AUC score. The detailed optimization results, the retained configurations for each model, and the corresponding performances in terms of ROC-AUC, accuracy, F1-score, MCC, and Cohen's Kappa coefficient

were reported in **Supplementary Table S3**. Learning rates, dropout, and the number of learned features appeared to be the most influential parameters on model performances. In fact, the optimal thresholds for the GCN model were a learning rate of

TABLE 2 | Tenfold cross-validation results for the best classifiers. Scores are presented as mean values \pm SD based on 10 iterations.

Model	ROC-AUC	F1-score	Recall
RF	0.85 \pm 0.026	0.78 \pm 0.027	0.76 \pm 0.032
DAG model	0.79 \pm 0.013	0.73 \pm 0.052	0.74 \pm 0.103
GCN model	0.83 \pm 0.026	0.73 \pm 0.037	0.70 \pm 0.082

0.001 and a dropout of 0.1. For the DAG model, the optimal learning rate was 0.0005, and the number of learned features per atom in the graph was equal to 30. The optimal batch size and number of epochs for both models were 64 and 40, respectively (**Supplementary Table S3**).

Radar plots representing all computed scores for each model on the train and test sets were generated (**Figure 3**). None of the algorithms presented an overfitting trend. They all exhibited round-shaped radar plots indicating no differential performance based on the different scoring metrics. Overall, the RF algorithm slightly outperformed both DL algorithms. All three models presented MCC values higher than 0.5, indicating their ability to provide a satisfying class prediction for anticoronavirus molecules (**Supplementary Table S3**). The RF and DAG models exhibited Cohen's Kappa coefficient higher than 0.6, which indicates the substantial power of these algorithms in distinguishing both classes. The GCN model presented a coefficient value equal to 0.56, indicating a fair interrater power.

The Receiver Operating Characteristic (ROC) curves exhibited smooth exponential-like shapes for all models, indicating satisfying classification power. The Precision-Recall (PR) curves also presented fair shapes for a balanced dataset (**Figure 3**). At last, we performed a tenfold cross-validation. The average values of ROC-AUC, the F1-score, and the Recall over ten iterations were reported with the standard deviation values in **Table 2**. RF kept exhibiting the highest scores, although values were comparable across the three models. GCN achieved a higher ROC-AUC score as compared to the DAG model, an equivalent F1-score, and a lower Recall. Our results so far indicated that DL models kept achieving scores slightly lower than those of RF, despite being comparable.

3.4 GCN Model Demonstrated Noticeable Generalization Power

The last validation step was performed on the three optimized algorithms in order to assess their predictive power in identifying lead compounds against coronaviruses in general and SARS-CoV-2 in particular. Considering the heterogeneity of our dataset in terms of experiments and targets, it is important to assess the ability of the AI algorithms to generalize when tested on unseen datasets. To this end, we split our dataset into category-based subsets. Only categories 3CLpro_Cov and PLpro_Cov presented sufficient data points (**Supplementary Table S1**) to be used for a stratified validation of the algorithms' performances.

Homogeneous training denotes all experiments where models were trained and tested on one category subset. Heterogeneous

training denotes all experiments where models were trained on the mixed dataset and tested on one category subset. Finally, we called mixed training the experiments where models were trained and tested on the dataset consisting of a mix of categories. Performances in terms of accuracy, F1-score, Recall/sensitivity, and specificity were reported in **Supplementary Table S4**.

Algorithms' performances on the 3CLpro_cov category presented comparable values with the mixed training results. On the other hand, low Recall values were obtained with the PLpro_cov category trained on homogeneous and heterogeneous data (**Supplementary Figure S1**). It is noteworthy to report that the 3CLpro_cov subset constitutes 41.6% of the mixed dataset and presents equivalent proportions between the "active" and "inactive" classes. This was not the case for the PLpro_cov subset, which constitutes 35.8% of the mixed dataset but presented nonequilibrated class distribution (71.0% of inactive molecules). This can explain the low Recall scores obtained for this particular category (**Supplementary Figure S1**).

Noticeably, RF and GCN models could achieve comparable Recall scores through the homogeneous and heterogeneous training experiments. This means that these algorithms exhibited a similar ability to correctly predict active molecules if trained either on the mixed dataset or on the subset of the 3CLpro_cov category and then tested on the 3CLpro_cov test set. In addition, the GCN scores were maintained close to those obtained on the mixed dataset and in comparison with cross-validation results (**Figure 4**). This revealed a generalization power of this particular DL algorithm superior to the other models.

