frontiers | Research Topics

AI and data science in drug development and public health: Highlights from the MCBIOS 2022 conference

Edited by
Ramin Homayouni, Prashanti Manda, Aik Choon Tan and Zhaohui Steve Qin

Published in
Frontiers in Artificial Intelligence
Frontiers in Big Data

ISSN 1664-8714
ISBN 978-2-83251-891-5
DOI 10.3389/978-2-83251-891-5

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

AI and data science in drug development and public health: Highlights from the MCBIOS 2022 conference

Topic editors

Ramin Homayouni — Oakland University William Beaumont School of Medicine, United States

Prashanti Manda — University of North Carolina at Greensboro, United States

Aik Choon Tan — The University of Utah, United States

Zhaohui Steve Qin — Emory University, United States

Citation

Homayouni, R., Manda, P., Tan, A. C., Qin, Z. S., eds. (2023). *AI and data science in drug development and public health: Highlights from the MCBIOS 2022 conference*. Lausanne: Frontiers Media SA. doi: 10.3389/978-2-83251-891-5

Table of contents

frontiers | Frontiers in Big Data

TYPE Editorial
PUBLISHED 27 February 2023
DOI 10.3389/fdata.2023.1156811

OPEN ACCESS

EDITED AND REVIEWED BY
Thomas Hartung,
Bloomberg School of Public Health, Johns Hopkins University, United States

*CORRESPONDENCE
Ramin Homayouni
✉ rhomayouni@oakland.edu

SPECIALTY SECTION
This article was submitted to Medicine and Public Health, a section of the journal Frontiers in Big Data

RECEIVED 01 February 2023
ACCEPTED 13 February 2023
PUBLISHED 27 February 2023

CITATION
Homayouni R, Manda P, Tan AC and Qin ZS (2023) Editorial: AI and data science in drug development and public health: Highlights from the MCBIOS 2022 conference. *Front. Big Data* 6:1156811. doi: 10.3389/fdata.2023.1156811

Editorial: AI and data science in drug development and public health: Highlights from the MCBIOS 2022 conference

Ramin Homayouni[1]*, Prashanti Manda[2], Aik Choon Tan[3] and Zhaohui S. Qin[4]

[1]Department of Foundational Medical Studies, Oakland University William Beaumont School of Medicine, Rochester, MI, United States, [2]Department of Informatics and Analytics, University of North Carolina at Greensboro, Greensboro, NC, United States, [3]Department of Oncological Sciences and Biomedical Informatics, Huntsman Cancer Institute, Salt Lake City, UT, United States, [4]Department of Biostatistics and Bioinformatics, Rollins School of Public Health, Emory University, Atlanta, GA, United States

KEYWORDS

deep learning, NLP, genomics (G), regulatory science research, metagenomics

Editorial on the Research Topic
AI and data science in drug development and public health: Highlights from the MCBIOS 2022 conference

This Research Topic is a product of the 18th annual conference of the MidSouth Computational Biology and Bioinformatics Society (MCBIOS), which has a broad membership of scientists and trainees with research interests in genomics, medicine, and regulatory sciences. The topic includes a total of nine papers appearing in *Frontiers in Artificial Intelligence* (Medicine and Public Health), which include five original research articles, two methods articles, one brief research report and one review article. The papers can be categorized into four general themes of regulatory sciences, genomics, protein modeling and natural language processing, as detailed below.

Regulatory science

The field of Artificial Intelligence (AI) has advanced significantly during the past few years, but its application to biomedical research, healthcare and regulatory sciences is still emerging. In particular, application of AI tools in regulatory decision-making and for drug safety and efficacy is not widely accepted, in part due to the perception that larger amount of data are needed to train accurate AI models. In their review article, Connor et al. challenge this perception with respect to adaptability of AI models on unseen data, focusing on evaluation of DeepDILI for predicting drug-induced liver injury (DILI). They concluded that the target test set plays a major role in assessing the adaptive behavior of AI models, but the training set does not significantly affect the predictive performance of the adaptive model.

Bisgin et al. demonstrated the use of machine learning in screening for food-contaminating beetles, which currently requires manual microscopic examination. They developed a convolutional neural network (CNN) model trained on high-quality elytral (hardened forewing) images to predict 27 different species of pantry beetles. The model achieved an average accuracy of ~90%. However, several species fell below that

average accuracy due to significant intraspecies variation of elytral pattern. This represents an improvement over previous models which will eventually lead to their goal of automated species identification of food-contaminating beetles.

Genomics

A major challenge in metagenomics is the identification and classification of bacteria in microbial communities that may consist of thousands microbial species. To address this Research Topic, Pham et al. developed a computationally efficient method by using compressed and low-sized genomic signatures of the bacteria to be classified. A modified Bloom filter is used to store k-mers with hash values corresponding to each bacterial species. They showed that most bacteria in many microbiomes can be represented uniquely using the proposed genomic signatures.

As the amount of genome sequencing data increases in the public databases, scalable methods are needed for efficient variant annotation and classification tasks. Li et al. described an updated version of SNPAAMapper, a variant annotation pipeline, with much improved computational efficiency on most updated information. This new version of the SNPAAMapper not only runs faster and more efficiently, it can also classify variants by type of genomic regions (Coding Sequence, Untranslated Regions, upstream, downstream, and intron), predict types of amino acid changes (missense, nonsense, etc.), and prioritize mutation effects (e.g., non-synonymous, synonymous).

Genotype imputation is an important aspect of genome-wide association studies (GWAS). Although deep learning (DL)-based methods have already been developed for this task, it is still challenging to optimize the learning process in DL-based methods in order to achieve high imputation accuracy. Song et al. developed a convolutional autoencoder (AE) model for genotype imputation. Additionally, they implemented a customized training loop by modifying the training process with a single batch loss rather than the average loss over batches. This modified AE-based imputation method was carefully evaluated using multiple real datasets. They found that the modified AE imputation method achieved comparable or better performance than the existing DL-based methods.

Gene prioritization based on molecular function is an important step in utilizing—omics data for understanding human diseases. Nguyen et al. presented a new tool called WINNER for characterizing and prioritizing biomolecules. The tool takes molecular interaction data and expands the network while ranking all nodes by their relevance to other network nodes. These networks can be used to evaluate candidate genes for diseases or proteins from high throughput experiments. The utility of WINNER was evaluated on several diseases such as Alzheimer's disease, breast cancer, myocardial infarctions, and Triple negative breast cancer.

Protein modeling

Protein structure-function analysis is important for understanding ligand binding properties of proteins as well as for developing new drugs. However, the crystal structures of many proteins are not available in public databases. In one such case, Gokulan et al. modeled VirD4 ATPase, a component of the bacterial type IV secretory system using a variety of bioinformatics and computational tools. The authors hypothesized that the unique insertion regions found in the VirD4 protein could play a role in the flexible movement of the hexameric unit during the relaxosome processing or transfer of the substrate.

Natural language processing

Machine learning approaches to utilize the vast amount of unstructured text have made tremendous progress in recent years. For example, a graph embedding-based method (MedGraph) was developed by Ebeid to provide a semantic relevance retrieval ranking for biomedical literature indexed in PubMed. Using objective metrics, this a proof-of-concept study provides evidence that graph modeling provides better search relevance than traditional methods.

A fundamental challenge in any social, behavioral or biological study is determination of causality. Further, assessing causality from unstructured text is manual and time-consuming. In their paper, Wang et al. describe a general causal fame work named DeepCausality, which incorporates AI-powered language models, named entity recognition and Judea Pearl's Do-calculus to fulfill different domain-specific applications. They evaluated their method using the LiverTox database to estimate drug-induced liver toxicity (DILI) and validating their results against the American College of Gastroenterology clinical guidelines.

Overall, the papers selected for this Research Topic represent the breadth of computational methods and applications in biomedical and regulatory sciences at the annual MCBIOS conference.

Author contributions

RH, PM, AT, and ZQ drafted the manuscript. All authors contributed to the article and approved the submitted version.

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

frontiers | Frontiers in Artificial Intelligence

TYPE Original Research
PUBLISHED 12 August 2022
DOI 10.3389/frai.2022.952424

OPEN ACCESS

EDITED BY
Aik Choon Tan,
Moffitt Cancer Center, United States

REVIEWED BY
Nisha Pillai,
Mississippi State University, United States
Omid Memarian Sorkhabi,
University of Isfahan, Iran

*CORRESPONDENCE
Joshua Xu
Joshua.Xu@fda.hhs.gov

†These authors have contributed equally to this work and share first authorship

SPECIALTY SECTION
This article was submitted to Medicine and Public Health, a section of the journal Frontiers in Artificial Intelligence

RECEIVED 25 May 2022
ACCEPTED 22 July 2022
PUBLISHED 12 August 2022

CITATION
Bisgin H, Bera T, Wu L, Ding H, Bisgin N, Liu Z, Pava-Ripoll M, Barnes A, Campbell JF, Vyas H, Furlanello C, Tong W and Xu J (2022) Accurate species identification of food-contaminating beetles with quality-improved elytral images and deep learning.
Front. Artif. Intell. 5:952424.
doi: 10.3389/frai.2022.952424

Accurate species identification of food-contaminating beetles with quality-improved elytral images and deep learning

Halil Bisgin[1†], Tanmay Bera[2†], Leihong Wu[2], Hongjian Ding[3], Neslihan Bisgin[1], Zhichao Liu[2], Monica Pava-Ripoll[4], Amy Barnes[3], James F. Campbell[5], Himansi Vyas[3], Cesare Furlanello[6], Weida Tong[2] and Joshua Xu[2]*

[1]Department of Mathematics and Applied Sciences, University of Michigan-Flint, Flint, MI, United States, [2]Division of Bioinformatics and Biostatistics, National Center for Toxicological Research, US Food and Drug Administration, Jefferson, AR, United States, [3]Food Chemistry Lab 1, Arkansas Regional Laboratory, Office of Regulatory Affairs, US Food and Drug Administration, Jefferson, AR, United States, [4]Office for Food Safety, Center for Food Safety and Applied Nutrition, US Food and Drug Administration, College Park, MD, United States, [5]Stored Product Insect and Engineering Research Unit, US Department of Agriculture, Manhattan, KS, United States, [6]HK3 Lab, Milan, Italy

Food samples are routinely screened for food-contaminating beetles (i.e., pantry beetles) due to their adverse impact on the economy, environment, public health and safety. If found, their remains are subsequently analyzed to identify the species responsible for the contamination; each species poses different levels of risk, requiring different regulatory and management steps. At present, this identification is done through manual microscopic examination since each species of beetle has a unique pattern on its elytra (hardened forewing). Our study sought to automate the pattern recognition process through machine learning. Such automation will enable more efficient identification of pantry beetle species and could potentially be scaled up and implemented across various analysis centers in a consistent manner. In our earlier studies, we demonstrated that automated species identification of pantry beetles is feasible through elytral pattern recognition. Due to poor image quality, however, we failed to achieve prediction accuracies of more than 80%. Subsequently, we modified the traditional imaging technique, allowing us to acquire high-quality elytral images. In this study, we explored whether high-quality elytral images can truly achieve near-perfect prediction accuracies for 27 different species of pantry beetles. To test this hypothesis, we developed a convolutional neural network (CNN) model and compared performance between two different image sets for various pantry beetles. Our study indicates improved image quality indeed leads to better prediction accuracy; however, it was not the only requirement for achieving good

accuracy. Also required are many high-quality images, especially for species with a high number of variations in their elytral patterns. The current study provided a direction toward achieving our ultimate goal of automated species identification through elytral pattern recognition.

KEYWORDS

food-contaminating beetle, species identification, deep learning, convolutional neural networks, machine learning, food safety, image classification

Introduction

A large group of nuisance insects that contaminate grains and other food items are commonly termed pantry beetles (Bell, 2013). They are notorious for spoiling stored grain and processed food products, leading to significant economic damage (Belluco et al., 2013). Some of these pantry beetles are aggressively invasive and can cause damage to local agriculture and ecological insects if they spread through the transportation of contaminated food products (Heeps, 2016). Some of the pests also pose a serious threat to public health, as they are active carriers of pathogens (Olsen et al., 2001).

To counter such adversities, food grains and products are monitored and routinely screened for pantry beetles or their remains (Bell, 2013; Belluco et al., 2013). The most common and widely-used method involves highly-trained analysts manually screening food samples for insect remains using optical microscopes. Any insect or insect remains found are then scrutinized using a comparison optical microscope to match the patterns from the insect fragments with reference images to identify the exact insect species, genus, or family. This identification step is crucial, as each species poses different threat levels and their contamination may require different methods of management and regulatory procedures. Currently, no reliable alternatives to the manual screening method are available, as spectroscopic or PCR-based detection techniques have remained challenging for this application. Moreover, due to the manual nature of the microanalysis, the current method is highly dependent on the experience and expertise of the individual analyst, making it more susceptible to human error and higher variation across institutions. Also, manual methods are difficult to scale up, hindering the screening of a larger number of samples in a shorter time frame, especially in the absence of experienced and dexterous analysts.

Species identification through image analysis has been explored for efficient taxonomical and environmental applications for several years (Norouzzadeh et al., 2018; Terry et al., 2019; Høye et al., 2020). These computer-aided applications have tried to address a wide range of problems from food safety to identification of insect pests (Daly et al., 1982; Weeks et al., 1997; O'Neill et al., 2000; Larios et al., 2008; Yalcin, 2015). With the advent of machine learning methods, image-based species identification has gained further momentum and well-known discriminative models such as support vector machines (SVM) (Cortes and Vapnik, 1995) and generative models have been widely adopted for insect classification (Martineau et al., 2017). Examples of these models include, but are not limited to: insect or pest identification using SVM (Qing et al., 2012; Wang et al., 2012; Yang et al., 2015), honeybee and moth identification with decision trees (Mayo and Watson, 2007; da Silva et al., 2015), and red palm weevil and insect recognition systems through neural networks (Al-Saqer and Hassan, 2011; Wang et al., 2012). With increasing computational power, more complex neural network architectures, i.e., deep learning (DL) approaches have recently helped in tackling more challenging tasks in the field of food and agricultural science (Lee et al., 2015; DeChant et al., 2017; Lu et al., 2017; Zhang et al., 2018). Although there have been relatively fewer DL studies to identify filth elements for food contamination (Reinholds et al., 2015; Bansal et al., 2017), variations of DL designs such as Region-based Fully Convolutional Network (R-FCN), convolutional block attention module (CBAM), convolutional neural network (CNN) and pre-trained models have shown promising performances for pest, stored-grain insect, and fly classification (Chen et al., 2020; Kuzuhara et al., 2020; Shi et al., 2020). The DL models have not only achieved high classification accuracies, but also offered a new way of feature extraction embedded in the process as an alternative to conventional features such as domain-dependent, global, local, and mid-level features (Martineau et al., 2017).

We have also investigated similar approaches, i.e., machine learning techniques, with the aim of automating the identification process of pantry beetles whose elytra (hardened forewing) have unique patterns that can be considered as fingerprints or features. In a previous study, we demonstrated that a specific pantry beetle species could indeed be identified through elytral pattern recognition using machine learning (Martin et al., 2016). In our subsequent study, we observed that classical machine learning techniques such as artificial neural network (ANN) and SVM could be used for this application (Bisgin et al., 2018). However, optimized ANN and SVM models yielded about 80 and 85% of average accuracies, respectively. We observed that some species consistently performed less than others; which could be attributed to their misidentification with

another species from the same genus or family with similar or near-identical elytral patterns. We further studied more advanced machine learning techniques such as CNN, which also performed similarly (Wu et al., 2019).

Our findings in our earlier studies led us to scrutinize the image set and observe that images lacking visual clarity due to the reflective glare of the elytra surface were more prone to misidentification. To remedy this, we amended the optical and imaging settings and optimized the imaging conditions to obtain a high-quality image set unaffected by artifacts and showing the finer details of an elytron (Bera et al., 2021). We hypothesized that using such a high-quality image set would help us achieve a near-perfect prediction accuracy in identifying each pantry beetle species. In the current study, therefore, we tested this hypothesis by using a CNN model on an extended dataset which consisted of high-quality images of 27 species. We further shed light on the impact of enhanced images of 12 species in the same dataset that were previously studied. Our experiments showed both the utility of the prediction framework and the improvement in species identification due to image quality which could potentially guide any future efforts for auto-detection tools.

The rest of the paper is organized as follows: Section Material and methods details the dataset for 27 species and introduces our approach, Section Results and discussion presents our results, Section Discussion discusses our findings, and Section Conclusion concludes our work.

Materials and methods

Beetle sample collection and image acquisition

We elaborated on the details of sample collection, preparation, and imaging technique in our previous publications on imaging optimization (Bisgin et al., 2018; Wu et al., 2019; Bera et al., 2021). Briefly, we used 12 different pantry beetle species harvested from our in-house collection. We chose these species due to their prevalence and significance in food contamination, especially in North American food samples. Another 15 different species were collected from the U.S. Department of Agriculture's (USDA) Animal and Plant Health Inspection Service (APHIS) laboratory. Elytra from each beetle specimen were harvested, thoroughly cleaned through sonication in an ethanol solution, and subsequently preserved in 70% ethanol prior to imaging. Table 1 shows the full list of 27 species which include both our in-house collection (12 species) and additional 15 species.

The harvested elytra were then air-dried and imaged using stereo microscopes (Leica M205, Allendale, New Jersey). Unlike the older image set, which was subjected to varied magnification (in the 75–100× range) and two-point reflected light, we used a fixed magnification of 100× and transmitted light for this study. These amendments significantly reduced glare spots and other imaging artifacts, and drastically improved the clarity of elytral patterns (Bera et al., 2021). We used a Leica MC170HD camera to acquire the images with an image resolution set to 2,592 × 1,944 dpi (dots per inch, the highest resolution available). In this study, only images from the ventral side (underside) of the elytra were used. The concave shape of the elytra naturally preserves the ventral side elytral patterns. This selection allowed us to focus our attention on only the pattern recognition aspect without having to worry about such artifacts as variation or loss of setae (surface hair) or other sample damages that often occur on the frontal side of the elytra during food or sample preparation steps.

We used 20 elytral images per species. Each image subsequently was divided into smaller subimages (tiles) to simulate physical fragmentation of the elytra that are often observed in contaminated samples. This simulated fragmentation step was critical to our application, as it allowed us to increase the sample size and to validate our algorithms in close to real-life scenarios, in which elytral fragments are the only viable remains found in contaminated food samples.

Image preprocessing

Each image frame (captured at 100× magnification) had the elytra at the center of the white background. Thus, in the first step of preprocessing, we removed the white background by determining the elytral border (line of maximum change in contrast). Next, we randomly split images belonging to the same species to construct training and test sets by observing a 4:1 ratio, as shown schematically in Supplementary Figure 1, which was the same practice we used in our previous studies. Since an early study showed the utility of images with a size of 448 × 448 (Wu et al., 2019), we randomly cropped 100 regions so that each image was the same size. These sub-images guaranteed they would be inside the borders detected in the previous step and allowed to have overlap. This resulted in 46,300 training and 10,800 test images. By following such an exercise, we ensured that all sub-images of a particular image were put either in the training or test set in order to prevent information leak. This "blind" cross-validation strategy reduced bias and minimized the possibility of overfitting.