In order to confirm such findings, we performed an external validation of the three algorithms' ability to predict potential inhibitors targeting SARS-CoV-2 out of the FDA-approved drugs collection. We used a PubChem bioassay that consisted in a primary screen of 1,518 FDA-approved molecules against SARS-CoV-2-infected cells, out of which 17 molecules were retained as potentially active. Out of our mixed dataset, we removed all molecules included within this external validation set. We retrained all three models on our mixed dataset using its full content. Then, we predicted for all FDA-approved molecules from the validation set their activity class. We assessed the classification outcome in comparison with the experimental data and calculated the confusion matrix elements (TP, TN, FP, and FN) for each model under two scenarios (**Table 3**). First, we calculated the confusion matrix elements while comparing the predicted activity class without regard to the classification confidence (**Supplementary Table S5**). Then, we applied a threshold of 80% confidence to select the molecules that would be prioritized by each algorithm. Examining this set of prioritized molecules shall assess the usefulness of our classifiers in providing a successful subselection of molecules for experimental validation.

For each algorithm, we first observed the TP and FN counts out of the 17 active molecules. Overall, the GCN model achieved the highest TP count of 8/17 and the lowest FN count of 9/17. The next best performer was the DAG model with TP counts of 7/17, while RF demonstrated the lowest TP count of 4/17 (**Table 3**). Interestingly, when considering the prioritized list of molecules

TABLE 3 | External validation of the three models' performances in comparison with experimental results from the PubChem bioassay AID_1409594. Columns 2–5 report TP, TN, FP, and FN counts based on the overall predictions of the algorithms. Columns 6–9 report the TP, TN, FP, and FN counts based on the subselection of molecules with prediction confidence higher than 80%.

Activity criterion	All molecules: no confidence threshold				Subselection of molecules above the 80% confidence threshold			
	TP	TN	FP	FN	TP	TN	FP	FN
RF	4	490	119	13	1	425	12	8
DAG	7	719	340	10	3	359	99	3
GCN	8	877	182	9	5	835	147	9

using the 80% selection threshold, the GCN model achieved the best performances with most of the TP being within the priority list (5 out of 8). The same trend was observed for the TN count with 835/877 being correctly classified as inactive with confidence higher than 80%. Less satisfying rates were achieved by the DAG model (3/7 of TP and 3/10 of FN within the 80% confidence threshold selection) and RF (1/4 of TP and 8/13 of FN within the 80% confidence threshold selection). Thus, the GCN model demonstrated a higher ability to correctly classify both active and inactive molecules within the FDA-approved drugs collection.

4 DISCUSSION

AI, precisely ML and DL, have now demonstrated high potential of delivering successful research outcomes in the field of DD (Achdout et al., 2020; Gupta et al., 2021). The application of ML algorithms to cheminformatics and DD is heavily dependent on the rise of molecular encoding systems. The early descriptors consisted in a series of physicochemical properties of the molecules that rapidly demonstrated their limitations. Thus, chemical structure encoding appeared as a promising venue with the underlying hypothesis that the activity of a molecule is heavily correlated with its chemical structure (Gfeller et al., 2014; Shi et al., 2015; Perualila-Tan et al., 2016). Multiple approaches dedicated to calculate molecular fingerprints were then proposed (Bero et al., 2017). These consist in capturing topological and connectivity information within the molecule structure for an enhanced description as compared to simple physicochemical descriptors. Other groups proposed graph convolution-based algorithms that consider the molecule structure as an undirected graph where atoms are nodes and bonds are vertices. These methods were readily useful to implement DL architectures toward DD (Zitnik et al., 2018; Li et al., 2019; Zhang et al., 2019). Conventional ML methods such as RF, SVM, and simple NN demonstrated their ability to predict the inhibitory activity of molecules (Heikamp and Bajorath, 2014; Cano et al., 2017) in the particular case where datasets are limited to a few hundred molecules. On the other hand, DL algorithms achieved interesting results on larger datasets (Unterthiner et al., 2014; Aliper et al., 2016; Lenselink et al., 2017). This reflects the consistent dependency of DL algorithm performances on data size, although they are noticeably gaining ground, exhibiting as high performances as

classic ML algorithms (Gupta et al., 2021; Walters and Barzilay, 2021). As DD is a low-data domain, adapted DL approaches were proposed such as one-shot (Altae-Tran et al., 2017) and few-shot (Liu et al., 2021) learning methods based on structure-activity relationships for activity predictions. Compared to more classical approaches, they demonstrated higher predictive power using a small number of positives in their training sets. However, they showed poor capability of generalization to distinct datasets.

In the present work, we assessed the performances of seven ML algorithms and four DL algorithms in predicting the activity of molecules against the COVID-19 viral agent. The training data is a unique collection of 2,610 data points integrated from different sources. It includes molecules presenting inhibiting actions against SARS-CoV and SARS-CoV-2 through multiple and heterogeneous experiments. Our results demonstrated the usefulness of such a dataset in building ML algorithms for activity prediction tasks toward DD against COVID-19. Best performing algorithms, namely, GCN and RF, demonstrated stable performances across different training/testing simulations on stratified subsets of the data. Through external validation on unseen data, the GCN model demonstrated the highest predictive power overall. The MoleculeNet authors performed an extensive benchmarking of multiple ML/DL algorithms, including those studied herein on different tasks and datasets (Wu et al., 2018). RF and the GCN model were tested on multiple datasets (biophysics, physical chemistry, physiology, and quantum mechanics) and were often identified as the best performing algorithms within the conventional methods and the graph-based methods, respectively (Wu et al., 2018). This is in line with our findings, although no direct comparison is possible due to the difference in the datasets used and the tasks on which performances were evaluated.

Data have always been a determinant factor in delivering robust ML. In the field of DD, it is a constant challenge to overcome. Many groups made considerable efforts in constituting dedicated datasets for DD (Gaulton et al., 2012; Yang et al., 2021). The interest in merging data from multiple sources (projects, experiments, etc.) was explored by other groups (Duran-Frigola et al., 2019; Zeng et al., 2019; Irwin et al., 2020). Irwin et al. demonstrated the ability of the Alchemite, a state-of-the-art DL algorithm, to outperform the RF-based QSAR model in property prediction (Irwin et al., 2020). A recent work described a database called D3Similarity that contains 603 molecules with a validated

activity against coronaviruses or human receptors (Yang et al., 2021). The database has a web interface that allows for the screening of novel ligands to predict their potential to affect one of the main targets of SARS-CoV-2, namely, the 3CLpro and the PLpro. The activity prediction is performed through a direct assessment of the 2D or 3D similarity of a target molecule to the database elements. In this context, we have deployed important efforts in collecting and curating a dataset that can serve in training and validating different ML and DL approaches in tackling the search for therapeutics against SARS-CoV-2. Our dataset is larger than the D3Similarity dataset and yet ready for use in ML/DL applications against SARS-CoV-2. Conversely, it does not account for quantitative activity information.

It seems important to engage further efforts to integrate more information in our dataset toward its use for a quantitative prediction of molecules activity. Moreover, a deeper analysis of the dataset content may reveal important knowledge for DD projects. Further tuning of the dataset will aim to integrate valuable knowledge on what to expect from effective anti-SARS-CoV-2 molecules (Tummino et al., 2021). In fact, it has been demonstrated that the cationic amphiphilic nature of some drugs may induce phospholipidosis rather than actual antiviral effects (Tummino et al., 2021). Such properties should be further examined to enhance the relevance of our dataset to the development of COVID-19 therapeutics.

5 CONCLUSION

In the present study, we collected and curated a dedicated dataset of 2,610 molecules having anticoronavirus effects. This valuable resource was formatted and used to perform different simulations and optimization of eleven ML and DL algorithms toward the classification of molecules into active and inactive classes. We were able to obtain three highly accurate classifiers that were validated through cross-validation and on an external set of data. The DL algorithms demonstrated the best performances.

DATA AVAILABILITY STATEMENT

The datasets presented in this study along with the jupyter notebooks can be found online on https://github.com/Harigua/ML_DD-applications/tree/main/COVID-19.

REFERENCES

- Achdout, H., Aimon, A., Bar-David, E., Barr, H., Ben-Shmuel, A., Bennett, J., et al. (2020). Covid Moonshot: Open Science Discovery of Sars-Cov-2 Main Protease Inhibitors by Combining Crowdsourcing, High-Throughput Experiments, Computational Simulations, and Machine Learning. New York: BioRxiv.
- Aliper, A., Plis, S., Artemov, A., Ulloa, A., Mamoshina, P., and Zhavoronkov, A. (2016). Deep Learning Applications for Predicting Pharmacological Properties of Drugs and Drug Repurposing Using Transcriptomic Data. *Mol. Pharmaceutics* 13, 2524–2530. doi:10.1021/acs.molpharmaceut.6b00248

AUTHOR CONTRIBUTIONS

EH-S conceived the research. EH-S and IA-T designed the experiments. EH-S, OS, and YA collected and curated the data. EH-S and MH implemented the code, tested the performances, and generated the figures. EH-S analyzed the results and drafted the original manuscript. EH-S, MH, IA-T, OS, and IG reviewed and edited the manuscript. All authors read and approved the final manuscript.