Convolutional neural network and the model structure

For the classification task here, we adopted CNNs, which have been widely used in the research community for image classification and segmentation in recent years (Lawrence et al., 1997; Krizhevsky et al., 2012; LeCun, 2021). The ability of

TABLE 1 The complete list of pantry beetles used in this study, listed alphabetically by their family, genus, species and common names, with abbreviations.

	Family	Genus	Species	Common Name	SP Id new
1	Anthribidae	*Araecerus*	*fasciculatus*	Coffee Bean Weevil	AAF
2	Anobiidae	*Lasioderma*	*serricorne*	Cigarette Beetle	ALS
3	Anobiidae	*Stegobium*	*paniceum*	Drugstore Beetle	ASP
4	Bostrichidae	*Rhyzopthera*	*dominica*	Lesser Grain Borer	BRD
5	Chrysomelidae	*Callosobruchus*	*maculatus*	Cowpea Weevil	CCM
6	Curculionidae	*Sitophilus*	*granarius*	Granary Weevil	CSG
7	Curculionidae	*Sitophilus*	*oryzae*	Rice Weevil	CSO
8	Curculionidae	*Sitophilus*	*zeamaise*	Maize Weevil	CSZ
9	Dermestidae	*Attagenus*	*Unicolor*	Black Carpet Beetle	DAU
10	Dermestidae	*Trogoderma*	*inclusum*	Cabinet Beetle	DTI
11	Laemophloeidae	*Cryptolestes*	*ferrugineus*	Rusty Grain Beetle	LCF
12	Laemophloeidae	*Cryptolestes*	*pusillus*	Flat Grain Beetle	LCP
13	Laemophloeidae	*Cryptolestes*	*turcicus*	Flour Mill Beetle	LCT
14	Silvanidae	*Ahasverus*	*advena*	Foreign Grain Beetle	SAA
15	Silvanidae	*Ahasverus*	*species*	Fungus Beetle	SAS
16	Silvanidae	*Cathartus*	*quadricollis*	Squarenecked Grain Beetle	SCQ
17	Silvanidae	*Oryzaephilus*	*mercator*	Merchant Grain Beetle	SOM
18	Silvanidae	*Oryzaephilus*	*surinamensis*	Saw-toothed Grain Beetle	SOS
19	Tenebrionidae	*Cynaeus*	*angustus*	Larger Black Flour Beetle	TCA
20	Tenebrionidae	*Gnatocerus*	*cornutus*	Broad-horned Flour Beetle	TGC
21	Tenebrionidae	*Latheticus*	*oryzae*	Longheaded Flour Beetle	TLO
22	Tenebrionidae	*Lophocateres*	*pusillus*	Siamese Grain Beetle	TLP
23	Tenebrionidae	*Palorus*	*ratzeburgii*	Smalleyed Flour Beetle	TPR
24	Tenebrionidae	*Tribolium*	*castaneum*	Red Flour Beetle	TTCa
25	Tenebrionidae	*Tribolium*	*confusum*	Confused Flour Beetle	TTCo
26	Tenebrionidae	*Tribolium*	*Destructor*	Dark Flour Beetle	TTD
27	Tenebrionidae	*Tribolium*	*madens*	Black Flour Beetle	TTM

CNN to learn features while applying convolutional filters during the training stage makes it appealing and different from conventional image classification methods (Zheng et al., 2006). These types of deep neural network structures comprise cascaded convolutional and pooling layers in which filters are utilized to attain the most informative features that eventually provide significantly reduced image sizes. The CNN final output is then passed to a dense layer in a flattened representation, allowing passage to subsequent dense layers that finally terminate in another fully-connected layer with a number of neurons equal to the number of classes (i.e., species, in our case).

We constructed a CNN by using Keras (Chollet, 2015), which is an application programming interface (API) that runs the Tensorflow machine learning platform (Abadi et al., 2016) in the backend and offers further image preprocessing utilities for more generalizable models. Specifically, our network architecture consists of four convolutional layers along with corresponding pooling layers. These perform downsampling, usually by either choosing the maximum or average value in a given region, and two additional dense layers. We employed 3×3 filters in the convolutional layers that were followed by max pooling layers using 2×2 windows to choose the maximum value. In order to avoid overfitting, we further adapted the dropout approach that randomly ignores some units at a desired level to prevent coadapting (Srivastava et al., 2014). In Figure 1, we illustrate the details of our network structure, listing all six layers and the number of nodes for each layer. We used Rectified Linear Unit (ReLU) activation function in the first five layers. In the final layer, we used a softmax function due to the multi-class nature of our predictions. For the optimizer, we used the Adam algorithm because of its efficient management of larger datasets and parameters (Kingma and Ba, 2014).

Keras's data augmentation features enabled us to artificially increase the sample size (i.e., number of subimages). Additionally, it helped generalize the model by applying image processing functions to the existing training samples. These functions perform image manipulations, such as rotations, that

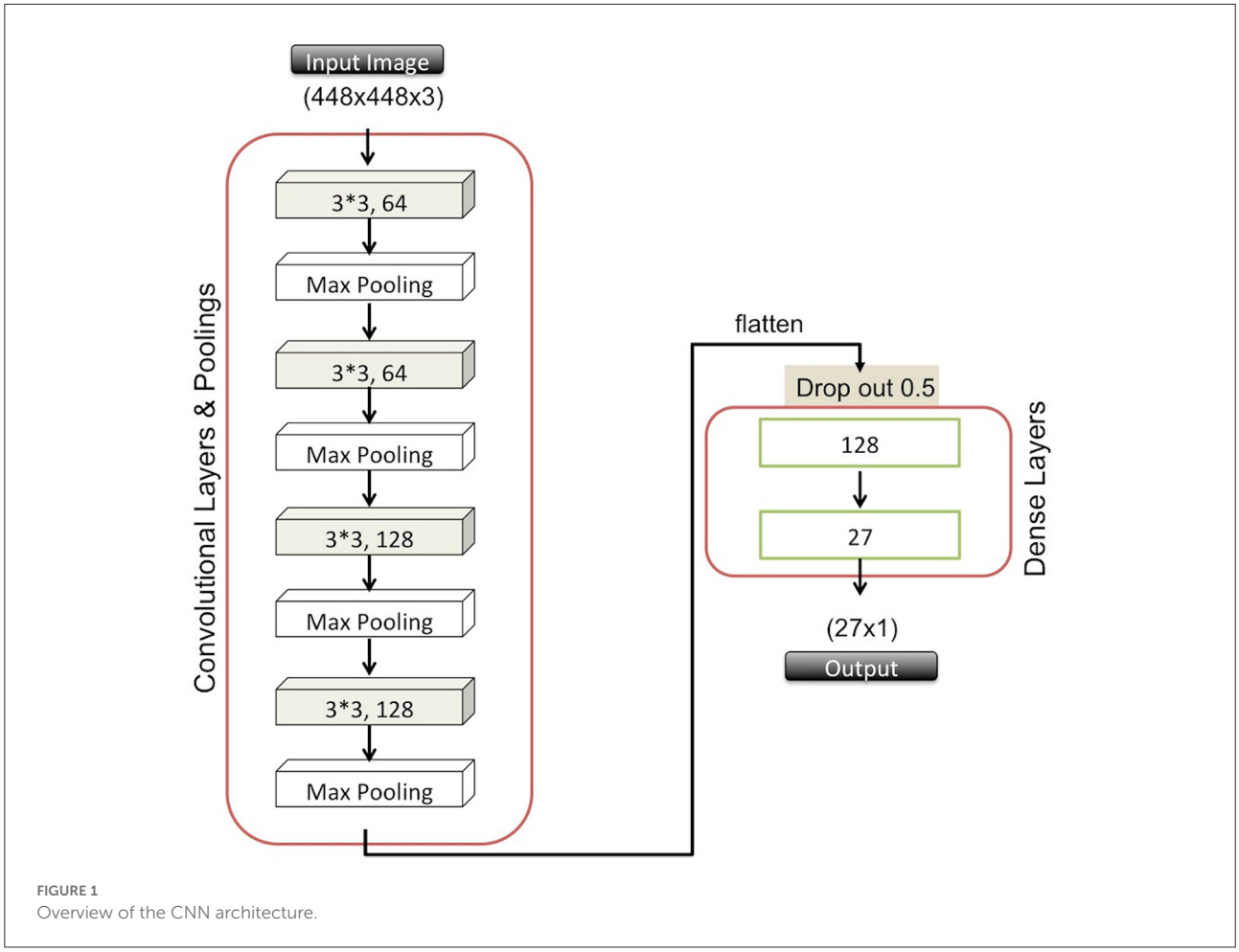


FIGURE 1
Overview of the CNN architecture.

TABLE 2 List of augmentation options and parameter values used in our study.

Option	Explanation	Value
rotation_range	Creates images with random rotations up to N degrees.	40
width_shift_range	Handles off-center objects by artificially creating shifted versions of the training data	0.2
height_shift_range		0.2
shear_range	Shear angle in counterclockwise direction in degrees	0.2
zoom_range	Random zoom range	0.2
horizontal_flip	Creates random flips of the image (supposes you feed a mirror image)	True
fill_mode	Helps in filling values outside the boundaries of an image	nearest

lead to a more diverse and larger set of images derived from the original set. We list details about the augmentation options and parameter values used in our study in Table 2. From the details shown in Table 2, we derived an augmented training set which could include additional images that might be shifted 20%, rotated 30 degrees, magnified 15%, sheared 10%, and horizontally flipped. If any pixels were lost due to the operations and needed to be filled to keep the image integrity, the nearest pixels could be used.

Model training and validation

Keras offers a user-friendly interface for data augmentation and experimental design, including the arrangement of training and test sets consisting of image folders maintained by the *ImageDataGenerator* module of the *keras_preprocessing* library. In our case, for 27 species we created a training directory that included 27 folders, from which class labels were inherited.

Similarly, we created a validation directory using the *flow_from_directory* function.

We passed these settings to the *fit_generator* function, along with the compiled neural network detailed above, with the *categorical_crossentropy* loss function, *adam* optimizer, and the default batch size (Bisgin et al., 2018). We trained our model for 100 epochs and tested its performance on the validation images after each epoch.

Model evaluation

As in our previous studies, we first computed the accuracy values for each species by computing the mean and standard deviation for each round of validation (Bisgin et al., 2018; Wu et al., 2019). This yielded a confusion matrix after the cross-validation from which true positive (TP), false positive (FP), true negative (TN), and false negative (FN) were computed. These were subsequently used to calculate the prediction parameters, namely Precision, Sensitivity (Recall), Specificity, Matthews Correlation Coefficient (MCC) using the standard formula, which can also be found in our previous report (Bisgin et al., 2018). Average prediction accuracy was also calculated by averaging species-wise accuracies.

Code and experimental environment

Given the significantly increased image size (average 14 mb per full elytra image and 600 kb per sub-images), we used the NCTR/FDA High-Performance Computing Cluster containing approximately 1100 CPU cores. The script used in this study can be found in github[1].

Results

Beetle species and the classifier

We initiated the study with 15 species of food-contaminating beetles most prevalent to North America. In the later part of the study, this number was expanded to 27 species. Table 1 contains a list of test species alphabetized by their family names with details on their nomenclature; namely, family, genus, species, and common names, along with their abbreviations. Those abbreviations were used to refer to each tested species. Supplementary Figure 2 shows some of the representative elytra images. For comparison, we provided images obtained though both the traditional and optimized methods. It was quite evident that imaging optimization significantly improved image quality and clarity of the elytral patterns. Compared to the traditionally acquired image set, the optimized image set was devoid of such artifacts as glare spots and other surface anomalies. Details on the imaging improvements, described elsewhere, are beyond the scope of this discussion (Bera et al., 2021). This image set subsequently was processed to obtain the set of sub-images used for our model.

Model summary

The analysis of training and validation progress of the 27 classes along epochs is reported in Figure 2. We observed that the training loss (i.e., categorical cross-entropy) began to stabilize after ∼50 epochs, beyond which the decrease was much more gradual. Also, we observed that testing accuracy approached saturation after ∼50 epochs. Both observations might indicate that the model had reached nearly optimal accuracy, and that 50 epochs would have been enough, which was close to our earlier observations. However, the loss function for the testing (validation loss) fluctuated, but tended to stay in a limited bandwidth around the value at 50 epochs.

Species-wise performance and comparison

To test the hypothesis that a high-quality image set may increase prediction accuracy, we made a head-to-head comparison of the prediction results (Recall and Precision) for the same 12 species for earlier and current image sets, as shown in Figure 3. Evidently, the newer high-quality image set improves the prediction performance for most species, with an average prediction accuracy increasing from 80% to above 90%. The improvements were particularly notable for such species as ALS and ASP, SOM and SOS, and TTCa and TTCo; these had previously been difficult to accurately identify, however, can now be identified with >90% accuracy. These 12 species, especially, SOM, SOS, TTCa and TTCo, are some of most commonly encountered pantry beetles in North America. Therefore, improving the accuracy of their prediction identification will have regulatory significance. The traditionally-obtained images with higher artifacts and lower quality lacked the pattern clarity to distinguish one species from another. This was particularly true for species with near-identical elytral patterns (due to their genetic similarity) and belonging to the same genus and/or family [referred to as "difficult pairs" in our previous works (Bisgin et al., 2018; Wu et al., 2019)]. The high-quality images significantly improved the pattern clarity, allowing for distinct identification of each species, even within the difficult pairs. To our surprise, we observed exceptions to this general trend, especially for the species CSO. Of all 12 species, this one performed the poorest and showed a significant decrease in prediction accuracy

1 https://github.com/hbisgin/beetleCNN

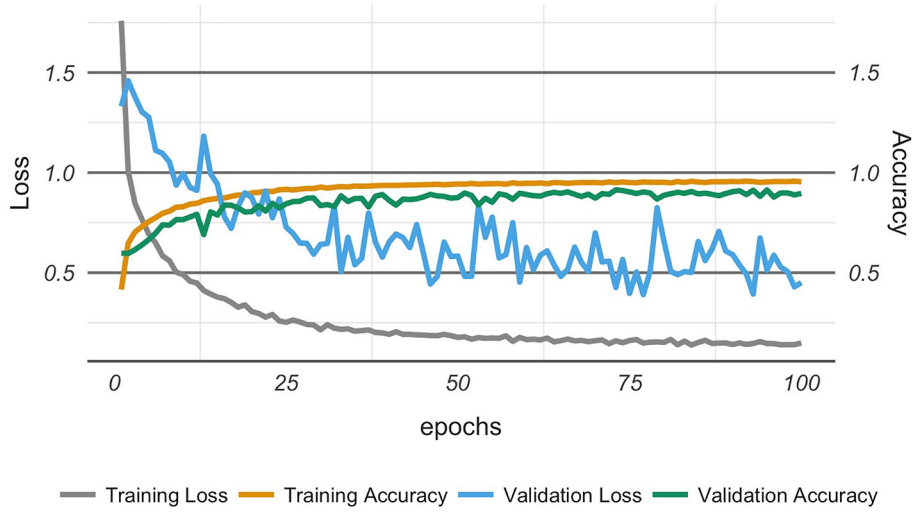


FIGURE 2
Model optimization showing the model achieving optimal performance after about 50 epochs.

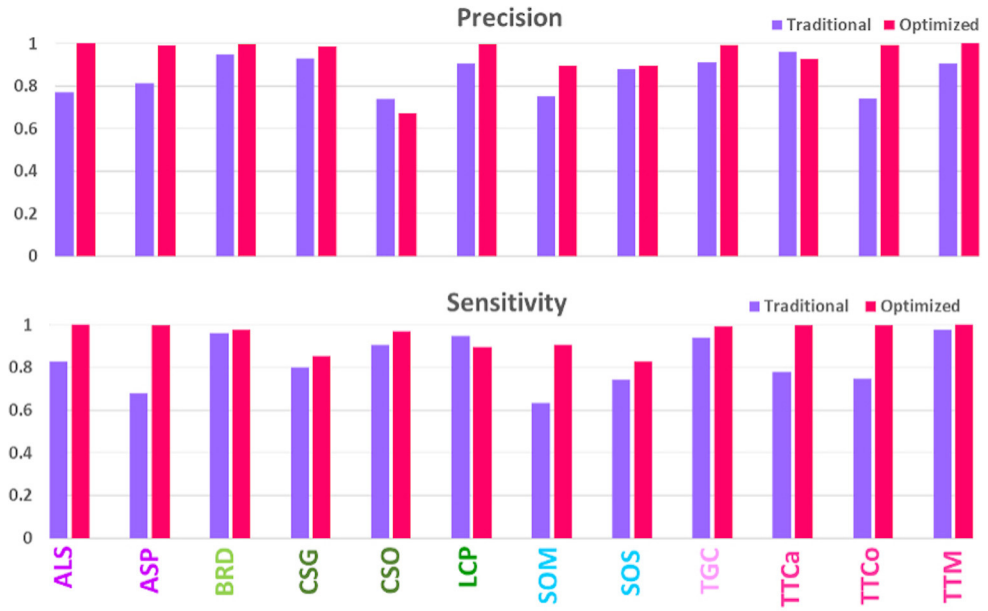


FIGURE 3
Comparison of model performances on validation sets of traditionally- and optimally-acquired images.

compared to the traditionally-acquired images. The image set for this particular species possibly contained an anomaly, resulting in this decrease.

Expansion to more species—performance parameters

Expanding the number of species to 27 enabled us to verify the observations made with the initial 12 species with our newly built model in this study. Four prediction parameters, namely Precision, Recall (or Sensitivity), Specificity and MCC, for these species are presented in Figure 4. The general trend of improved prediction is evident from this figure. Specificity values for all the species validate our hypothesis that high-quality images can improve prediction accuracy. However, there were exceptions to the general trend; as some species, such as CSO and LCP, performed quite poorly. Several other species, namely AAF, CSG, LCF, SAA, SOM, SOS, and TCA, performed below average, i.e., 90%. This suggests poor performance is not a

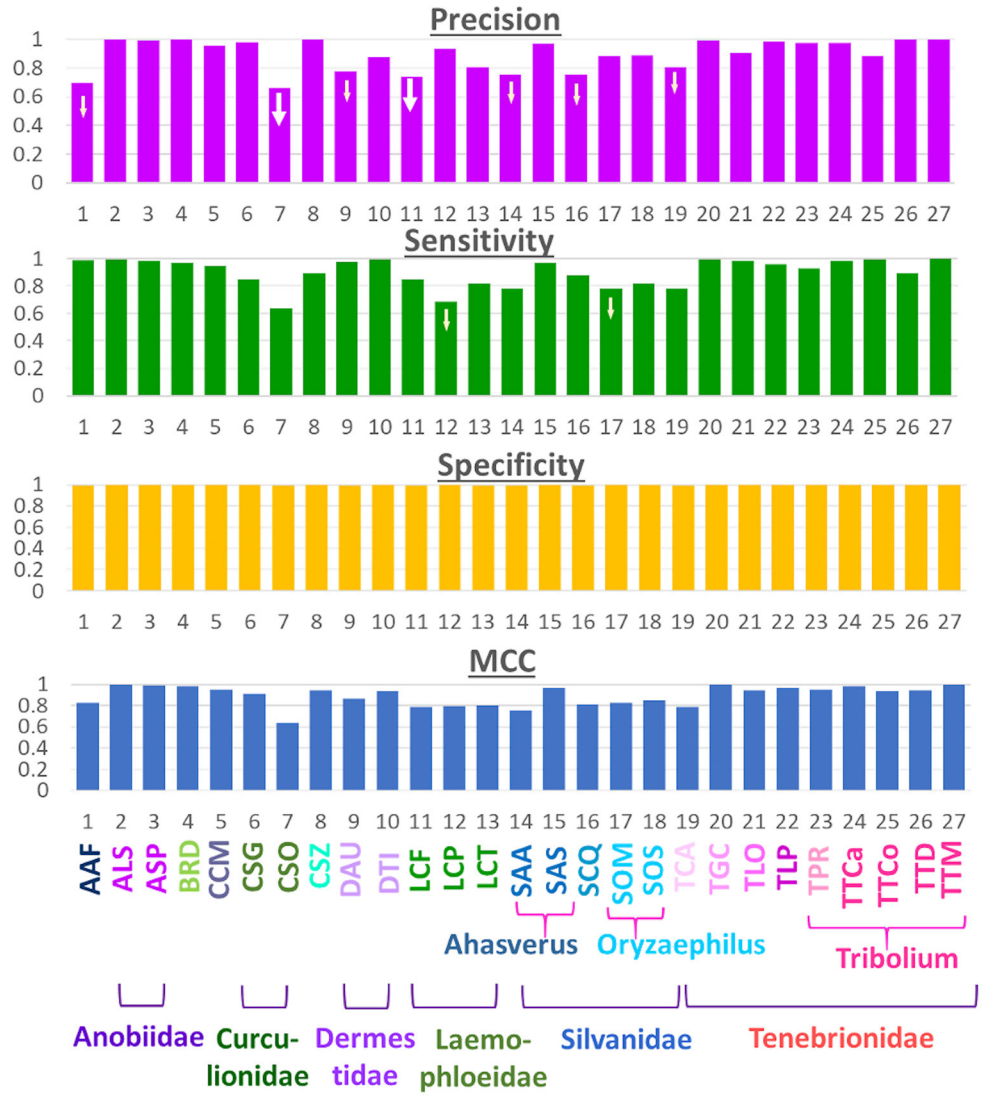


FIGURE 4
Performance metrics for the 27-class model.

singular anomaly in the image set of one species. Instead, there may be underlying factors that play a crucial role in a species' prediction performance and these need further research. One possibility, as we observed previously, is that species with similar elytral patterns (belonging to the same genus and/or family) were confused with one another during the prediction.

Confusion matrix

Figure 5 shows the confusion matrix for all the species, with horizontal rows showing the True class and vertical rows the Predicted class. It is evident that overall performance of the model is quite accurate, as the red diagonal entities are clearly prominent. Although the model is far from perfect, as one can observe several non-diagonal entities in yellow, it is a good working model since the deviations were fairly low (mostly yellow and not orange non-diagonal entities) as indicated by the color scale. A closer look at the matrix, especially for the poorly-performing species (marked with red arrows) such as CSO, indicated that its low prediction performance was not due to the similarity of elytral patterns with a species from the same genus or family (marked by dotted squares). Rather, it was being predicted for several different species across various families. For instance TTCo was predicted as SOM and SOS for 9 and 6 times, respectively, compared to TGC, which is in the same family. This suggests that the image quality that showed distinction (or resulted in confusion) between

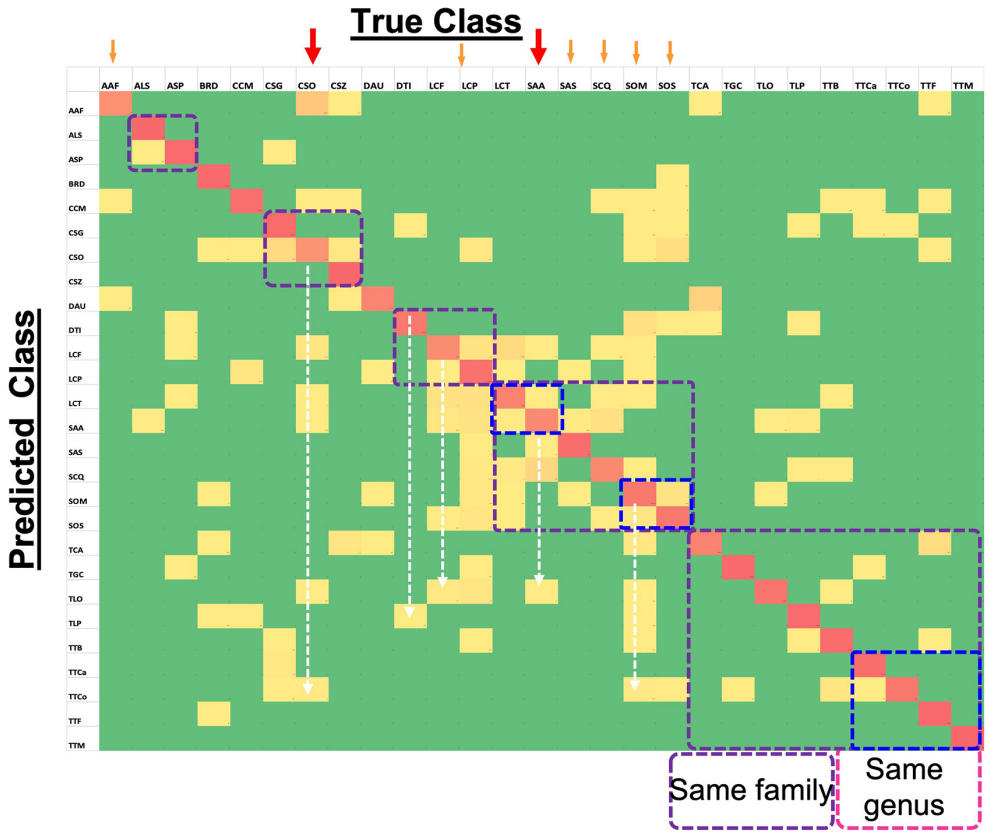


FIGURE 5
Confusion Matrix for 27-class task (computed on test set) showing the level of agreement between true and predicted classes. Red colored tiles (diagonal) represent correct classification of each species and represent values between 67% and 100%. Yellow tiles represent incorrect classification ratios that are non-zero and go up to 28%. Finally, green tiles represent zero values which means targeted species is not confused with the corresponding species.

similar elytral patterns is not the major factor at play on our data. We made a similar observation for the second-lowest performer, LCF, which was also predicted beyond its own genus and/or family. Other low-performing species, such as AAF, SAA, SCQ, SOM, SOS, and TCA, showed comparable trends. The two-dimensional UMAP representation of all classes based on their extracted 128 features from the last layer of the network (Supplementary Figure 3) also illustrates misclassified species. This observation further bolstered our speculation that something other than pattern clarity may be affecting the prediction performance, and deserved detailed discussion.

Discussion

In most academic and research settings, the architecture of the model often receives more attention than does the quality of the data, possibly because cleaning the dataset often is beyond the scope of many researchers. This has been found to be true, particularly in image classification for species identification applications. Users of prediction models, even models with the best-known architecture, have found achieving good accuracy for noisy datasets challenging as quality of the data has impact on the classifier performance (Sáez et al., 2016). Our study also highlights this fact in the context of species identification and food safety, as the prediction performance showed improvement when a better-quality dataset was used to build the model.

Furthermore, our results indicated the importance and relevance of other factors beyond data quality. As discussed previously, we observed that species performing below average were not being inaccurately predicted or confused with another species from their own genus and/or family due to elytral pattern similarity, but were being misclassified into various different and unrelated species. To better understand this problem, we delved deeper and looked through the images of those species. Figure 6A shows three different elytral images of the same species, CSO. The difference in elytral patterns are obvious, and believed to be mostly due to age of the beetle. However, differences

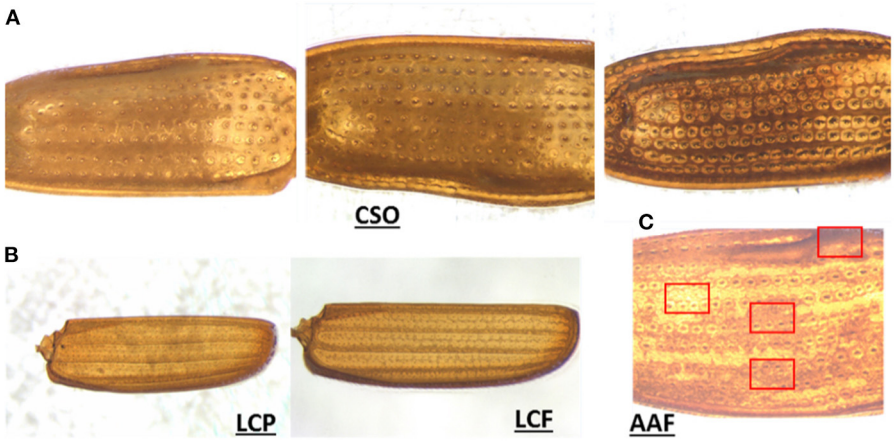


FIGURE 6
Representative images of elytral variation. **(A)** Intraspecies pattern variation in CSO (possibly due to the difference in maturity), **(B)** pattern variation due to background interference in LCP and LCF, and **(C)** regional variation in elytral patterns in AAF.

could also be due to sex and/or individual variation, as the older beetles tend to develop a darker elytral color and prominent pattern, possibly to attract a mate. These variations are not uncommon and were found in such other species as SAA and SOS (Supplementary Figure 4), which also performed poorly.

Surprisingly, these species did not perform so poorly in our previous models using ANN and SVM. Supplementary Figure 5 shows the Recall and Precision comparisons for ANN, SVM, and CNN models using a conventionally-obtained (lower quality) image set than that of the present model (CNN, using a higher-quality image set). In comparison to ANN and SVM models, the performances of CSO dropped in both CNN models (using conventional and high-quality image sets), even though the CNN model is known to, in most cases, outperform ANN and SVM (Shin and Balasingham, 2017; Senyurek et al., 2019). We argue that this anomaly is due to the difference between explicitly defining features or trusting the CNN to develop its own feature extraction internally. In both ANN and SVM, the image features (such as size, shape, distribution, and color of the elytral pattern), were preprocessed before being used for training and testing the model. It is during this feature selection process that the intraspecies variations in elytral patterns probably did not get selected in the top-ranked features, as they appeared in only a handful of species. Subsequently, they remained unused in the ANN and SVM models and showed no influence in performance. On the other hand, output of convolutional layers served as the feature set in the current model, which could not take advantage of earlier select features, possibly causing a decrease in performance.

Unlike CSO, the species LCP (the second-lowest performer) did not show significant intraspecies variation. On minute observation, we found they contained imaging artifacts. LCP belongs to the family Laemophloeidae, which is one of the smallest species of pantry beetles. They also have extremely thin elytra and faint patterns, which when imaged on filter papers (a common practice in food filth analysis), in some cases resulted in a fibrous paper background getting embedded in the elytral images (see Figure 6B). This imaging artifact was prominent in some parts of a few of the elytral images, which appeared quite different from the actual elytral pattern and could very well be the reason for their poorer performance. AAF was another species performing below average. In this case, each elytron had regions that appeared different from one another (variegated pattern). In some areas, the elytra appeared much brighter, while in other regions they appeared much darker. Some regions had more prominent patterns compared to others (see Figure 6C). When the images of the whole elytra were divided into subimages, the subimage set had much more pattern diversity. Some of the randomly-selected subimages used in testing probably appeared quite different from the training subimages, yielding a lower prediction value. It can also be noted that the AAF had a high Recall value but low Precision values. This indicated that our model was impressive in choosing relevant species, but in this case was slightly less exact due to highly diverse subimages.

While our collective results showed that model performance improved significantly when using better-quality images, thus validating our initial hypothesis, they indicated that species with higher intraspecies elytral diversity or with enhanced variegated

elytral patterns do not perform as well. These observations seemed reasonable and have room for improvement without needing significant change in the model architecture. They are also aligned with a known general limitation of CNN models, which require training sets with both high-quality and large-quantity of images to yield better prediction accuracies (Valan et al., 2019; Høye et al., 2020).

While the cropped subimages were a way of imitating the actual beetle fragments and artificially increasing the size of the dataset, the limited number of elytral images remained one of the challenges in this study. Adding Keras image augmentation became a possible solution, as it has been used to solve imaging issues in domains such as medical image analysis (Shorten and Khoshgoftaar, 2019). In this step, several other scenarios, such as rotation, shearing, and zooming to some extent, were incorporated. During the training stage, the model was exposed to data augmentation to prepare it for possible variations, including likely presence of fragmented patterns. Even though this approach worked very well both for training accuracy and training loss, slightly lower accuracy and fluctuating loss observed in the validation stage also indicates that high variability of novel patterns is much harder to control and beyond the reach of data augmentation.

The broader objective of our work is to automate the process of elytral pattern recognition to better alleviate insect food contamination. We foresee this can only be achieved by concatenating the following three steps: (1) establishing a mechanism of acquiring high-quality images, (2) accumulating beetle images with proper labels in a repository with a growing number of samples for species with high variability, and (3) making them accessible for model development/improvement. Before developing a full-force effort to implement the whole process, it was critical to validate with a proof of concept the hypothesis that high-quality images can significantly improve predictive accuracy. The present study served this purpose and indicated that a high number of high-quality images is indeed a promising way forward in achieving precise identification over a large number of species. In our recent report on imaging optimization techniques, we elaborated on the method for acquiring high-quality images of pantry pests. Through this study, we developed a step-by-step procedure and a detailed instruction manual for high-quality image acquisition, which we will make publicly available. We currently are in the process of developing a high-quality image database containing 40 images per species for about 40 different pantry beetles, which will also be made public. Efforts currently are underway to construct a graphical user interface (GUI), from which any user can upload elytral images (preferably obtained by following the SOP and imaging manual) of pantry beetles in order to identify species using a CNN model similar to the one reported here. This use of the GUI will further enhance the high-quality image database and will provide a large number of high-quality, well-labeled image sets which can be used to further improve this CNN model in the future. At this point, the present work explores advantages and limitations of using a CNN model for classifying various species of pantry pests through elytral pattern recognition. We are optimistic that the current study has put us a step closer to achieving automated species identification of pantry pests, and thus toward a more efficient regulatory system to better manage food contamination scenarios.

Conclusions

In this study, we aimed at scouring the landscape and moving closer to achieving near-perfect species-level identification. We set out to explore whether high-quality elytral images were sufficient for improving the prediction accuracy of pantry beetle species identification. To test this hypothesis, we first compared two CNN models; one developed with traditionally-obtained, low-resolution images, and another with optimized imaging conditions, yielding high-quality images. Overall, we observed an improvement in average prediction accuracy due to the improved image quality. When we extended the analysis to 27 different pantry beetles, we achieved an average accuracy of ∼90%; however, several species fell below that average accuracy. A data review elucidated that below-average performance was not due to poor image quality, but rather to significant intraspecies variation of elytral pattern, and in some cases, to enhanced regional variation of patterns within one elytron. Detailed analysis indicated that greater numbers of high-quality images are necessary to account for these variations and achieve higher accuracy of the model. In future studies, we aim to achieve this objective using a publicly-available GUI for pantry beetle identification, allowing us to accumulate larger quantities of high-quality images through user participation. We hope this exploratory study will help achieve our ultimate goal of automated species identification of food-contaminating beetles.

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://github.com/hbisgin/BeetleCNN.

Author contributions

TB acquired the images. HB performed the calculations. TB and HB analyzed the results with help from LW, NB, ZL, CF, and JX. HD and AB provided the in-house entomological support including samples and imaging facility. MP-R and JC provided the external entomological support and consultation. JX and HD led the projects. HV and WT managed and supported the study. All authors reviewed and approved the manuscript.

Funding

This work was supported by NCTR Grant E0759101.

Acknowledgments

The authors thank to Drs. Pierre Alusta and Tucker Patterson of the National Center for Toxicological Research (NCTR), and Drs. Andrew Fang and Michael Wichman of FDA's Arkansas Laboratory (ARKL) for their comments and suggestions during internal reviews of the manuscript. HB was grateful to NCTR and the Oak Ridge Institute for Science and Education (ORISE) for the Faculty Research Fellowship, where the work initiated. TB was grateful to NCTR and ORISE for his postdoctoral fellowship.

Conflict of interest

Author CF was employed by company HK3 Lab.

The remaining 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.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frai.2022.952424/full#supplementary-material

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., et al. (2016). "Tensorflow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation*, 265–283.

Al-Saqer, S. M., and Hassan, G. M. (2011). Artificial neural networks based red palm weevil (*Rynchophorus ferrugineous*, Olivier) recognition system. *Am. J. Agric. Biol. Sci.* 6, 356–364. doi: 10.3844/ajabssp.2011.356.364

Bansal, S., Singh, A., Mangal, M., Mangal, A. K., and Kumar, S. (2017). Food adulteration: sources, health risks, and detection methods. *Critic. Rev. Food Sci. Nutr*. 57, 1174–1189. doi: 10.1080/10408398.2014.967834

Bell, C. H. (2013). *Food Safety Management: Chapter 29. Pest Management*. Amsterdam: Elsevier Science. doi: 10.1016/B978-0-12-381504-0.00029-9

Belluco, S., Losasso, C., Maggioletti, M., Alonzi, C. C., Paoletti, M. G., Ricci, A., et al. (2013). Edible insects in a food safety and nutritional perspective: a critical review. *Comprehens. Rev. Food Sci. Food Saf.* 12, 296–313. doi: 10.1111/1541-4337.12014

Bera, T., Wu, L., Ding, H., Semey, H., Barnes, A., Liu, Z., et al. (2021). Optimized imaging methods for species-level identification of food-contaminating beetles. *Sci. Rep.* 11, 1–13. doi: 10.1038/s41598-021-86643-y

Bisgin, H., Bera, T., Ding, H., Semey, H. G., Wu, L., Liu, Z., et al. (2018). Comparing SVM and ANN based machine learning methods for species identification of food contaminating beetles. *Sci. Rep*. 8, 6532. doi: 10.1038/s41598-018-24926-7

Chen, Y., Zhang, X., Chen, W., Li, Y., and Wang, J. (2020). Research on recognition of fly species based on improved RetinaNet and CBAM. *IEEE Access*. 8, 102907–102919. doi: 10.1109/ACCESS.2020.2997466

Chollet, F. (2015). *Keras*. Available online at: https://keras.io (accessed May 11, 2022).