FUNDING

EH-S is a recipient of a NAS grant within the USAID PEER Women Mentoring Programme, Grant Award Number AID-OAA-A-11-00012. EH-S is also a recipient of the “Cov2-Anti-Proteases” project funded by Institut Pasteur-Paris, “Levée de Fond-Urgence COVID-19.” EH-S and IG are financially supported by the programs of the Ministry of Higher education and Research of the Republic of Tunisia.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/fgene.2021.744170/full#supplementary-material>

Supplementary Figure S1 | Performances of the three algorithms GCN, DAG and RF assessed on homogeneous vs. heterogeneous data. (a) Performances for the subsets of the category 3CLpro. (b) Performances for the subset of the category PLpro. (c) Comparison between the models performances on the mixed dataset and the 3CLpro subset.

Supplementary Table S1 | Statistics on the anticoronavirus dataset according to their origin (bioassays vs. literature) and their type (experiments, targets, etc). Composition of the training, validation and test set in terms of active and inactive molecules are indicated.

Supplementary Table S2 | Performances of all 11 algorithms using different splitting ratios (80/10/10 vs. 60/20/20) and methods (random vs. scaffold split) on different datasets (heterogeneous vs. homogeneous).

Supplementary Table S3 | Hyperparameters' tuning and optimization for the three best performers: GCN, DAG and RF.

Supplementary Table S4 | Performances of the optimized algorithms GCN, DAG and RF in terms of accuracy, F1-score, sensitivity and specificity, on different subsets of the dataset.

Supplementary Table S5 | Predictions assessment of the three algorithms GCN, DAG and RF on the subset of experimentally validated molecules (sheet 1). Sheets 2-4 contain the prediction outcomes for each algorithm for all molecules.

- Altae-Tran, H., Ramsundar, B., Pappu, A. S., and Pande, V. (2017). Low Data Drug Discovery with One-Shot Learning. *ACS Cent. Sci.* 3, 283–293. doi:10.1021/acscentsci.6b00367
- Bero, S. A., Muda, A. K., Choo, Y. H., Muda, N. A., and Pratama, S. F. (2017). Similarity Measure for Molecular Structure: a Brief Review. *J. Phys. Conf. Ser.* 892, 012015. doi:10.1088/1742-6596/892/1/012015
- Bung, N., Krishnan, S. R., Bulusu, G., and Roy, A. (2021). De Novo design of New Chemical Entities for Sars-Cov-2 Using Artificial Intelligence. *Future Med. Chem.* 13, 575–585. doi:10.4155/fmc-2020-0262
- Cano, G., Garcia-Rodriguez, J., Garcia-Garcia, A., Perez-Sanchez, H., Benediktsson, J. A., Thapa, A., et al. (2017). Automatic Selection of Molecular Descriptors Using Random forest: Application to Drug