Cortes, C., and Vapnik, V. (1995). Support-vector networks. *Mach. Learn*. 20, 273–297. doi: 10.1007/BF00994018

da Silva, F. L., Sella, M. L. G., Francoy, T. M., and Costa, A. H. R. (2015). Evaluating classification and feature selection techniques for honeybee subspecies identification using wing images. *Comput. Electron. Agric*. 114, 68–77. doi: 10.1016/j.compag.2015.03.012

Daly, H. V., Hoelmer, K., Norman, P., and Allen, T. (1982). Computer-assisted measurement and identification of honey bees (Hymenoptera: Apidae). *Ann. Entomol. Soc. Am*. 75, 591–594. doi: 10.1093/aesa/75.6.591

DeChant, C., Wiesner-Hanks, T., Chen, S., Stewart, E. L., Yosinski, J., Gore, M. A., et al. (2017). Automated identification of northern leaf blight-infected maize plants from field imagery using deep learning. *Phytopathology*. 107, 1426–1432. doi: 10.1094/PHYTO-11-16-0417-R

Heeps, J. (2016). *Insect Management for Food Storage and Processing*. Amsterdam: Elsevier Science.

Høye, T. T., Ärje, J., Bjerge, K., Hansen, O. L., Iosifidis, A., Leese, F., et al. (2020). Deep learning and computer vision will transform entomology. *bioRxiv*. 2020.07.03.187252.doi: 10.1101/2020.07.03.187252

Kingma, D. P., and Ba, J. (2014). Adam: a method for stochastic optimization. *arXiv preprint arXiv:*14126980.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). "ImageNet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1 (NIPS'12)* (Red Hook, NY: Curran Associates Inc), 1097–1105.

Kuzuhara, H., Takimoto, H., Sato, Y., and Kanagawa, A. (2020). Insect pest detection and identification method based on deep learning for realizing a pest control system. *IEEE* 2020, 709–714. doi: 10.23919/SICE48898.2020.9240458

Larios, N., Deng, H., Zhang, W., Sarpola, M., Yuen, J., Paasch, R., et al. (2008). Automated insect identification through concatenated histograms of local appearance features: feature vector generation and region detection for deformable objects. *Mach. Vis. Appl*. 19, 105–123. doi: 10.1007/s00138-007-0086-y

Lawrence, S., Giles, C. L., Tsoi, A. C., and Back, A. D. (1997). Face recognition: a convolutional neural-network approach. *IEEE Trans. Neural Netw*. 8, 98–113. doi: 10.1109/72.554195

LeCun, Y. (2021). *LeNet-5, Convolutional Neural Networks* (2021). Available online at: http://yann.lecun.com/exdb/lenet (accessed August 01, 2021).

Lee, S. H., Chan, C. S., Wilkin, P., and Remagnino, P. (2015). Deep-plant: plant identification with convolutional neural networks. *IEEE* 2015, 452–456. doi: 10.1109/ICIP.2015.7350839

Lu, Y., Yi, S., Zeng, N., Liu, Y., and Zhang, Y. (2017). Identification of rice diseases using deep convolutional neural networks. *Neurocomputing*. 267, 378–384. doi: 10.1016/j.neucom.2017.06.023

Martin, D., Ding, H., Wu, L., Semey, H., Barnes, A., Langley, D., et al. (2016). "An image analysis environment for species indentification for food contaminating beetles," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, *vol. 16*, 4375–4376. doi: 10.1609/aaai.v30i1.9846

Martineau, M., Conte, D., Raveaux, R., Arnault, I., Munier, D., Venturini, G. A., et al. (2017). survey on image-based insect classification. *Pattern Recogn.* 65, 273–284. doi: 10.1016/j.patcog.2016.12.020

Mayo, M., and Watson, A. T. (2007). Automatic species identification of live moths. *Knowledge-Based Syst.* 20, 195–202. doi: 10.1016/j.knosys.2006.11.012

Norouzzadeh, M. S., Nguyen, A., Kosmala, M., Swanson, A., Palmer, M. S., Packer, C., et al. (2018). Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. *Proc. Natl. Acad. Sci. USA* 115, E5716. doi: 10.1073/pnas.1719367115

Olsen, A. R., Gecan, J. S., Ziobro, G. C., and Bryce, J. R. (2001). Regulatory action criteria for filth and other extraneous materials v. strategy for evaluating hazardous and nonhazardous filth. *Regul Toxicol Pharmacol.* 33, 363–392. doi: 10.1006/rtph.2001.1472

O'Neill, M. A., Gauld, I. D., Gaston, K. J., and Weeks, P. J. D. (2000). "Daisy: an automated invertebrate identification system using holistic vision techniques," in *Proceedings of the Inaugural Meeting BioNET-INTERNATIONAL Group for Computer-Aided Taxonomy (BIGCAT)*, 13–22.

Qing, Y. A., Jun, L. V., Liu, Q. J., Diao, G. Q., Yang, B. J., Chen, H. M., et al. (2012). An insect imaging system to automate rice light-trap pest identification. *J. Integr. Agric.* 11, 978–985. doi: 10.1016/S2095-3119(12)60089-6

Reinholds, I., Bartkevics, V., Silvis, I. C., van Ruth, S. M., and Esslinger, S. (2015). Analytical techniques combined with chemometrics for authentication and determination of contaminants in condiments: a review. *J. Food Compos. Anal.* 44, 56–72. doi: 10.1016/j.jfca.2015.05.004

Sáez, J. A., Luengo, J., and Herrera, F. (2016). Evaluating the classifier behavior with noisy data considering performance and robustness: the equalized loss of accuracy measure. *Neurocomputing*. 176, 26–35. doi: 10.1016/j.neucom.2014.11.086

Senyurek, V. Y., Imtiaz, M. H., Belsare, P., Tiffany, S., and Sazonov, E. A. (2019). comparison of SVM and CNN-LSTM based approach for detecting smoke inhalations from respiratory signal. *IEEE* 2019, 3262–3265. doi: 10.1109/EMBC.2019.8856395

Shi, Z., Dang, H., Liu, Z., and Zhou, X. (2020). Detection and identification of stored-grain insects using deep learning: a more effective neural network. *IEEE Access*. 8, 163703–163714. doi: 10.1109/ACCESS.2020.3021830

Shin, Y., and Balasingham, I. (2017). Comparison of hand-craft feature based SVM and CNN based deep learning framework for automatic polyp classification. *IEEE* 2017, 3277–3280. doi: 10.1109/EMBC.2017.8037556

Shorten, C., and Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *J. Big Data*. 6, 60. doi: 10.1186/s40537-019-0197-0

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15, 1929–1958.

Terry, J. C. D., Roy, H. E., and August, T. A. (2019). Thinking like a naturalist: enhancing computer vision of citizen science images by harnessing contextual data. *bioRxiv*. 730887. doi: 10.1101/730887

Valan, M., Makonyi, K., Maki, A., Vondráček, D., and Ronquist, F. (2019). Automated taxonomic identification of insects with expert-level accuracy using effective feature transfer from convolutional networks. *Syst. Biol.* 68, 876–895. doi: 10.1093/sysbio/syz014

Wang, J., Lin, C., Ji, L., and Liang, A. A. (2012). new automatic identification system of insect images at the order level. *Knowledge-Based Syst.* 33, 102–110. doi: 10.1016/j.knosys.2012.03.014

Weeks, P. J., Gauld, I. D., Gaston, K. J., and O'Neill, M. A. (1997). Automating the identification of insects: a new solution to an old problem. *Bull. Entomol. Res.* 87, 203–211. doi: 10.1017/S000748530002736X

Wu, L., Liu, Z., Bera, T., Ding, H., Langley, D. A., Jenkins-Barnes, A., et al. (2019). A deep learning model to recognize food contaminating beetle species based on elytra fragments. *Comput. Electron. Agric.* 166, 105002. doi: 10.1016/j.compag.2019.105002

Yalcin, H. (2015). Vision based automatic inspection of insects in pheromone traps. *IEEE;* 2015, 333–338. doi: 10.1109/Agro-Geoinformatics.2015.7248113

Yang, H.-P., Ma, C.-S., Wen, H., Zhan, Q.-B., and Wang, X.-L. (2015). A tool for developing an automatic insect identification system based on wing outlines. *Sci. Rep.* 5, 1–11. doi: 10.1038/srep12786

Zhang, X., Qiao, Y., Meng, F., Fan, C., and Zhang, M. (2018). Identification of maize leaf diseases using improved deep convolutional neural networks. *IEEE Access*. 6, 30370–30377. doi: 10.1109/ACCESS.2018.2844405

Zheng, C., Sun, D.-., W., and Zheng, L. (2006). Recent developments and applications of image features for food quality evaluation and inspection-a review. *Trends Food Sci. Technol.* 17, 642–655. doi: 10.1016/j.tifs.2006.06.005

frontiers | Frontiers in Artificial Intelligence

TYPE Methods
PUBLISHED 12 September 2022
DOI 10.3389/frai.2022.991733

OPEN ACCESS

EDITED BY
Zhaohui Steve Qin,
Emory University, United States

REVIEWED BY
Yanting Huang,
Emory University, United States
Yanqing Zhang,
Georgia State University, United States

*CORRESPONDENCE
Xiaoming Liu
xiaomingliu@usf.edu
Yongsheng Bai
bioinformaticsresearchtomorrow@gmail.com

†These authors have contributed equally to this work and share first authorship
‡These authors have contributed equally to this work and share senior authorship

SPECIALTY SECTION
This article was submitted to Medicine and Public Health, a section of the journal Frontiers in Artificial Intelligence

RECEIVED 11 July 2022
ACCEPTED 17 August 2022
PUBLISHED 12 September 2022

CITATION
Li C, Ma K, Xu N, Fu C, He A, Liu X and Bai Y (2022) SNPAAMapper-Python: A highly efficient genome-wide SNP variant analysis pipeline for Next-Generation Sequencing data. *Front. Artif. Intell.* 5:991733. doi: 10.3389/frai.2022.991733

SNPAAMapper-Python: A highly efficient genome-wide SNP variant analysis pipeline for Next-Generation Sequencing data

Chang Li[1†], Kevin Ma[2†], Nicole Xu[3], Chenjian Fu[4], Andrew He[5], Xiaoming Liu[1*‡] and Yongsheng Bai[6,7*‡]

[1]USF Genomics and College of Public Health, University of South Florida, Tampa, FL, United States, [2]Canyon Crest Academy, San Diego, CA, United States, [3]Obra D. Tompkins High School, Katy, TX, United States, [4]College of Arts and Sciences, Kent State University, Kent, OH, United States, [5]Pittsford Mendon High School, Pittsford, NY, United States, [6]Next-Gen Intelligent Science Training, Ann Arbor, MI, United States, [7]Department of Biology Eastern Michigan University, Ypsilanti, MI, United States

Currently, there are many publicly available Next Generation Sequencing tools developed for variant annotation and classification. However, as modern sequencing technology produces more and more sequencing data, a more efficient analysis program is desired, especially for variant analysis. In this study, we updated SNPAAMapper, a variant annotation pipeline by converting perl codes to python for generating annotation output with an improved computational efficiency and updated information for broader applicability. The new pipeline written in Python can classify variants by region (Coding Sequence, Untranslated Regions, upstream, downstream, intron), predict amino acid change type (missense, nonsense, etc.), and prioritize mutation effects (e.g., synonymous > non-synonymous) while being faster and more efficient. Our new pipeline works in five steps. First, exon annotation files are generated. Next, the exon annotation files are processed, and gene mapping and feature information files are produced. Afterward, the python scrips classify the variants based on genomic regions and predict the amino acid change category. Lastly, another python script prioritizes and ranks the mutation effects of variants to output the result file. The Python version of SNPAAMapper accomplished the overall speed by running most annotation steps in a substantially shorter time. The Python script can classify variants by region in 53 s compared to 166 s for the Perl script in a test sample run on a Latitude 7480 Desktop computer with 8GB RAM and an Intel Core i5-6300 CPU @ 2.4Ghz. Steps of predicting amino acid change type and prioritizing mutation effects of variants were executed within 1 s for both pipelines. SNPAAMapper-Python was developed and tested on the ClinVar database, a NCBI database of information on genomic variation and its relationship to human health. We believe our developed Python version of SNPAAMapper variant annotation pipeline will benefit the community by elucidating the variant consequence and speed up the discovery of causative genetic variants through whole genome/exome sequencing. Source codes, test data files, instructions, and

further explanations are available on the web at https://github.com/BaiLab/SNPAAMapper-Python.

KEYWORDS

Next-Generation Sequencing, SNP, python, mutation, variant annotation, pipeline

Introduction

Next-Generation Sequencing is a technique to rapidly sequence a genome and was developed because of the Human Genome Project, which successfully sequenced a human genome over a period of 23 years (www.genome.gov/human-genome-project) and cost around $2.7 billion in 1991 Fiscal Year Dollars, equivalent to $5.6 Billion 2022 Fiscal Year Dollars. Today, a human genome can be accurately sequenced for as low as $600 (Preston et al., 2022). In 2013, sequencing a whole human genome took between 1 and 2 days (Lewis, 2013).

With the decreasing cost and increasing availability of the Next-Generation-Sequencing technique (Barba et al., 2014), our ability to discover variants in the human genome has been revolutionized. More variants have been reported and discovered. Our ability to interpret or annotate these variants becomes a major gap in effectively using genomics data in understanding diseases. To address this issue, multiple variant annotation tools that locate and assign information about variants have been developed.

One such tool is SNPAAMapper, a variant analysis tool developed in 2013 in the Perl coding language. SNPAAMapper contains two general algorithms: one that generates annotation tables with coding and other information annotated for each exon, and one that reads the generated annotation tables and assigns identified variants to the genomic loci and classifies them by region (Bai and Cavalcoli, 2013).

The original SNPAAMapper used the Perl coding language to make alignment of input DNA sequences, which may have sub-optimal performance. This inefficiency and inability to handle big data in a timely manner placed a hurdle in its wide applications. In this study, we chose to update and modify SNPAAMapper to substantially increase the speed of the program to fulfill the current need of the genomics field. Additionally, we presented an improved output to facilitate downstream data processing and analysis.

Methods

Input data acquisition

The reference genomes used in the paper were sourced from the UCSC genome browser (https://genome.ucsc.edu/). The UCSC Genome Browser is a web-based tool that allows researchers to example all 23 chromosomes of the human genome all the way down to an individual nucleotide. It also contains data on the genomes of more than a 100 other organisms. The genome browser was created and maintained by Jim Kent and David Haussler at UCSC in 2000 as a resource for the distribution of results from the Human Genome Project. It was funded by the Howard Hughes Medical Institute and the National Human Genome Research Institute (NHGRI) (https://genome.ucsc.edu/goldenPath/history.html).

When testing our tool, we used both a small test dataset (number of variants = 80) and data file from the ClinVar database (https://www.ncbi.nlm.nih.gov/clinvar/) (number of variants = 1,440,883), a publicly accessible archive of reports of relationships between human variations and phenotypes. ClinVar is crowdsourced and relies on the submission of reports by researchers and clinical labs. The default format for ClinVar database is VCF and the database file was downloaded in assembly GRCh37/hg19 for the human reference genome on May 28, 2022. VCF is the default format for ClinVar to store and report variants, including point mutations and short insertions/deletions. In the "INFO" columns, some related annotation information was also provided by ClinVar, such as the associated clinical significance, associated diseases, gene annotation etc.

Algorithm description

The pseudocodes for SNPAAMapper-Python algorithms are described in Algorithm 1 (see Supplementary materials). There are two modules of the algorithm: (1) Preprocess the gene structure to build annotation for each exon; (2) Map identified variants onto the genomic location and report the hit class. In the Python version of SNPAAMapper, the second script for processing exon annotation files and generating feature start and gene mapping files performs extremely better than the one in the original Perl version. The screenshot for SNPAAMapper on the GitHub site is shown in Figure 1.

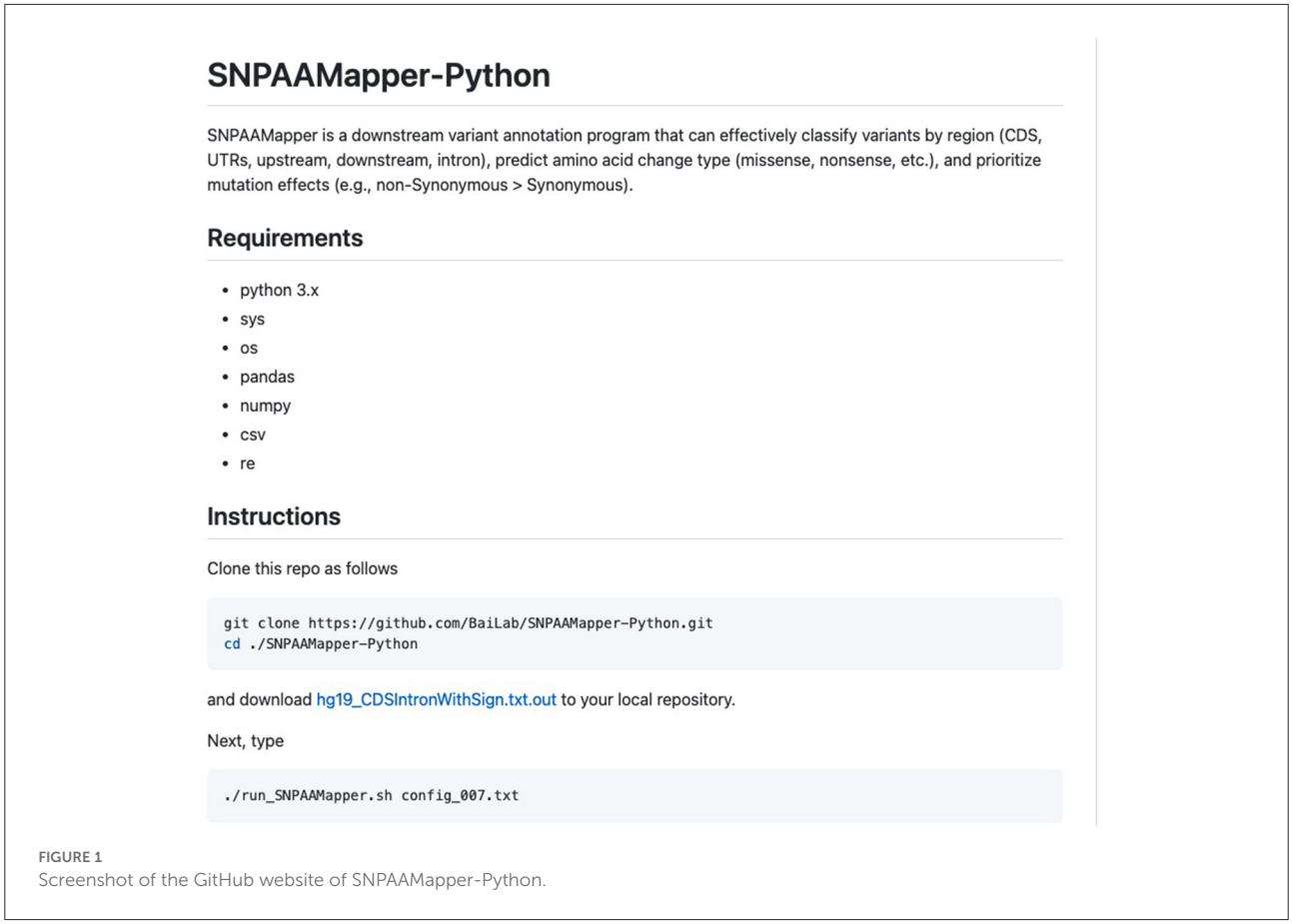


FIGURE 1
Screenshot of the GitHub website of SNPAAMapper-Python.

Usage

As the input, a VCF file is required for annotation. There are two methods to use the program, an end-to-end option and a step-by-step option. For the end-to-end option, users can use the *config.txt* file to configure the running parameters and define input files. The running parameters are "vcfFile = clinvar_20220528.vcf, intronBoundary = 6, geneAnnotation = ChrAll_knownGene.txt, conversionFile = kgXref.txt, sequenceFile = hg19_CDSIntronWithSign.txt.out." Then users can use command.*/run_SNPAAMapper-Python.sh config.txt* to generate the final output by running through each step automatically. This option is recommended for all users by default. For the step-by-step option, the users will have to run through the Python scripts step-by-step in the following orders: (1) Generate exon annotation file; (2) Process exon annotation files and generate feature start and gene mapping files (*Algorithm_preprocessing_exon_annotation_RR.py*); (3) Classify variants by regions (CDS, Upstream, Downstream Intron, UTRs...) (*Algorithm_mapping_variants_reporting_class_intronLocation_updown.py*); 4() Predict amino acid change type (*Algorithm_predicting_full_AA_change_samtools_updown.py*); (5) Prioritize mutation effects (*Algorithm_prioritizing_mutation_headerTop_updown.py*). This option is recommended for more advanced users and for users who are only interested in the intermediate outputs. The final output will be an annotated variant file, with each row representing a unique input variant and each column representing one piece of annotated information.

Results

Annotated file

For each variant in the input VCF file called by SAMTools (Li et al., 2009), there is a corresponding row in the output annotated file. The final output will be an annotated variant file, with each row representing a unique input variant and each column representing one piece of annotated information. Specifically, there 21 columns with unique annotation information. For VCF files containing individual genotype data, the first column specifies the sample ID. The other 20 columns are as follows: "Chromosome," "Variant Position," "Gene Symbol," "UCSC ID," "Strand," "AA Position

TABLE 1 Speed comparison between original and updated SNPAAMapper for the test dataset.

Steps	Python execution time in seconds	Perl execution time in seconds
Step 1	13	2
Step 2	16	166
Step 3	2	138
Step 4	62	407
Step 5	1	1
Total	94	714

of Mutation (for CDSHIT)," "Variant Type," "Amino Acid Ref (Codon) -> AA SNP (Codon)," "Variant Class," "Ref AA chain," "Alt AA chain," "Hit Type," "Known dbSNP," "Ref nt," "Alt nt," "Quality," "Depth," "Allele Freq," "Read Categories," and "Info." A table with column descriptions for the first 15 columns of the VCF output file can be found in Supplementary Table 1. The remaining five columns are variant calling information extracted from the VCF output file.

ClinVar database

ClinVar is a widely used database that links variants to their functional importance (pathogenicity) (Landrum et al., 2015). ClinVar provides a full download of their database in VCF format. Aside from potential phenotype/clinical association information, ClinVar provides some basic annotation for the variants, such as HGVS-nomenclature, types of variants (single nucleotide variant, indels, etc.), and functional consequences (missense, UTR, etc.). While this information is valuable, it is common to expand these annotations for the clinicians/researchers to better understand the functional impact of the variants for the purpose of disease diagnosis, hypothesis-generating/validation.

Currently, ClinVar (20220508) includes 1,440,883 unique variants that are associated with various diseases. It uses a 5-category classification system that groups these variants into pathogenic, likely pathogenic, benign, likely benign, and variant of uncertain significance.

We first compared the running speed of annotating the test file using the original and updated SNPAAMapper (Table 1). We then ran the updated Python version of SNPAAMapper for entire ClinVar database file and found that our updated version was able to generate exon annotation file in 48 s; generate feature start and gene mapping files in 49 s; classify variants by regions in 256 s, It took 124,311 s for predicting amino acid change type. It took 497 s for prioritizing mutation effects. For the Perl version of SNPAAMapper, it takes more than 2 weeks to run all the pipeline steps.

Next, we examined the concordance of annotation between ClinVar and SNPAAMapper-Python. We found that 91.4% of the variants have concordant annotations between ClinVar and our tool (Figure 2). Among those variants with discordant annotations, we found that 48.55% of them are annotated as non-coding transcripts variants in ClinVar which was not specifically annotated in SNPAAMapper. Additionally, our tool provided annotation for 10% of the variants that showed no annotation in ClinVar, which highlighted the usefulness of our tool. Importantly, annotations from ClinVar were buried into the "INFO" column with other information, which makes parsing and understanding the information much more difficult, whereas, for our tool, there is a separate column for each specific annotation. Using position-based annotation from SNPAAMapper, we examined the distribution of variants by functional genomics regions (Figure 3). We found that the majority (661,958 for pathogenic variants and 592,638 for benign variants) of reported variants in ClinVar (n = 947,008), regardless of their pathogenicity, reside in coding sequences (CDS).

Comparing pathogenic variants (n = 103,909) to benign ones (n = 340,726), we found that there was a higher percentage of CDS variants and lower percentages of 3'UTR, 5'UTR, and other non-coding sequences for pathogenic variants. This observation illustrated that most of the studies focused on CDS as variants from this region usually have clearer functional consequences. Additionally, using the SNPAAMapper's output, we can easily examine the distribution of variants across different genes. For example, gene TTN has the most unique variants (n = 17,915), while 2,448 genes have only 1 unique variant. These observations highlighted our need for investigating the under-studied genes to gain a well-rounded understanding of human genes and genetic mutations. We note that this gain of knowledge is attributable to the easy-to-use format of SNPAAMapper's output.

Finally, we illustrated the importance of including additional exome-specific annotations to help users interpret their data using annotated output from the previous step. First, we looked under the hood for variants residing in CDS. As illustrated in Figure 4, it is not surprising that the vast majority of nonsense (NSN) variants are pathogenic, while most synonymous (SYN) variants are benign. Interestingly, for nonsynonymous (NSM) variants, we observed a similar percentage of the reported pathogenic and benign variants. This highlighted the importance of NSM variants in helping us interpret sequencing variants. As a result, numerous methods have been developed to target NSM variants and predict whether they are functional or not.

Another key strength of our SNPAAMapper pipeline was to retrieve the most damaging amino acid variants from genomic variants. This can be used to investigate the property of variants and their impact on the biochemical and physical properties of the amino acid and protein. As illustrated in Figure 5, we

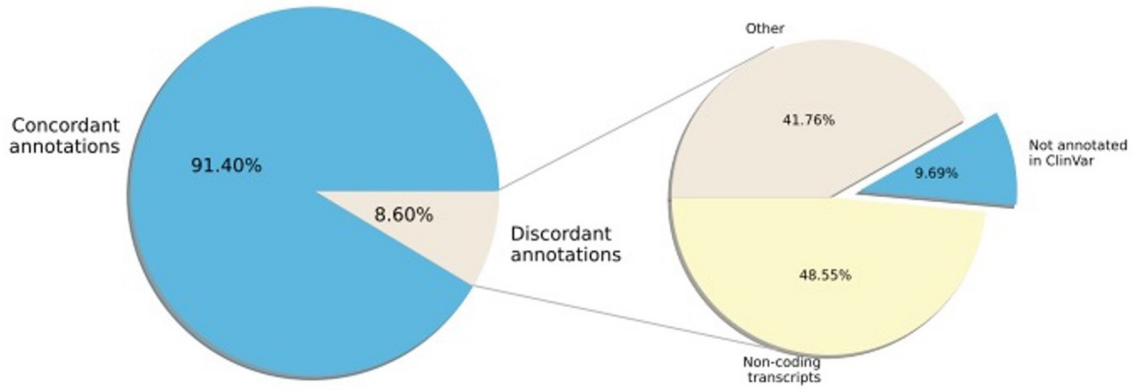


FIGURE 2
Percentage of ClinVar variant region annotation that are concordant with SNPAAMapper-Python.

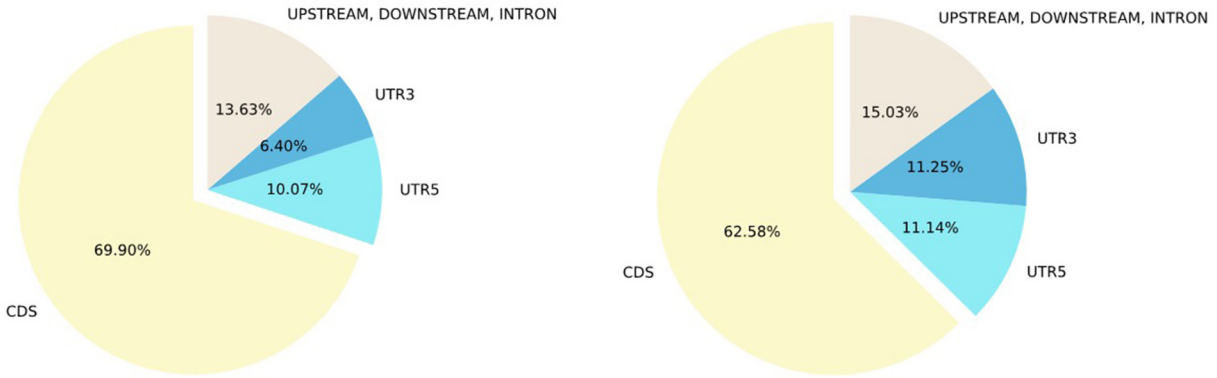


FIGURE 3
Percentage of ClinVar variants by variant type. Left, Percentage of ClinVar pathogenic variants by variant type; Right, Percentage of ClinVar benign variants by variant region.

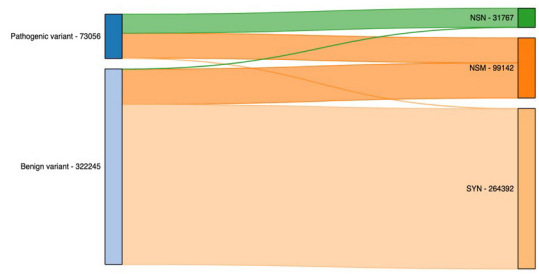


FIGURE 4
Count of pathogenic and benign variants by their functional annotation. NSN, nonsense; NSM, nonsynonymous; SYN, synonymous.

plotted the hydropathy of the variants in ClinVar database grouped by their clinical significance. We found that a change in hydropathy was more commonly observed in pathogenic variants. For example, hydrophobic to hydrophilic conversion was substantially enriched in pathogenic variants. On the other hand, benign variants were substantially enriched in variants without pathogenic conversion (hydrophilic to hydrophilic, etc.). This analysis briefly highlighted the importance of providing easy-to-access amino acid variants, as their properties are crucial in understanding the functional consequence of the underlying genomic variants.

Comparison of performance in run times

We compared the execution time between the original SNPAAMapper and updated SNPAAMapper-Python using the same sample VCF file. The updated program runs significantly faster (8 times) than the original Perl program, with an almost

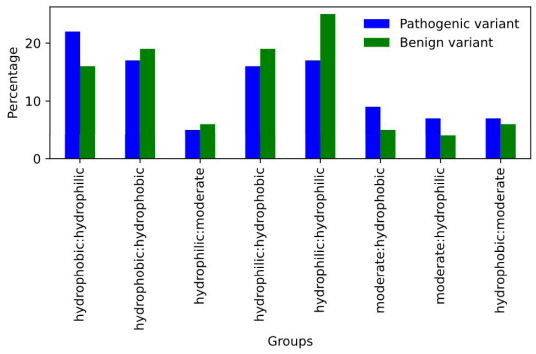


FIGURE 5
Percentage of variants grouped by their hydropathy.

10-fold increase in speed. This time increase will be substantially more prominent when hundreds or thousands of samples were queried.

Discussion

The biggest difference between the old tool (SNPAAMapper) and our updated tool (SNPAAMapper-Python) is the change in the programming language. The former runs on Perl, while the latter runs in Python, as the name states. To convert the original Perl pipeline codes, we downloaded and analyzed the original SNPAAMapper code reported in the paper, which was sourced from the previous study (Bai and Cavalcoli, 2013).

Our updated program maintains all the previous features of SNPAAMapper. It grants downstream variant identification and analysis at a record speed. Our tool is self-sufficient and lightweight; external alignment tools and such are not necessary since they are all included in this package. In addition, our tool preserves the original customizability of SNPAAMapper, meaning that it can be easily configured for other species and reference genomes. Another benefit of our program is its greater compatibility; the popularity and use of the Perl programming language are rapidly decreasing (https://www.tiobe.com/tiobe-index/perl/) while the use of Python has been growing at an extreme rate for the last decade (https://www.tiobe.com/tiobe-index/python/). We believe that this upgrade is crucial for researchers due to the impractical run time of the original SNPAAMapper on a test sample (Table 1).

The python version performs more efficiently than the perl version. The reason is that we use an optimized Python-built in module "csv" to read and write tabular data. In particular, the python version is not using any embedded loop as the Perl version by iterating over almost approximately a million rows in the ChrAll_knownGene.txt.exons file for "the number of chromosomes" (about 60) times. Instead, the python version iterates only one time.

We used the same dataset tested in SNPAAMapper for both programs. By running both tools on a Latitude 7480 Desktop computer with 8GB RAM and an Intel Core i5-6300 CPU @ 2.4Ghz, we were able to make an accurate comparison of the execution times for each program. Using this method, we were also able to make time comparisons for each step of both programs.

In addition, we also run SNPAAMapper-python on the ClinVar database file to collect the running statistics. Specifically, we ran the pipeline on an lntel_Core_i7-4770K_CPU@3.5GHz Gentoo Linux box to collect running statistics for ClinVar database file.

Additionally, with the improved output representation, this update enables easy-to-use output where each column represents a single piece of information. These improvements can greatly facilitate downstream analyses and open up opportunities for users to analyze their data using tools like Excel, which is expected to accelerate the translation of information to knowledge. Lastly, our codes are open-sourced and hosted on GitHub, which enables the continuing maintenance, updates and improvements from us and all the users.

We created an end-to-end pipeline with intermediate outputs. The final output is the one that's interesting to most of our users.

To test the ease of use and convenience of our program, we asked a student to act as a user and attempt to operate our system. As our tester, we asked the student to document running statistics and surveyed the practicality of our tool.

Our ultimate goal is to create a very efficient and multifunctional pipeline which can not only do variant annotation, but also has multiple functional annotation databases incorporated into the pipeline. This would require downloading many databases and consistently formatting them.

In the future, we plan to add additional features/annotations to the pipeline. Some examples include population allele frequencies, functional prediction scores etc. This will be a priority for us. Additionally, we expect to compare the annotations made by SNPAAMapper with other established tools in the future version to give users a better understanding of the performance of our tool. Furthermore, to make SNPAAMapper more easily accessible to a wider range of users, we plan to extend our program to support R in future development.

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

CL, KM, and YB drafted the manuscript. NX deployed the SNPAAMapper-Python GitHub website. CF assisted to modify the pipeline codes. CL has processed the input files to generate the ClinVar results. AH conducted the student test runs. XL and YB supervised the project and provided suggestions and guidance on directions. All authors participated in the discussions and revisions. All authors contributed to the article and approved the submitted version.

Funding

Funding for article processing charge is provided by the University of South Florida to XL.

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.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frai.2022.991733/full#supplementary-material

SUPPLEMENTARY FIGURE 1
Algorithm for classifying variants by genomic regions for SNPAAMapper-Python.

SUPPLEMENTARY TABLE 1
Column description for the first 15 columns of the VCF output file.

References

Bai, Y., and Cavalcoli, J. (2013). SNPAAMapper: An efficient genome-wide SNP variant analysis pipeline for next-generation sequencing data. *Bioinformation* 9, 870–872. doi: 10.6026/97320630009870

Barba, M., Czosnek, H., and Hadidi, A. (2014). Historical perspective, development and applications of next-generation sequencing in plant virology. *Viruses* 6, 106–136. doi: 10.3390/v6010106

Landrum, M., Lee, J., Benson, M., Brown, G., Chao, C., Chitipiralla, S., et al. (2015). ClinVar: public archive of interpretations of clinically relevant variants. *Nucl. Acids Res.* 44, D862–D868. doi: 10.1093/nar/gkv1222

Lewis, T. (2013). *Human Genome Project Marks 10th Anniversary*. livescience.com. Available online at: https://www.livescience.com/28708-human-genome-project-anniversary.html (accessed July 11, 2022).

Li, H. B., Handsaker, A., Wysoker, T., Fennell, J., Ruan, N., Homer, G., et al. (2009). The Sequence Alignment/Map Format and SAMtools. *Bioinformatics* 25, 2078–2079. doi: 10.1093/bioinformatics/btp352

Preston, J., VanZeeland, A., and Peiffer A. (2022). *Innovation at Illumina: The Road to the $600 Human Genome*. Nature.com. Available online at: https://www.nature.com/articles/d42473-021-00030-9

frontiers | Frontiers in Artificial Intelligence

TYPE Original Research
PUBLISHED 13 September 2022
DOI 10.3389/frai.2022.952997

OPEN ACCESS

EDITED BY
Zhaohui Steve Qin,
Emory University, United States

REVIEWED BY
Soraya Chaturongakul,
Mahidol University, Thailand
Meenakshisundaram Balasubramaniam,
University of Arkansas for Medical Sciences, United States

*CORRESPONDENCE
Kuppan Gokulan
kuppan.gokulan@fda.hhs.gov

SPECIALTY SECTION
This article was submitted to Medicine and Public Health, a section of the journal Frontiers in Artificial Intelligence

RECEIVED 25 May 2022
ACCEPTED 15 August 2022
PUBLISHED 13 September 2022

CITATION
Gokulan K, Khare S and Foley SL (2022) Structural analysis of VirD4 a type IV ATPase encoded by transmissible plasmids of *Salmonella enterica* isolated from poultry products. *Front. Artif. Intell.* 5:952997. doi: 10.3389/frai.2022.952997

Structural analysis of VirD4 a type IV ATPase encoded by transmissible plasmids of *Salmonella enterica* isolated from poultry products

Kuppan Gokulan*, Sangeeta Khare and Steven L. Foley

Division of Microbiology, National Center for Toxicological Research, U.S. Food and Drug Administration, Jefferson, AR, United States

Bacterial species have evolved with a wide variety of cellular devices, and they employ these devices for communication and transfer of genetic materials and toxins. They are classified into secretory system types I to VI based on their structure, composition, and functional activity. Specifically, the bacterial type IV secretory system (T4SS) is a more versatile system than the other secretory systems because it is involved in the transfer of genetic materials, proteins, and toxins to the host cells or other bacterial species. The T4SS machinery is made up of several proteins with distinct functions and forms a complex which spans the inner and outer membranes. This secretory machinery contains three ATPases that are the driving force for the functionality of this apparatus. At the initial stage of the secretion process, the selection of substrate molecules and processing occurs at the cytoplasmic region (also known as relaxosome), and then transfer mechanisms occur through the secretion complex. In this process, the VirD4 ATPase is the first molecule that initiates substrate selection, which is subsequently delivered to the secretory machinery. In the protein data bank (PDB), no structural information is available for the VirD4 ATPase to understand the functional property. In this manuscript, we have modeled VirD4 structure in the Gram-negative bacterium *Salmonella enterica* and described the predicted functional importance. The sequence alignment shows that VirD4 of *S. enterica* contains several insertion regions as compared with the template structure (pdb:1E9R) used for homology modeling. In this study, we hypothesized that the insertion regions could play a role in the flexible movement of the hexameric unit during the relaxosome processing or transfer of the substrate.