- Discovery. *Expert Syst. Appl.* 72, 151–159. doi:10.1016/j.eswa.2016.12.008
- Chellapandi, P., and Saranya, S. (2020). Genomics Insights of Sars-Cov-2 (Covid-19) into Target-Based Drug Discovery. *Med. Chem. Res.* 31, 1–15. doi:10.1007/s00044-020-02610-8
- Chung, N. C., Miasojedow, B., Startek, M., and Gambin, A. (2019). Jaccard/tanimoto Similarity Test and Estimation Methods for Biological Presence-Absence Data. *BMC bioinformatics* 20, 644–711. doi:10.1186/s12859-019-3118-5
- David, L., Thakkar, A., Mercado, R., and Engkvist, O. (2020). Molecular Representations in Ai-Driven Drug Discovery: a Review and Practical Guide. *J. Cheminform.* 12, 56–22. doi:10.1186/s13321-020-00460-5
- Dietterich, T. G. (2000). “Ensemble Methods in Machine Learning,” in International Workshop on Multiple Classifier Systems (Springer), 1–15. doi:10.1007/3-540-45014-9_1
- Dragojevic Simic, V., Miljkovic, M., Stamenkovic, D., Vekic, B., Ratkovic, N., Simic, R., et al. (2021). An Overview of Antiviral Strategies for Coronavirus 2 (Sars-cov-2) Infection with Special Reference to Antimalarial Drugs Chloroquine and Hydroxychloroquine. *Int. J. Clin. Pract.* 75, e13825. doi:10.1111/ijcp.13825
- Duran-Frigola, M., Fernández-Torras, A., Bertoni, M., and Aloy, P. (2019). Formatting Biological Big Data for Modern Machine Learning in Drug Discovery. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* 9, e1408. doi:10.1002/wcms.1408
- Duvenaud, D., Maclaurin, D., Aguilera-Iparraguirre, J., Gómez-Bombarelli, R., Hirzel, T., Aspuru-Guzik, A., et al. (2015). *Convolutional Networks on Graphs for Learning Molecular Fingerprints*. New York: arXiv preprint arXiv:1509.09292.
- Erickson, S. S., Wu, H., Zhang, H., Michael, L. A., Newton, M. A., Hoffmann, F. M., et al. (2017). Machine Learning Consensus Scoring Improves Performance across Targets in Structure-Based Virtual Screening. *J. Chem. Inf. Model.* 57, 1579–1590. doi:10.1021/acs.jcim.7b00153
- Filippov, I. V., and Nicklaus, M. C. (2009). Optical Structure Recognition Software to Recover Chemical Information: OSRA, an Open Source Solution. *J. Chem. Inf. Model.* 49, 740–743. doi:10.1021/ci800067r
- Galan, L. E. B., Santos, N. M. d., Asato, M. S., Araújo, J. V., de Lima Moreira, A., Araújo, A. M. M., et al. (2021). Phase 2 Randomized Study on Chloroquine, Hydroxychloroquine or Ivermectin in Hospitalized Patients with Severe Manifestations of Sars-Cov-2 Infection. *Pathog. Glob. Health* 115, 235–242. doi:10.1080/20477724.2021.1890887
- Gaulton, A., Bellis, L. J., Bento, A. P., Chambers, J., Davies, M., Hersey, A., et al. (2012). ChEMBL: a Large-Scale Bioactivity Database for Drug Discovery. *Nucleic Acids Res.* 40, D1100–D1107. doi:10.1093/nar/gkr777
- Gfeller, D., Grosdidier, A., Wirth, M., Daina, A., Michielin, O., and Zoete, V. (2014). Swisstargetprediction: a Web Server for Target Prediction of Bioactive Small Molecules. *Nucleic Acids Res.* 42, W32–W38. doi:10.1093/nar/gku293
- Gupta, R., Srivastava, D., Sahu, M., Tiwari, S., Ambasta, R. K., and Kumar, P. (2021). Artificial Intelligence to Deep Learning: Machine Intelligence Approach for Drug Discovery. *Mol. Divers.* 25, 1–46. doi:10.1007/s11030-021-10217-3
- Heikamp, K., and Bajorath, J. (2014). Support Vector Machines for Drug Discovery. *Expert Opin. Drug Discov.* 9, 93–104. doi:10.1517/17460441.2014.866943
- Hoffmann, M., Mösbauer, K., Hofmann-Winkler, H., Kaul, A., Kleine-Weber, H., Krüger, N., et al. (2020). Chloroquine Does Not Inhibit Infection of Human Lung Cells with Sars-Cov-2. *Nature* 585, 588–590. doi:10.1038/s41586-020-2575-3
- Irwin, B. W. J., Levell, J. R., Whitehead, T. M., Segall, M. D., and Conduit, G. J. (2020). Practical Applications of Deep Learning to Impute Heterogeneous Drug Discovery Data. *J. Chem. Inf. Model.* 60, 2848–2857. doi:10.1021/acs.jcim.0c00443
- Jing, Y., Bian, Y., Hu, Z., Wang, L., and Xie, X. Q. (2018). Deep Learning for Drug Design: an Artificial Intelligence Paradigm for Drug Discovery in the Big Data Era. *AAPS J.* 20, 58–10. doi:10.1208/s12248-018-0210-0
- Kearnes, S., McCloskey, K., Berndl, M., Pande, V., and Riley, P. (2016). Molecular Graph Convolutions: Moving beyond Fingerprints. *J. Comput. Aided Mol. Des.* 30, 595–608. doi:10.1007/s10822-016-9938-8
- Kelleni, M. T. (2021). Tocilizumab, Remdesivir, Favipiravir, and Dexamethasone Repurposed for Covid-19: A Comprehensive Clinical and Pharmacovigilant Reassessment. *SN Compr. Clin. Med.* 3, 919–923. doi:10.1007/s42399-021-00824-4
- Keshavarzi Arshadi, A., Webb, J., Salem, M., Cruz, E., Calad-Thomson, S., Ghadrian, N., et al. (2020). Artificial Intelligence for Covid-19 Drug Discovery and Vaccine Development. *Front. Artif. Intell.* 3, 65. doi:10.3389/frai.2020.00065
- Kim, S., Chen, J., Gindulyte, A., He, A., He, S., Li, Q., et al. (2020). PubChem in 2021: New Data Content and Improved Web Interfaces. *Nucleic Acids Res.* 49, D1388–D1395. doi:10.1093/nar/gkaa971
- Kipf, T. N., and Welling, M. (2016). *Semi-supervised Classification with Graph Convolutional Networks*. New York: arXiv preprint arXiv:1609.02907.
- Korkmaz, S. (2020). Deep Learning-Based Imbalanced Data Classification for Drug Discovery. *J. Chem. Inf. Model.* 60, 4180–4190. doi:10.1021/acs.jcim.9b01162
- Lavecchia, A. (2019). Deep Learning in Drug Discovery: Opportunities, Challenges and Future Prospects. *Drug Discov. Today* 24, 2017–2032. doi:10.1016/j.drudis.2019.07.006
- Le, N. Q. K., and Huynh, T.-T. (2019). Identifying Snares by Incorporating Deep Learning Architecture and Amino Acid Embedding Representation. *Front. Physiol.* 10, 1501. doi:10.3389/fphys.2019.01501
- Le, N. Q. K., and Nguyen, V.-N. (2019). Snare-cnn: a 2d Convolutional Neural Network Architecture to Identify Snare Proteins from High-Throughput Sequencing Data. *PeerJ Comp. Sci.* 5, e177. doi:10.7717/peerj-cs.177
- Le, N. Q. K., Yapp, E. K. Y., Nagasundaram, N., and Yeh, H.-Y. (2019). Classifying Promoters by Interpreting the Hidden Information of Dna Sequences via Deep Learning and Combination of Continuous Fasttext N-Grams. *Front. Bioeng. Biotechnol.* 7, 305. doi:10.3389/fbioe.2019.00305
- Lenselink, E. B., Ten Dijke, N., Bongers, B., Papadatos, G., Van Vlijmen, H. W. T., Kowalczyk, W., et al. (2017). Beyond the Hype: Deep Neural Networks Outperform Established Methods Using a ChEMBL Bioactivity Benchmark Set. *J. Cheminform.* 9, 45–14. doi:10.1186/s13321-017-0232-0
- Li, X., Yan, X., Gu, Q., Zhou, H., Wu, D., and Xu, J. (2019). Deepchemstable: Chemical Stability Prediction with an Attention-Based Graph Convolution Network. *J. Chem. Inf. Model.* 59, 1044–1049. doi:10.1021/acs.jcim.8b00672
- Liu, Y., Wu, Y., Shen, X., and Xie, L. (2021). Covid-19 Multi-Targeted Drug Repurposing Using Few-Shot Learning. *Front. Bioinformatics* 1, 18. doi:10.3389/fbinf.2021.693177
- Lo, Y.-C., Rensi, S. E., Torng, W., and Altman, R. B. (2018). Machine Learning in Chemoinformatics and Drug Discovery. *Drug Discov. Today* 23, 1538–1546. doi:10.1016/j.drudis.2018.05.010
- Lusci, A., Pollastri, G., and Baldi, P. (2013). Deep Architectures and Deep Learning in Chemoinformatics: the Prediction of Aqueous Solubility for Drug-like Molecules. *J. Chem. Inf. Model.* 53, 1563–1575. doi:10.1021/ci400187y
- Matthews, B. W. (1975). Comparison of the Predicted and Observed Secondary Structure of T4 Phage Lysozyme. *Biochim. Biophys. Acta (Bba) - Protein Struct.* 405, 442–451. doi:10.1016/0005-2795(75)90109-9
- Micheli, A. (2009). Neural Network for Graphs: A Contextual Constructive Approach. *IEEE Trans. Neural Netw.* 20, 498–511. doi:10.1109/tnn.2008.2010350
- Moiseev, S., Avdeev, S., Brovko, M., Novikov, P., and Fomin, V. (2021). Is There a Future for Hydroxychloroquine/chloroquine in Prevention of Sars-Cov-2 Infection (Covid-19)? *Ann. Rheum. Dis.* 80, e19. doi:10.1136/annrheumdis-2020-217570
- Pastick, K. A., Okafor, E. C., Wang, F., Lofgren, S. M., Skipper, C. P., Nicol, M. R., et al. (2020). “Hydroxychloroquine and Chloroquine for Treatment of Sars-Cov-2 (Covid-19),” in *Open Forum Infectious Diseases* (Oxford University Press US), ofaa130.
- Paul, D., Sanap, G., Shenoy, S., Kalyane, D., Kalia, K., and Tekade, R. K. (2020). *Artificial Intelligence in Drug Discovery and Development*. Amsterdam: Drug Discovery Today.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine Learning in python. *J. machine Learn. Res.* 12, 2825–2830.
- Perualila-Tan, N. J., Shkedy, Z., Talloen, W., Göhlmann, H. W. H., Moerbeke, M. V., Kasim, A., et al. (2016). Weighted Similarity-Based Clustering of Chemical Structures and Bioactivity Data in Early Drug Discovery. *J. Bioinform. Comput. Biol.* 14, 1650018. doi:10.1142/s0219720016500189