KEYWORDS

type IV secretion system, ATPases, *S. enterica*, transmissible plasmids, ligand docking, homology modeling, secretary mechanism

Introduction

Secretion is a central biological process in living organisms, which facilitates the transfer of chemicals, molecules, and toxins across the cell membrane. Bacterial species use multiple secretory apparatuses to facilitate the translocation of several molecules into the host cells, which helps bacterial survival and communication with other organisms in the surrounding environments (Schroder and Lanka, 2005; Fronzes et al., 2009). To date, six types of secretory systems (types I, II, III, IV, V, and VI) have been identified and characterized in the microbial world. Among these systems, the type IV secretion system (T4SS) is the most versatile, which facilitates various functions and has been observed in both Gram-positive and Gram-negative bacteria. The T4SS shares several structural and functional features with bacterial conjugation systems. The functions of T4SS in bacterial species include (1) translocation of proteins or toxins to the host cell, (2) horizontal transfer of plasmid DNA between bacteria during conjugation, and (3) uptake and release of DNA molecules that allow the exchange of DNA with the extracellular environment or host cells (Grohmann et al., 2003; Wallden et al., 2010).

The presence of T4SS machineries has been documented in several pathogenic bacteria that include *Helicobacter pylori, Streptococcus suis, Bordetella pertussis, Brucella* spp., and *Legionella pneumophila* (Kwok et al., 2007; Zhao et al., 2011). These bacterial species employ T4SS to inject virulence factors into host cells (Corbel, 1997; Ninio and Roy, 2007). Most of the studies elucidating *Salmonella* pathogenicity have been focused on serovar Typhimurium; however, there is a knowledge gap in understanding how different serovars lead to infection and whether putative virulence factors located on plasmids impact the ability of *Salmonella* to infect different hosts; for example, how *Salmonella enterica* isolates containing T4SS differ from those that lack T4SS. Recent CDC data show that non-typhoidal *Salmonella* is the leading cause of diarrhea globally, which accounts for roughly 153 million gastroenteritis cases and 57,000 deaths annually (Healy, 2020). The mode of transmission occurs *via* the consumption of contaminated food products including animal-derived products, seafood, fresh produce, and fruits (Mellou et al., 2021). *S. enterica* serovar Heidelberg is another leading serovar that mostly infects poultry (turkey and chicken) and is a major cause of severe illness in humans through the consumption of contaminated poultry products. *S.* Heidelberg strains are often resistant to several antimicrobial agents, and surveillance data show that drug-resistant strains are on the rise. National Antimicrobial Resistance Monitoring System (NARMS) data show that the percentage of *S.* Heidelberg isolates from human and poultry that are resistant to cephalosporin has been on the rise (Winokur et al., 2000) and correlates with the spread of AmpC β-lactamase. This β-lactamase is encoded by bla_{CMY} genes and is linked with transmissible plasmids. Studies have shown that *S.* Heidelberg harbor plasmids are able to transfer genes and are also responsible for multidrug resistance and virulence.

Studies have shown that certain *S. enterica* strains isolated from food-animal sources harbor transmissible plasmids (Johnson et al., 2010). In addition, multiple isolates have been shown to have transmissible plasmids that harbor T4SS encoding genes in *S. enterica* (Han et al., 2012). Moreover, the importance of T4SS encoding genes in bacterial invasion and virulence of *S. enterica* on macrophage infection was demonstrated by our group (Gokulan et al., 2013). In Gram-negative bacteria, the T4SS core complex is composed of 12 proteins (VirB1 to VirB11 and VirD4) that span across both inner and outer transmembrane domains to facilitate secretion. The T4SS complex is further divided into three groups: (i) scaffold with translocation channel; (ii) ATPases, and (iii) pilus. The plasmid sequence analysis of *S. enterica* revealed that the presence of the VirB/D4 T4SS core complex is similar to the *Agrobacterium tumefaciens* VirB/D T4SS. The sequence analysis also revealed the absence of the VirB7 sequence in the core complex in *S. enterica*. This finding was consistent with whole-genome sequence results of 44 outbreak strains of *S.* Heidelberg isolates (animal, retail meats, and human clinical isolates) that revealed the presence of transmissible plasmids that encode T4SS in 21 isolates (Hoffmann et al., 2013).

The T4SS inner membrane complex (ATPases system) contains three ATPase proteins (i.e., VirD4, VirB4, and VirB11), which are the driving force for the assembly of T4SS, substrate transfer that can include virulence factors. The VirB4 crystal structure bound with ADP has been reported elsewhere (Wallden et al., 2012). VirB4 is a highly conserved protein in the T4SS machinery and is composed of N-terminal and C-terminal domains. There is no structural information available for VirD4 of *S. enterica*; therefore, this study was undertaken to understand the structural and functional details of VirD4. In this study, we employed bioinformatics tools for the homology modeling of the T4SS machinery of *S. enterica* to understand the functional aspects of the VirD4 ATPases.

Methods

Salmonella strain and sequence selection for homology modeling

The VirD4 sequence used for homology modeling was derived from *S. enterica* strain 163 (which was isolated from an infected turkey) (pSH163_34, GenBank accession No. JX258656). Protein sequences were determined with the RAST annotation pipeline (Argonne National Laboratory, Chicago, IL, USA), and the protein identities were determined by BLAST comparisons to GenBank.

Secondary structure and protein structural fold prediction

VirD4 protein sequences were submitted to the protein fold reorganization server to predict structural folding based on the sequence and similarity in protein folding (www.sbg.bio.ic.ac.uk/phyre2) and the Swiss model (swissmodel.expasy.org) (Soding, 2005; Waterhouse et al., 2018). We compared the predicted structures from these programs and selected the best fit secondary structure and protein fold conserved template for homology model building. To analyze the confidence of the predicted model, we used sequence identity, similarity, secondary structure prediction, and structural superposition for conservation of protein fold and obtained root mean square deviation (RMSD) value during structural alignment and *Z*-score value (DALI search) (Holm and Rosenstrom, 2010). The final model was energy minimized using SYBYL (www.tripos.com), which was further inspected using WINCOOT (www.ysbl.york.ac.uk) to see if any clashes occur between side chain residues by comparing with the template model. The stereochemistry of the predicted structure was assessed with the program PROCHECK (www.ebi.ac.uk).

Structural cavity analysis by CASTp

In the PDB, no structural information is available for the VirD4 protein of T4SS machinery; therefore, there is a lack of information about the substrate binding region. In addition, the VirD4 sequence had several insertion regions compared with the template structure (bacterial conjugate coupling protein pdb 1E9R). To predict the nucleotide-binding region and exclude it from the shallow depression, cavity prediction was performed for modeled VirD4 structure. Cavities exhibit an entrance that connects the interior of protein with the outside solution or small molecules. Before submission to cavity prediction, we removed residues 1-106 from the model, which is predicted to be in the transmembrane region. To predict a probable substrate binding site, we initially employed the CASTp program that analyzes the topology of the structure and predicts the concave cavities, surface area, and location (Binkowski et al., 2003). In addition, we also used the protein structural fold search engine to identify the functional site in the homology model. The CASTp predicted concave cavity location, and the functional site predicted by the protein structural fold search engine was further validated by docking the nucleotide at the active site.

Ligand docking by CB-DOCK and 3D-ligand docking method

Based on the cavity prediction and functional site prediction, ligand docking was performed by CB-DOCK and 3D-ligand docking for validation (Liu et al., 2020). The VirD4 homology model was converted into pdbqt format for

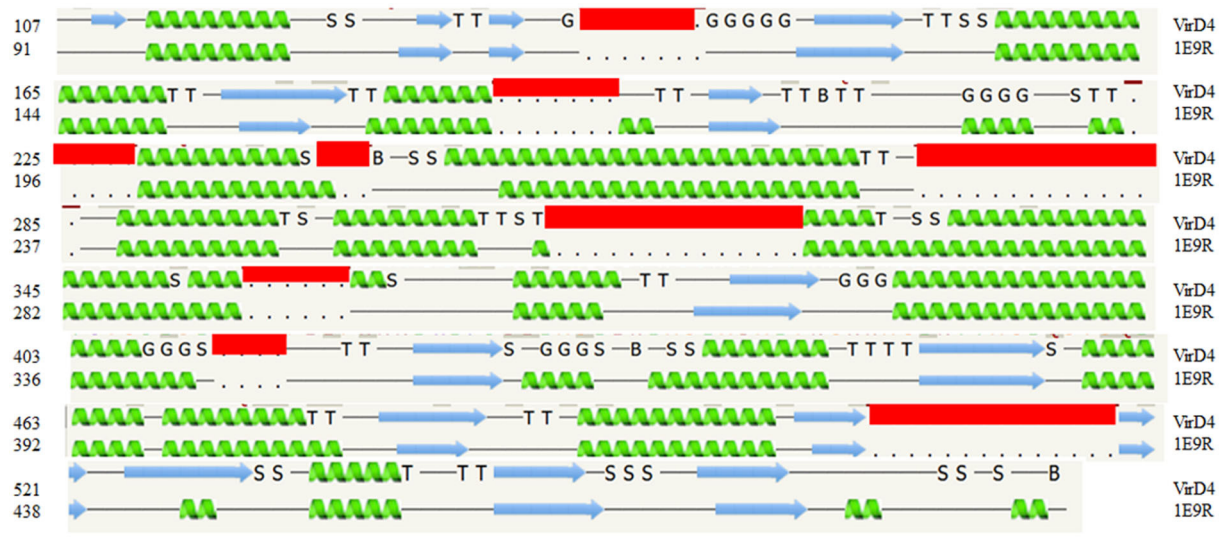


FIGURE 1
The secondary structure alignment between the template structure (1E9R) and *Salmonella enterica* VirD4 sequence. The secondary structure prediction shows that approximately 400 amino acids are aligned very well between them. The *S. enterica* VirD4 protein sequence has several insertion regions, which are shown in the red box. The α-helix prediction is shown in green, and the β-strand is shown by the blue arrow. In this figure, G-indicates the 3-turn helix, T-indicates the hydrogen-bonded turn, and S indicates the bend. The top row is VirD4 starts with residue 107, and the bottom row is template starts with residue 91(1E9R).

docking purposes. VirD4 ATPase initiates protein assembly and facilitates the secretion of toxins into the host cells in association with partner proteins. Therefore, an ADP ligand was generated and converted into an SDF file format for docking.

Hexameric structure

The fundamental functions of the VirD4 protein are to recruit the substrates and then deliver them to the secretion channel. It is also known as coupling protein that contains Walker A and B sequence motifs. These motifs play a major role in nucleotide binding and hydrolysis. The crystal structure of the cytoplasmic region of the P-loop containing nucleoside triphosphate hydrolases (1E9R) assembles to form a hexameric structure. The VirD4 protein displayed 70 to 80% conservation of secondary structure with 1E9R structure for 395 residues. The conservation of secondary structure and structural fold implicates the functional similarity between them. To construct *S. enterica* VirD4 hexameric form, we employed 1E9R hexameric structure as a template and translated modeled VirB4 into each monomer. The hexameric structure was globally energy-minimized using SYBYL. Figures were generated using the program Pymol (www.pymol.org).

Results

The protein structural fold search engine (Phyre2) predicted a few structural coordinates from the PDB based on protein sequence identity and similarity that aided as a template for homology modeling for *S. enterica* VirD4 protein, and specifically, these include P-loop containing nucleoside triphosphate hydrolases (1E9R), Type IV Coupling Complex (T4CC) from *L. pneumophila* (6SZ9), and structure of VirB4 of *Thermoanaerobacter pseudethanolicus* (4AG5) (Pena et al., 2012; Wallden et al., 2012; Meir et al., 2020). The predicted secondary structure of VirD4 protein displayed a high percentage of structure conservation with P-loop containing nucleoside triphosphate hydrolase structure (Pena et al., 2012). The predicted secondary structure of VirD4 protein was found to be around a 70 to 80% match with the template secondary structure (1E9R) for 395 residues. The sequence alignment analysis reveals that the VirD4 sequence of *S. enterica* showed 21% sequence identity and 56% sequence homology with P-loop containing nucleoside triphosphate hydrolase sequence (1E9R). The sequence alignment displayed that the VirD4 protein sequence had several insertion sequences in comparison with template structure (pdb#1E9R) (Figure 1). The protein structural fold search engine also predicted VirB4 of *T. pseudethanolicus* as a template model, which belongs to T4SS (Wallden et al., 2012); however, the secondary structure

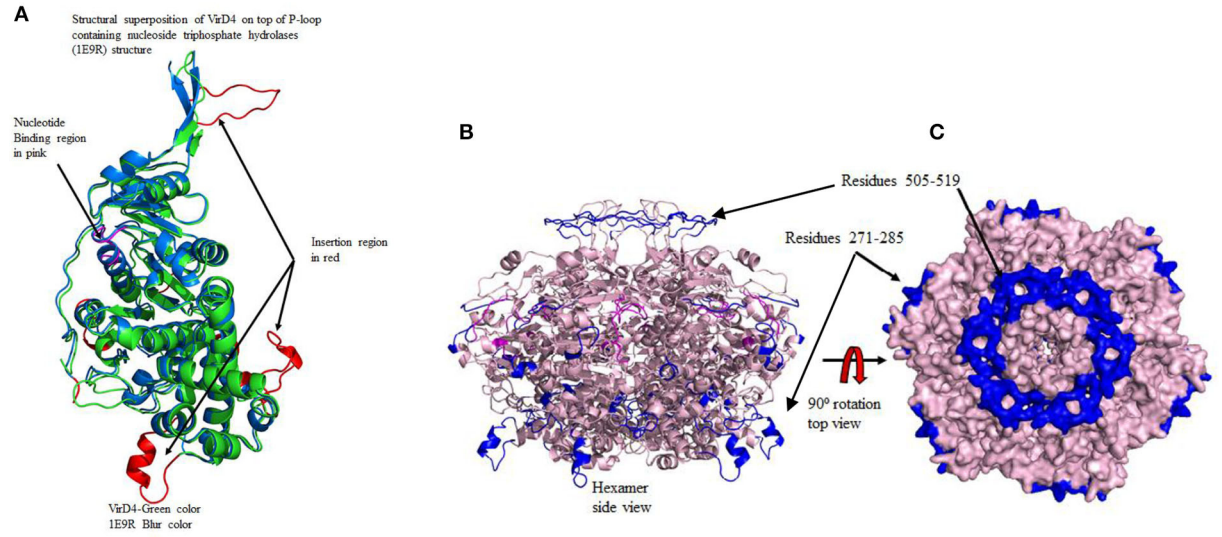


FIGURE 2
(A) The structural superposition of modeled VirD4 structure on top of 1E9R template structure. This diagram also shows the insertion regions and nucleotide binding regions in different colors. Color code: green VirD4 and cyan 1E9R template structure. The two long insertion regions are labeled as Regions I (residues 271-295) and II (residues 505-519). **(B)** The left side of the diagram shows the formation of the hexamer. The nucleotide-binding regions are located at the dimer interface shown in pink. Two long insertion regions are shown in blue, one occupies the top of the hexamer and the other is at the bottom. The transmembrane binding region forms a circular disc to insert into the membrane. **(C)** The right surface diagram shows the top view. In the hexameric form, the opening of the tunnel is much wider near the transmembrane region, whereas, in the cytoplasmic region, the opening is very narrow. The two insertion regions are also shown in blue. At the transmembrane insertion region, all six monomers connected to each other forming a ring-like disc.

prediction and structural alignment displayed less conservation in comparison with 1E9R coordinates. Therefore, the 1E9R coordinate was used for homology modeling of the VirD4 structure based on the secondary structure conservation and similarity in the structural fold. Although VirD4 had only 21% sequence identity with 1E9R coordinates, the structural fold was highly similar (Figure 2A). Approximately 70% of the predicted model was built with more than 90% confidence.

For the structural analysis, we deleted the N-terminal transmembrane region of VirD4 protein residues 1-116 and the last 40 amino acids (580-620) from the C-terminal region due to low confidence in the homology model building. Then, we superimposed C-α carbon atoms of the VirD4 homology model on top of 1E9R coordinates for structural analyses. The VirD4 C-α carbon atoms of residues from 107 to 474 were superimposed on top of the template structure (1E9R) and C-α carbon atom residues from 91 to 491 with RMSD 0.6 Å (Figure 2A). The structural alignment revealed that VirD4 structure insertion regions occupy the connecting loop and are located away from the core structure (shown in red color in Figure 2A). Most of the insertions are between 4 and 6 residues except two regions (Table 1). Two insertion regions were around 10 to 15 residues long, and one insertion was positioned in the connecting loop at the bottom of the hexamer (Figure 2A shown in red color). The second region occupies the top of the core structure (left side cartoon diagram shown in blue in Figure 2B) that forms a donut-like structure highlighted in blue (right side surface diagram shown in Figure 2C). The homology model was minimized, and the quality of the structure was inspected for clashes, rotamers, and amino acid geometry (Ramachandran plot), and all were in acceptable ranges. We also generated a hexameric form of the VirD4 model, which forms a ring-like structure (Figure 2B). The final VirD4 homology model was submitted to DALI search for structural alignment prediction from the PDB. The DALI search predicted several structures from the PDB; however, the 1E9R structure was the top-most structure with a *Z*-score of 37.4%, with a low RMSD, and a clear separation from the remaining predicted structures which all had very low Z-scores with higher RMSD.