- Pillaiyar, T., Meenakshisundaram, S., and Manickam, M. (2020). Recent Discovery and Development of Inhibitors Targeting Coronaviruses. *Drug Discov. Today* 25, 668–688. doi:10.1016/j.drudis.2020.01.015
- Ramsundar, B., Eastman, P., Walters, P., and Pande, V. (2019). *Deep Learning for the Life Sciences: Applying Deep Learning to Genomics, Microscopy, Drug Discovery, and More*. Sebastopol, CA: "O'Reilly Media, Inc."
- Ramsundar, B., Liu, B., Wu, Z., Verras, A., Tudor, M., Sheridan, R. P., et al. (2017). Is Multitask Deep Learning Practical for Pharma? *J. Chem. Inf. Model.* 57, 2068–2076. doi:10.1021/acs.jcim.7b00146
- Rifaioğlu, A. S., Atas, H., Martin, M. J., Cetin-Atalay, R., Atalay, V., and Doğan, T. (2019). Recent Applications of Deep Learning and Machine Intelligence on In Silico Drug Discovery: Methods, Tools and Databases. *Brief. Bioinformatics* 20, 1878–1912. doi:10.1093/bib/bby061
- Rogers, D., and Hahn, M. (2010). Extended-connectivity Fingerprints. *J. Chem. Inf. Model.* 50, 742–754. doi:10.1021/ci100050t
- Shi, J.-Y., Yiu, S.-M., Li, Y., Leung, H. C. M., and Chin, F. Y. L. (2015). Predicting Drug-Target Interaction for New Drugs Using Enhanced Similarity Measures and Super-target Clustering. *Methods* 83, 98–104. doi:10.1016/j.ymeth.2015.04.036
- Song, L. G., Xie, Q. X., Lao, H. L., and Lv, Z. Y. (2021). Human Coronaviruses and Therapeutic Drug Discovery. *Infect. Dis. Poverty* 10, 28–21. doi:10.1186/s40249-021-00812-9
- Sun, K., Lin, Z., and Zhu, Z. (2020). Multi-stage Self-Supervised Learning for Graph Convolutional Networks on Graphs with Few Labeled Nodes. *Aaai* 34, 5892–5899. doi:10.1609/aaai.v34i04.6048
- Trezza, A., Iovinelli, D., Santucci, A., Prisci, F., and Spiga, O. (2020). An Integrated Drug Repurposing Strategy for the Rapid Identification of Potential Sars-Cov-2 Viral Inhibitors. *Sci. Rep.* 10, 13866–13868. doi:10.1038/s41598-020-70863-9
- Tummino, T. A., Rezeli, V. V., Fischer, B., Fischer, A., O'Meara, M. J., Monel, B., et al. (2021). Drug-induced Phospholipidosis Confounds Drug Repurposing for Sars-Cov-2. *Science* 373, 1. doi:10.1126/science.abi4708
- Unterthiner, T., Mayr, A., Klambauer, G., Steijaert, M., Wegner, J. K., Ceulemans, H., et al. (2014). Deep Learning as an Opportunity in Virtual Screening. *Proc. deep Learn. Workshop NIPS* 27, 1–9.
- Vamathevan, J., Clark, D., Czodrowski, P., Dunham, I., Ferran, E., Lee, G., et al. (2019). Applications of Machine Learning in Drug Discovery and Development. *Nat. Rev. Drug Discov.* 18, 463–477. doi:10.1038/s41573-019-0024-5
- Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. (2018). Graph Attention Networks. *Stat* 1050, 4.
- Vincent, M. J., Bergeron, E., Benjannet, S., Erickson, B. R., Rollin, P. E., Ksiazek, T. G., et al. (2005). Chloroquine Is a Potent Inhibitor of Sars Coronavirus Infection and Spread. *Virology* 339, 69–78. doi:10.1016/j.virol.2005.07.019
- Walters, W. P., and Barzilay, R. (2021). Critical Assessment of AI in Drug Discovery. *Expert Opin. Drug Discov.* 16, 1–11. doi:10.1080/17460441.2021.1915982
- Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. S. (2020). A Comprehensive Survey on Graph Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.* 32, 4–24. doi:10.1109/TNNLS.2020.2978386