The homology structure of VirD4 contains two domains that include an α-helical domain and β-strands surrounded by α-helices or a nucleotide binding region, which is very similar to the 1E9R structure (Figure 2A). The sequence alignment shows that the C-terminal region contains highly conserved amino acids as compared with the N-terminal region. Earlier it was shown that VirD4 is essential machinery in first recruiting the substrate and subsequently transferring the substrate to VirB11; therefore, VirD4 is known as a coupling protein. Due to its role in secretory pathways, we analyzed the functional property of VirD4 protein structure by analyzing surface topology to predict a cavity, which could be a probable nucleotide-binding pocket (CASTp server). The surface topology characterization predicted two pockets in the VirD4 structure with a surface area of 1,395 Å (Pocket-1) and 2,539 Å (Pocket-2) (Figure 3A). The boundary of Pocket-1 is well-defined with a concave surface located in the region of β-strands surrounded by α-helices, whereas the predicted Pocket-2 boundary is scattered and occupies a larger surface area. We also analyzed the functional sites of the VirD4 structure of *S. enterica* using a 3D ligand docking server and predicted the probable nucleotide-binding region (Figure 3B, which is shown in red color). 3D-ligand docking server predicted 5 binding clusters, and among them, clusters 3 and 5 predicted the Walker A and B motifs; however, predicted cluster 3 was found to be a more accurate and superimposed well with the earlier solved crystal structure. The 3D-ligand predicted cluster and CB-DOCK predicted nucleotide binding site matched well with Pocket-1 predicted by CASTp. The ligand binding cavity is surrounded by several charged residues with interacting distances, and these residues overlap with functional site residues identified by CASTp, CB-DOCK, and 3D-ligand dock predicated region (Figure 3C). The 3D-ligand docking predicts the probability of active site residues (probability score 0.33 or above of a residue is considered involving in binding) that are involved in ligand binding. Table 1 shows the list of residues that can participate in nucleotide binding and shows the solvent accessibility of these residues. In addition, the table provides the conservation information of the binding residues. The 3D-ligand cluster prediction was based on the binding of 15 ligands (ADP: 5, SO4: 7, and GNP: 3) (Figure 3D).

Then, we analyzed conserved residues of the active site in comparison with other ATPases of bacterial secretory systems. Like other ATPases, VirD4's predicted functional site residues possess conserved nucleotide binding motifs (Walker A and B motifs, Supplementary Table 1). We also analyzed the presence of Walker A and B motifs from other related ATPases as well. The result shows that these motifs are highly conserved and aligned with other bacterial ATPases that include the family of conjugative coupling factor, conjugation transfer

TABLE 1 The position of the insertion region and the number of residues in each region.

VirD4 Insertion region	Residues
136–142	KDKKIIR
189–195	SLIRKVI
224–228	SEGFN
271–285	NDKAGLKTLDIEPV
312–325	SELRGKTLADI
355–360	ANPNVA
411–415	MPTD
505–519	TIGSKSKSRSRGGTS

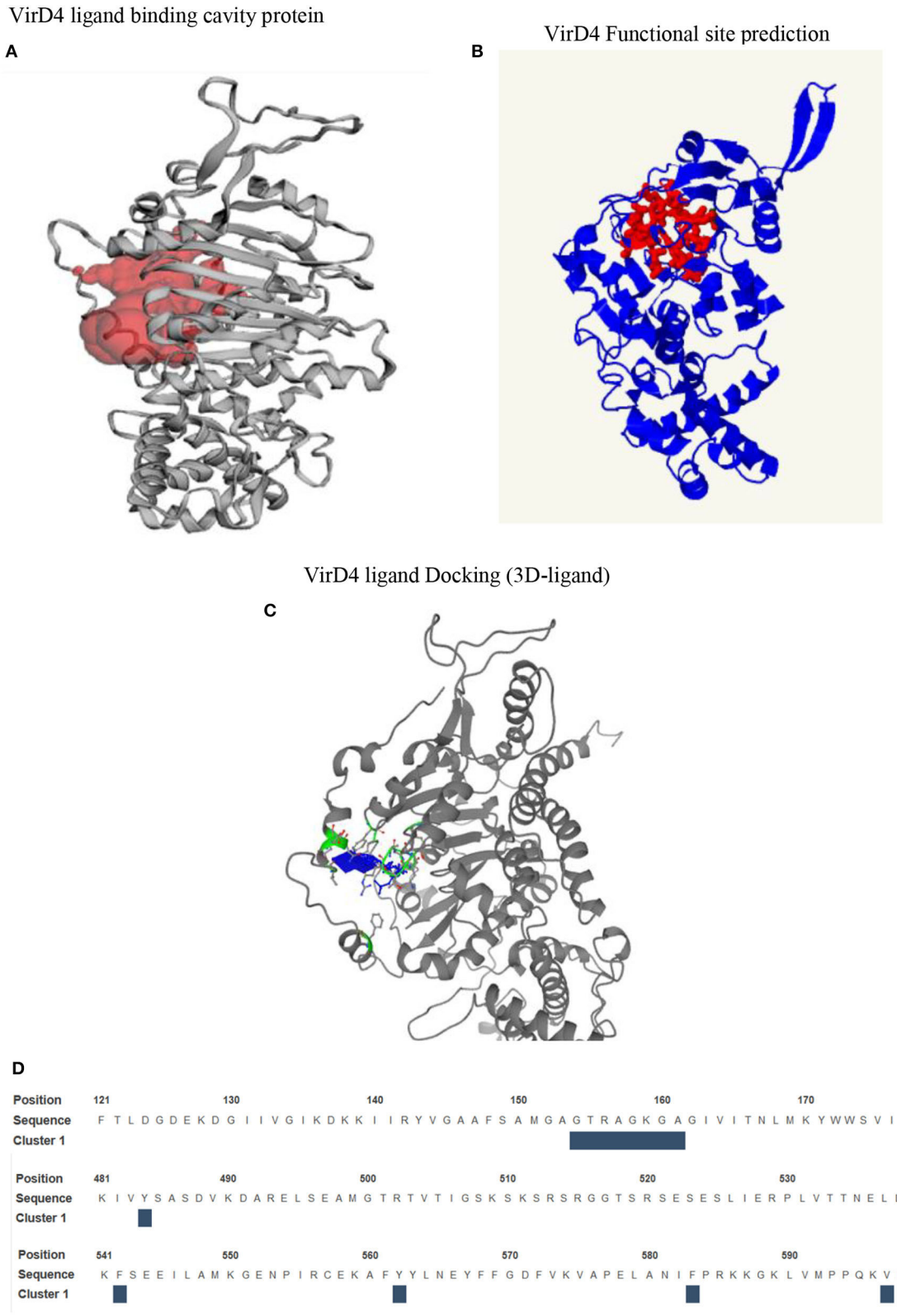


FIGURE 3
(A) Surface topology predicts the cavity on VirD4 structure. In this figure, we have shown the well-defined cavity that occupies the N-terminal region between the β-strands and is surrounded by α-helices. **(B)** The prediction of functional residues that are likely involved in substrate binding. **(C)** The nucleotide-binding region is shown in blue, which is surrounded by Walker A and B motifs, shown in green with stick representation. **(D)** This sequence shows the location of Walker A and B motifs. The interacting residues are shown above the blue box.

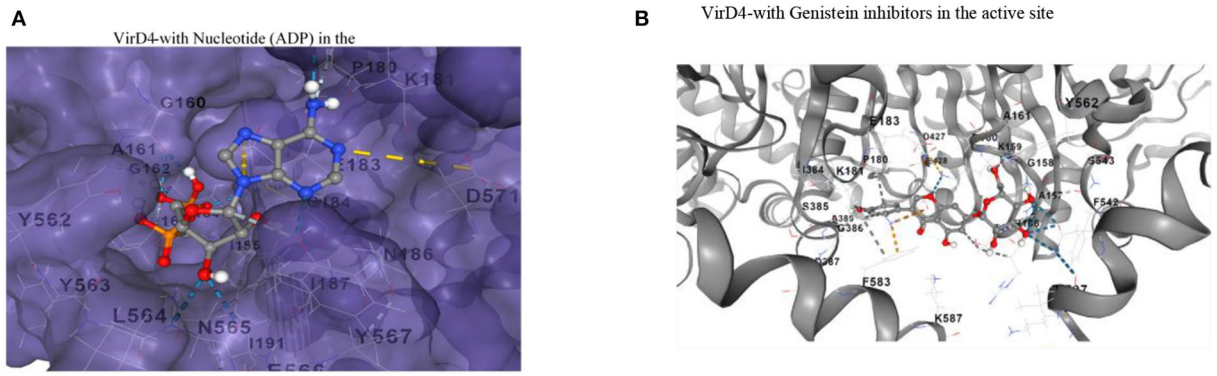


FIGURE 4
(A) The substrate binding pocket in which the ADP molecule is docked in the active site and represented in the stick model. The interacting residues are shown and labeled in the active site cavity. The α-and β-phosphate groups of ADP are surrounded by P-loop binding residues. The adenine and ribose molecules are interacting with several charged residues. **(B)** Binding of genistein in the active site cavity of VirD4 structure on docking. The interacting residues are very similar for both ADP and genistein. Part of the molecule is surrounded by Walker A structure, a very similar phosphate group of ADP. Walker A and B motifs are mostly involved in interacting with genistein.

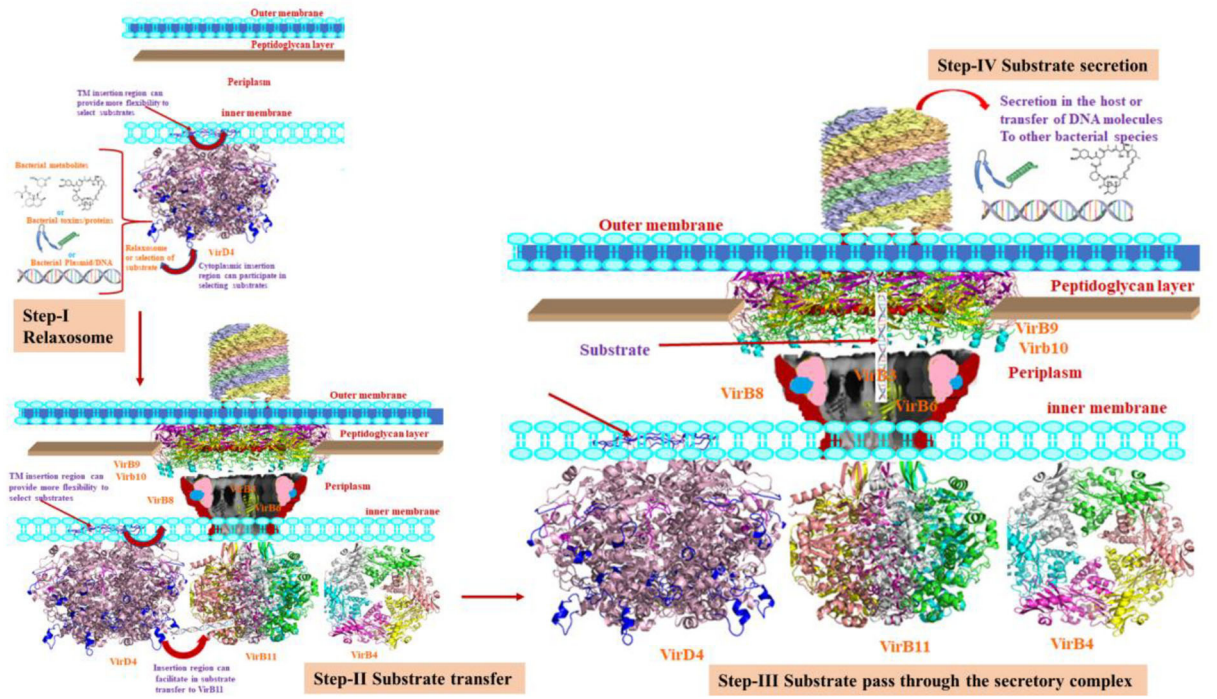


FIGURE 5
The various steps of the secretion process. Step-I shows the relaxosome where VirD4 recruits the substrate. During this process, the insertion regions may allow flexible movements. Step II shows the transfer of substrate to VirB11. This cartoon also depicts VirD4 possibly interacting with VirB11. Step III shows the translocation of the substrate through the secretory machinery. Step-IV shows the secretory products to the host cells to transfer to bacterial species.

system, and TraN, a hallmark protein of the F-type IV secretion system (Supplementary Table 2). The docked ligand is surrounded by Walker A and B motif sequence indicating the functional role as an ATPase (Figure 4A). This docked conformation is positioned to interact with several conserved residues at the active site, and they are comparable with nucleotide-bound VirB4 and other ATPases (Wallden et al., 2012). The phosphate groups of ADP ligand are surrounded by residues of Gly-Thr-Arg-Ala-Gly-Lys-Gly-Ala-Gly-Iso-Val-Iso, Tyr562, and Tyr563 like other ATPases. Specifically, α-

and β-phosphates interact with backbone nitrogen amides of the P-loop of the Walker A-motif. Similarly, adenine ring and ribose sugar molecules are surrounded by several charged residues including Lys181, Arg182, Glu183, Asn186, Asn565, and Asp571 positioned within hydrogen bonding distances (Figure 4A). Ligand docking showed several orientations of the ligand with energy minimization scores, and Figure 4A shows the best-fitted ligand with lowest ΔG energy (-8.3 kcal/mol). We also docked genistein (a known RecA inhibitor) at the predicted substrate binding cavity. The docking results show that the 5-hydoxy-3-(4-hydroxyphenyl)-7 group of genistein is surrounded by Walker A motif sequence very similar to phosphate groups of nucleotides (ADP). The hydroxyphenyl group is similarly occupied to the adenine ring of nucleotide, and interacting residues Pro180, Lys181, and Glu183 are very similar for both nucleotide and genistein (Figure 4B).

Discussion

The T4SS is large multiprotein machinery spanning the inner and outer membranes of Gram-negative bacteria. This system is more versatile compared with other types of secretion systems and is involved in DNA conjugation between two cells of the same bacterial taxa, injecting proteins or toxins to the host cells or other bacterial species and contributing to the release or uptake of genetic material (Lederberg and Tatum, 1953; Lawley et al., 2003; Backert and Meyer, 2006). The bacterial conjugation system has been linked to multidrug resistance due to horizontal DNA transfer, which poses a threat to human health (Leclercq et al., 1988; Douard et al., 2010). The structural components of the bacterial T4SS are ATPases, channel-forming multi-proteins, and pilus (Alvarez-Martinez and Christie, 2009; Bhatty et al., 2013). Each component arbitrates a specific biological function during the secretion process. T4SS contains three ATPases that include VirB4, VirB11, and VirD4, which are responsible for powering the secretory machinery on binding with nucleotide. The next major structural component is the translocation channel formed by several proteins including VirB3, B6, B7, B8, B9, and B10, which facilitate the translocation of toxins or genetic material to the host cells or other bacteria. The pilus is an extracellular structure located on the outer membrane, and it helps with the adhesion function during interaction with host cells. Characterizing the function of individual plasmid-encoded genes and proteins involved in secretion will provide an improved understanding of the structural basis of antimicrobial resistance and the molecular mechanism of pathogenesis.

In *S. enterica,* VirD4 consists of 640 amino acids, which form N-terminal and C-terminal domains. The amino acid sequences are highly variable at the N-terminal, but C-terminal has more conserved residues like other bacterial VirD4 proteins. The monomeric form retains structural folds very similar to P-loop containing nucleoside triphosphate hydrolase structure (Pena et al., 2012). The only difference is that the VirD4 structure has several insertion regions, and most of them occupy the outer surface of the core structure except amino acids 505-519. An earlier study showed the stoichiometry of VirD4, and it consists of six subunits assembled as a hexamer (Pena et al., 2012). In the generated hexameric form, the insertion region residues 505-519 occupy the top of the core structure and form a donut-like ring structure. In the hexameric form, the insertion region could (a) contribute to attachment with the inner transmembrane, (b) provide more flexibility to interact with a partner protein, and (c) recruit substrates. Likewise, residues 271-285 occupy the outer surface, and the substrate selection is mostly directed by the interaction between the relaxosome and coupling protein. Based on its location of residues 271-285, we proposed that it could contribute to the interaction with relaxosome or VirB11 during the transfer of substrate to the secretary channel. These insertions are absent in P-loop containing nucleoside triphosphate hydrolases. The hexameric structure has a wide opening (20Å) near the transmembrane region, whereas, in the cytoplasmic region, the opening is narrower (14Å). In the hexameric structure, the nucleotide-binding regions are occupied between two monomer interfaces with a wide cavity to enter the nucleotide. This is consistent with earlier reported crystal structures (Wallden et al., 2012). The α- and β-phosphates of ADP are at a favorable distance to interact with backbone amines of the P-loop of the Walker motif, which lines other ATPases. Genistein docking reveals that the 5-hydroxy-3-(4-hydroxyphenyl) group occupies the same location as that of α-and β-phosphates of ADP, and its hydroxyl groups interact with the P-loop of the Walker motif. The genistein binding location and ADP binding location are very similar in the active site cavity. In addition, the interacting residues are also very similar for both ligands. Earlier studies have shown that genistein inhibits RecA ATPase; therefore, we proposed that genistein could probably inhibit VirD4 ATPase as well. However, this hypothesis needs to be experimentally verified.

Conclusion

The VirD4 protein had a 21% sequence identity with P-loop containing nucleoside triphosphate hydrolase structure. Although it has several insertion regions, the structural fold is very similar, which indicates the functional and structural conservation between them. Based on the location of two insertion regions, we hypothesized that it could provide more flexibility to interact with partner proteins during substrate transfer to VirB11 (Figure 5). The ligand or inhibitor docking reveals that Walker A and B motifs are involved in

ligand binding. The proposed hypothesis needs to be biochemically validated.

Data availability statement

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

Author contributions

Conceptualization: KG and SF. Methodology and analysis: KG and SK. Writing—original draft and writing—review and editing: KG, SK, and SF. All authors contributed to the article and approved the submitted version.

Acknowledgments

The authors would like to thank Drs. Jyotshnabala Kanungo and Jing Han for reviewing the manuscript and providing valuable comments and suggestions.

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

Author disclaimer

This manuscript reflects the views of the authors and does not necessarily reflect those of the U.S. Food and Drug Administration.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frai.2022.952997/full#supplementary-material

SUPPLEMENTARY TABLE 1
Probability scores and conservation values of residues that are involved in ligand binding on docking.

SUPPLEMENTARY TABLE 2
The Walker A and B motif sequences. The motif search reveals that the Walker A and B motifs are highly conserved in bacterial ATPases that belong to the TRAG family, conjugal transfer proteins, and putative T4SS pathway proteins.