- Wu, Z., Ramsundar, B., Feinberg, E. N., Gomes, J., Geniesse, C., Pappu, A. S., et al. (2018). Moleculenet: a Benchmark for Molecular Machine Learning. *Chem. Sci.* 9, 513–530. doi:10.1039/c7sc02664a
- Yang, J., Shen, C., and Huang, N. (2020). Predicting or Pretending: Artificial Intelligence for Protein-Ligand Interactions Lack of Sufficiently Large and Unbiased Datasets. *Front. Pharmacol.* 11, 69. doi:10.3389/fphar.2020.00069
- Yang, Y., Zhu, Z., Wang, X., Zhang, X., Mu, K., Shi, Y., et al. (2021). Ligand-based Approach for Predicting Drug Targets and for Virtual Screening against Covid-19. *Brief. Bioinform.* 22, 1053–1064. doi:10.1093/bib/bbaa422
- Yao, X., Ye, F., Zhang, M., Cui, C., Huang, B., Niu, P., et al. (2020). In Vitro antiviral Activity and Projection of Optimized Dosing Design of Hydroxychloroquine for the Treatment of Severe Acute Respiratory Syndrome Coronavirus 2 (Sars-cov-2). *Clin. Infect. Dis.* 71, 732–739. doi:10.1093/cid/ciaa237
- Zeng, X., Zhu, S., Liu, X., Zhou, Y., Nussinov, R., and Cheng, F. (2019). Deepdr: a Network-Based Deep Learning Approach to In Silico Drug Repositioning. *Bioinformatics* 35, 5191–5198. doi:10.1093/bioinformatics/btz418
- Zhai, T., Zhang, F., Haider, S., Kraut, D., and Huang, Z. (2021). An Integrated Computational and Experimental Approach to Identifying Inhibitors for Sars-Cov-2 3cl Protease. *Front. Mol. Biosciences* 8, 267. doi:10.3389/fmolb.2021.661424
- Zhang, H., Saravanan, K. M., Yang, Y., Hossain, M. T., Li, J., Ren, X., et al. (2020). Deep Learning Based Drug Screening for Novel Coronavirus 2019-ncov. *Interdiscip. Sci. Comput. Life Sci.* 12, 368–376. doi:10.1007/s12539-020-00376-6
- Zhang, S., Tong, H., Xu, J., and Maciejewski, R. (2019). Graph Convolutional Networks: a Comprehensive Review. *Comput. Soc. Networks* 6, 1–23. doi:10.1186/s40649-019-0069-y
- Zitnik, M., Agrawal, M., and Leskovec, J. (2018). Modeling Polypharmacy Side Effects with Graph Convolutional Networks. *Bioinformatics* 34, i457–i466. doi:10.1093/bioinformatics/bty294

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article or claim that may be made by its manufacturer is not guaranteed or endorsed by the publisher.

Copyright © 2021 Harigua-Souiai, Heinhane, Abdelkrim, Souiai, Abdeljaoued-Tej and Guizani. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

Advantages of publishing in Frontiers



OPEN ACCESS

Articles are free to read
for greatest visibility
and readership



FAST PUBLICATION

Around 90 days
from submission
to decision



HIGH QUALITY PEER-REVIEW

Rigorous, collaborative,
and constructive
peer-review



TRANSPARENT PEER-REVIEW

Editors and reviewers
acknowledged by name
on published articles

Frontiers

Avenue du Tribunal-Fédéral 34
1005 Lausanne | Switzerland

Visit us: www.frontiersin.org

Contact us: frontiersin.org/about/contact



REPRODUCIBILITY OF RESEARCH

Support open data
and methods to enhance
research reproducibility



DIGITAL PUBLISHING

Articles designed
for optimal readership
across devices



FOLLOW US

@frontiersin



IMPACT METRICS

Advanced article metrics
track visibility across
digital media



EXTENSIVE PROMOTION

Marketing
and promotion
of impactful research



LOOP RESEARCH NETWORK

Our network
increases your
article's readership