References

Alvarez-Martinez, C. E., and Christie, P. J. (2009). Biological diversity of prokaryotic type IV secretion systems. *Microbiol. Mol. Biol. Rev.* 73, 775–808. doi: 10.1128/MMBR.00023-09

Backert, S., and Meyer, T. F. (2006). Type IV secretion systems and their effectors in bacterial pathogenesis. *Curr. Opin. Microbiol.* 9, 207–217. doi: 10.1016/j.mib.2006.02.008

Bhatty, M., Laverde Gomez, J. A., and Christie, P. J. (2013). The expanding bacterial type IV secretion lexicon. *Res. Microbiol.* 164, 620–639. doi: 10.1016/j.resmic.2013.03.012

Binkowski, T. A., Naghibzadeh, S., and Liang, J. (2003). CASTp: computed atlas of surface topography of proteins. *Nucleic Acids Res.* 31, 3352–3355. doi: 10.1093/nar/gkg512

Corbel, M. J. (1997). Brucellosis: an overview. *Emerging Infect. Dis.* 3, 213–221. doi: 10.3201/eid0302.970219

Douard, G., Praud, K., Cloeckaert, A., and Doublet, B. (2010). The Salmonella genomic island 1 is specifically mobilized in trans by the IncA/C multidrug resistance plasmid family. *PLoS ONE* 5:e15302. doi: 10.1371/journal.pone.0015302

Fronzes, R., Christie, P. J., and Waksman, G. (2009). The structural biology of type IV secretion systems. *Nat Rev Microbiol.* 7, 703–714. doi: 10.1038/nrmicro2218

Gokulan, K., Khare, S., Rooney, A. W., Han, J., Lynne, A. M., and Foley, S. L. (2013). Impact of plasmids, including those encodingVirB4/D4 type IV secretion systems, on Salmonella enterica serovar Heidelberg virulence in macrophages and epithelial cells. *PLoS ONE* 8:e77866. doi: 10.1371/journal.pone.0077866

Grohmann, E., Muth, G., and Espinosa, M. (2003). Conjugative plasmid transfer in gram-positive bacteria. *Microbiol. Mol. Biol. Rev.* 67, 277–301. doi: 10.1128/MMBR.67.2.277-301.2003

Han, J., Lynne, A. M., David, D. E., Tang, H., Xu, J., Nayak, R., et al. (2012). DNA sequence analysis of plasmids from multidrug resistant Salmonella enterica serotype Heidelberg isolates. *PLoS ONE* 7:e51160. doi: 10.1371/journal.pone.0051160

Healy, J. M. (2020). *Travel Realted Infectious Diseases. Center for Disease Control and Prevention Chapter* 4.

Hoffmann, M., Luo, Y., Lafon, P. C., Timme, R., Allard, M. W., Mcdermott, P. F., et al. (2013). Genome Sequences of Salmonella enterica Serovar Heidelberg Isolates Isolated in the United States from a Multistate Outbreak of Human Salmonella Infections. *Genome Announc.* 1:e00004-12. doi: 10.1128/genomeA.00004-12

Holm, L., and Rosenstrom, P. (2010). Dali server: conservation mapping in 3D. *Nucleic. Acids Res.* 38, W545–549. doi: 10.1093/nar/gkq366

Johnson, T. J., Thorsness, J. L., Anderson, C. P., Lynne, A. M., Foley, S. L., Han, J., et al. (2010). Horizontal gene transfer of a ColV plasmid has resulted in a dominant avian clonal type of Salmonella enterica serovar Kentucky. *PLoS ONE* 5:e15524. doi: 10.1371/journal.pone.0015524

Kwok, T., Zabler, D., Urman, S., Rohde, M., Hartig, R., Wessler, S., et al. (2007). Helicobacter exploits integrin for type IV secretion and kinase activation. *Nature* 449, 862–866. doi: 10.1038/nature06187

Lawley, T. D., Klimke, W. A., Gubbins, M. J., and Frost, L. S. (2003). F factor conjugation is a true type IV secretion system. *FEMS Microbiol. Lett.* 224, 1–15. doi: 10.1016/S0378-1097(03)00430-0

Leclercq, R., Derlot, E., Duval, J., and Courvalin, P. (1988). Plasmid-mediated resistance to vancomycin and teicoplanin in Enterococcus faecium. *N. Engl. J. Med.* 319, 157–161. doi: 10.1056/NEJM198807213190307

Lederberg, J., and Tatum, E. L. (1953). Sex in bacteria; genetic studies, 1945-1952. *Science.* 118, 169–175. doi: 10.1126/science.118.3059.169

Liu, Y., Grimm, M., Dai, W. T., Hou, M. C., Xiao, Z. X., and Cao, Y. (2020). CB-Dock: a web server for cavity detection-guided protein-ligand blind docking. *Acta. Pharmacol. Sin.* 41, 138–144. doi: 10.1038/s41401-019-0228-6

Meir, A., Mace, K., Lukoyanova, N., Chetrit, D., Hospenthal, M. K., Redzej, A., et al. (2020). Mechanism of effector capture and delivery by the type IV secretion system from Legionella pneumophila. *Nat. Commun.* 11, 2864. doi: 10.1038/s41467-020-16681-z

Mellou, K., Gkova, M., Panagiotidou, E., Tzani, M., Sideroglou, T., and Mandilara, G. (2021). Diversity and resistance profiles of human non-typhoidal salmonella spp. in greece, 2003-2020. *Antibiotics* 10, 983. doi: 10.3390/antibiotics10080983

Ninio, S., and Roy, C. R. (2007). Effector proteins translocated by Legionella pneumophila: strength in numbers. *Trends Microbiol.* 15, 372–380. doi: 10.1016/j.tim.2007.06.006

Pena, A., Matilla, I., Martin-Benito, J., Valpuesta, J. M., Carrascosa, J. L., De La Cruz, F., et al. (2012). The hexameric structure of a conjugative VirB4 protein ATPase provides new insights for a functional and phylogenetic relationship with DNA translocases. *J. Biol. Chem.* 287, 39925–39932. doi: 10.1074/jbc.M112.413849

Schroder, G., and Lanka, E. (2005). The mating pair formation system of conjugative plasmids-a versatile secretion machinery for transfer of proteins and DNA. *Plasmid* 54, 1–25. doi: 10.1016/j.plasmid.2005.02.001

Soding, J. (2005). Protein homology detection by HMM-HMM comparison. *Bioinformatics* 21, 951–960. doi: 10.1093/bioinformatics/bti125

Wallden, K., Rivera-Calzada, A., and Waksman, G. (2010). Type IV secretion systems: versatility and diversity in function. *Cell. Microbiol.* 12, 1203–1212. doi: 10.1111/j.1462-5822.2010.01499.x

Wallden, K., Williams, R., Yan, J., Lian, P. W., Wang, L., Thalassinos, K., et al. (2012). Structure of the VirB4 ATPase, alone and bound to the core complex of a type IV secretion system. *Proc. Natl. Acad. Sci. USA.* 109, 11348–11353. doi: 10.1073/pnas.1201428109

Waterhouse, A., Bertoni, M., Bienert, S., Studer, G., Tauriello, G., Gumienny, R., et al. (2018). SWISS-MODEL: homology modelling of protein structures and complexes. *Nucleic Acids Res.* 46, W296–W303. doi: 10.1093/nar/gky427

Winokur, P. L., Brueggemann, A., Desalvo, D. L., Hoffmann, L., Apley, M. D., Uhlenhopp, E. K., et al. (2000). Animal and human multidrug-resistant, cephalosporin-resistant salmonella isolates expressing a plasmid-mediated CMY-2 AmpC beta-lactamase. *Antimicrob. Agents Chemother.* 44, 2777–2783. doi: 10.1128/AAC.44.10.2777-2783.2000

Zhao, Y., Liu, G., Li, S., Wang, M., Song, J., Wang, J., et al. (2011). Role of a type IV-like secretion system of Streptococcus suis 2 in the development of streptococcal toxic shock syndrome. *J. Infect. Dis.* 204, 274–281. doi: 10.1093/infdis/jir261

frontiers | Frontiers in Big Data

TYPE Methods
PUBLISHED 19 October 2022
DOI 10.3389/fdata.2022.965619

OPEN ACCESS

EDITED BY
Ramin Homayouni,
Oakland University William Beaumont School of Medicine, United States

REVIEWED BY
Daizong Ding,
Fudan University, China
Sujoy Roy,
Oakland University, United States

*CORRESPONDENCE
Islam Akef Ebeid
iaebeid@ualr.edu

SPECIALTY SECTION
This article was submitted to Medicine and Public Health, a section of the journal Frontiers in Big Data

RECEIVED 09 June 2022
ACCEPTED 20 September 2022
PUBLISHED 19 October 2022

CITATION
Ebeid IA (2022) MedGraph: A semantic biomedical information retrieval framework using knowledge graph embedding for PubMed.
Front. Big Data 5:965619.
doi: 10.3389/fdata.2022.965619

MedGraph: A semantic biomedical information retrieval framework using knowledge graph embedding for PubMed

Islam Akef Ebeid*

Department of Information Science, University of Arkansas at Little Rock, Little Rock, AR, United States

Here we study the semantic search and retrieval problem in biomedical digital libraries. First, we introduce MedGraph, a knowledge graph embedding-based method that provides semantic relevance retrieval and ranking for the biomedical literature indexed in PubMed. Second, we evaluate our approach using PubMed's Best Match algorithm. Moreover, we compare our method MedGraph to a traditional TF-IDF-based algorithm. Third, we use a dataset extracted from PubMed, including 30 million articles' metadata such as abstracts, author information, citation information, and extracted biological entity mentions. We pull a subset of the dataset to evaluate MedGraph using predefined queries with ground truth ranked results. To our knowledge, this technique has not been explored before in biomedical information retrieval. In addition, our results provide some evidence that semantic approaches to search and relevance in biomedical digital libraries that rely on knowledge graph modeling offer better search relevance results when compared with traditional methods in terms of objective metrics.

KEYWORDS

knowledge graph, natural language processing, information retrieval, biomedical digital libraries, graph embedding

1. Introduction

1.1. PubMed

PubMed is the National Library of Medicine's (NLM) free authoritative database of citations and search engine of more than 30 million articles in biology, medicine, pharmacy, and life sciences and across multiple curated databases such as MEDLINE[1]. PubMed is used by more than 2.5 million users each day, serving clinicians, physicians, researchers, and students (Fiorini et al., 2018). It is worth mentioning that PubMed is a database of citations, not a database of full-text articles. About two-thirds of the articles indexed in PubMed do not provide access to full texts[2]. Instead, when a free full text is available by the publisher, published

1 http://www.nlm.nih.gov/pubs/factsheets/medline.html

2 https://pubmed.ncbi.nlm.nih.gov/

as open access, or supported by a National Institutes of Health (NIH)[3] grant, the full article gets indexed in PubMed Central[4], NLM's accessible repository of full-text articles. Accordingly, the PubMed search engine relies on metadata and citations instead of parsing full-text articles when providing a search experience. Articles' metadata are indexed and parsed in fields to be utilized in the search process. Metadata fields include titles, abstracts, authors, journal names, publication dates, submission dates, related Medical Subject Headings (MeSH)[5] terms, citation and references information, funding grants, and projects.

PubMed uses an algorithm that relies on fuzzy string matching to match the query with relevant citations. For example, when a user enters in the search box an author name followed by a journal name, all the articles that author published in that journal will appear. In addition, PubMed uses the Automatic Term Mapping system (ATM) (Thirion et al., 2009). The ATM system expands the input query and finds which fields the query entered intended. The expanded query is then matched with the most relevant documents using MeSH terms, keywords, and other metadata that could be treated as an index. The most relevant articles are then retrieved using the Term Frequency-Inverse Document Frequency (TF-IDF) algorithm Jones (1972) and ranked based on date or alphabetically using either the title or the author name (Fiorini et al., 2018). Other methods include ranking by date or author information. Recently PubMed deployed its newest relevance ranking algorithm named BestMatch (Fiorini et al., 2018).

BestMatch relies on a machine learning model trained on features extracted from user search logs on PubMed in the past several years. The system has been shown to outperform TF-IDF-based ranking. However, BestMatch does not consider that the user query logs that the system has been trained on contain ambiguous queries. In addition, even though the authors evaluated the system thoroughly using an A/B testing approach with real users to evaluate the ranking quality, the algorithm did not provide solutions for the problem of understanding query intentions through semantic models. For example, a user can enter the word "cancer" in the PubMed search box, and they might mean multiple things by "cancer". For instance, they might want an article in the journal named "Nature: Cancer". Alternatively, they might want authors who work and publish in the field of cancer. Or, they might want all relevant articles that mention cancer or research done in the field of cancer. They might also be looking for a specific citation with a title or author name, journal, and year. Alternatively, they might be looking for several articles related to cancer. Search engines and information retrieval systems such as PubMed and Google rely on objective metrics and algorithms to rank their search results. The ranking of the search results does not necessarily reflect what the user meant by the query. They, however, reflect the most objective relevance based on the text of the input query. That is done by analyzing the frequency of the strings in the input queries in the corpus of documents. In addition, other models incorporate the citation network of the documents, such as PageRank in the case of Google (Page et al., 1999). Hence, integrating semantics in search algorithms and information retrieval systems, especially in biomedical literature searches, is crucial to move toward systems that can sort out ambiguity, understand query intentions, and aid in true knowledge discovery.

In recent years and the Web 2.0 information revolution, Semantic Web technologies have proliferated (Berners-Lee et al., 2001). Semantic web technologies aim to create an understandable and readable web by machines. The graph model was introduced to represent knowledge in web pages semantically using standards such as the Resource Descriptor Framework (Lassila and Swick, 1998). The idea was driven by earlier work in digital ontology and concept maps. Knowledge graphs were then born as a data model used to store information and data semantically. Knowledge graphs have also been extended as graph databases for data persistence as it allows for a more flexible representation of data and relationships than the relational data model (Hogan et al., 2021).

1.2. Contribution

To help investigate the challenges associated with semantic understanding of queries when searching the biomedical literature in PubMed, we introduce MedGraph, a knowledge graph-based search engine and information retrieval method. MedGraph relies on converting the metadata associated with PubMed into a knowledge graph. The metadata includes disambiguated author names, grant information, MeSH terms, citation information, and a dataset of extracted bio entities such as drugs, genes, proteins, and species from the text of the title and the abstract of each article in PubMed. The dataset was introduced by Xu et al. (2020), and it includes NIH project involvement for each author and each article in PubMed. In addition, it has extracted biological entities using deep learning named entity recognition technique called BioBERT (Lee et al., 2020). The dataset is available as a relational database linked using each article's unique identifier PMID. The dataset contains articles from the year 1781 until December 2020. To prove the utility of MedGraph, we extracted a small dataset of 2,696 articles and their associated metadata and citation network from the PubMed dataset (Xu et al., 2020). We then extracted the entities from the dataset and linked them semantically as a knowledge graph. We then used a knowledge graph embedding method named Node2vec (Grover and Leskovec, 2016) to extract semantic features and embed the extracted knowledge

3 https://www.nih.gov/

4 https://pubmed.ncbi.nlm.nih.gov/

5 https://www.nlm.nih.gov/mesh/meshhome.html

graph in a Euclidean space. We then used the node vectors to rank the articles using a cosine distance similarity measure on the learned vectors according to the input query after pooling all the vectors of related first-order neighbor nodes for each article. On the query side, first, the input query is parsed and expanded using the extracted biological entities in the original dataset as an index. The expanded query is then matched to their corresponding nodes in the knowledge graph. The matched node vectors are then averaged to vectorize each query.

Using various metrics, we evaluate the proposed method against PubMed's BestMatch algorithm as ground truth. In addition, we compare our method with a traditional TF-IDF approach (Jones, 1972; Ramos, 2003). Our results show that MedGraph performs comparably to BestMatch. In addition, it outperforms the traditional TF-IDF method providing evidence that using knowledge graph-based semantic search will benefit the biomedical and life science research community when adopted as a widely used method in literature search through digital libraries.

1.3. Relevant previous work

Knowledge graphs (KG) (Paulheim, 2017) have been adapted to aid search engines and recommender systems. KGs are highly efficient in those applications due to their flexibility in modeling multi-cardinal relations at the entity level. For example, Xiong et al. (2017), the authors introduced explicit semantic ranking, harnessing KG embedding. The algorithm uses graph representation learning on the metadata of articles in the online search engine named Semantic Scholar (Fricke, 2018). They use a KG embedding model to represent queries and documents as vectors in the same vector space. This work is the closest to the work we present here. The authors provided strong evidence that using KG embedding in searching academic literature improves the relevance of the returned documents drastically due to the reliance on semantics and entity matching in the process. While in Wang et al. (2017), the authors demonstrated the usefulness of KGs and semantic modeling in search engines when retrieving web pages. They used a relation extraction algorithm to construct a KG. Though they have not used graph embedding, they devised a semantic matching approach based on support vector machines.

In Montes-y Gómez et al. (2000), the authors introduced extracting a KG from the text of two documents. They then measured the similarity between these two graphs extracted from the two articles, combining relational and conceptual similarities. In Ebeid et al. (2021), the authors showed the utility of ranking methods on embedded KGs using simple cosine distance metrics to perform tasks such as link prediction in the biomedical domain. While in Matsuo et al. (2006), the authors described a system built using keyword co-occurrence matching. They remodeled the keyword matching process as a graph and applied a graph clustering technique to match keywords and queries. In Blanco and Lioma (2012), The authors modeled the text in documents as a graph instead of a Bag of Words model (BoW). Then, they used PageRank (Page et al., 1999) to derive similarity measures between documents. At the same time, the authors (Farouk et al., 2018) argued that graph modeling could enhance search relevance results based on context rather than just string similarity. They developed a system where the input documents and indices are converted to a KG. Their findings support (Ma et al., 2016), where they drove the point that graph-based search engines are highly efficient and valuable despite their challenges. Evidence of the utility of graph-based search is strengthened in Guo et al. (2021). The authors constructed a network of the standardized MeSH headings assigned to articles in MEDLINE (Motschall and Falck-Ytter, 2005). The relationships between the MeSH headings were modeled as a graph where the edges represent different hierarchical roles in the original MeSH coding system. The graph of MeSH headings was then fed to various graph embedding algorithms. The output was a learned feature vector representing each MeSH heading for each node. The data set is helpful in downstream biomedical computational tasks.

While in Wang J. Z. et al. (2014), the authors used an efficient graph-based search engine on par with PubMed. Their approach tackled the problem of returning relevant documents from three angles. They first built a parallel document indexer. Second, they modeled each article's metadata, such as MeSH terms and keywords, as a graph and applied a personalized PageRank (Lofgren et al., 2016) to rank the concepts in the built graph, followed by TF-IDF (Pita et al., 2018) to rank the documents relative to a query. Third, they included the user's search behavior as a factor in relevance, similar to BestMatch (Fiorini et al., 2018). Despite its efficiency compared to PubMed, the algorithm requires user input and is not fully unsupervised. The BestMatch (Fiorini et al., 2018) is the newest algorithm used by the PubMed search engine to find the most relevant articles to a user's query. BestMatch relies on extracting features from articles and including prior user search logs into a relevance ranking prediction model. The model then finds the most relevant results personalized to each user. BestMatch provides excellent results compared with previous approaches in PubMed, yet it does not consider any semantics failing to distinguish ambiguity in queries.

In the next section, we describe our methodology and framework proposed in this article. In Section 3, we describe our evaluation experiments and results. Section 4 discusses the results, implications, and future work. We conclude in Section 5. A complete bibliography is available in Section 6. An additional literature review is included in the Supplementary material.

2. Method

In this section, we explain in detail the proposed KG based biomedical information retrieval framework MedGraph as shown in Figure 1. An additional illustrative example of our framework's pipeline is available in Supplementary Figure 1.

2.1. The PubMed metadata database

In Xu et al. (2020), the authors extracted a metadata database from the corpus of the PubMed articles available from 1781 until December 2020 (30 million). The extracted information includes names of biological entities such as genes, proteins, species, drugs, and diseases and disambiguated author information and citation information. The primary purpose of that dataset was to create a full KG of the articles in PubMed. The extracted biomedical KG could be used in various biomedical information retrieval and data mining tasks. Here we utilize the extracted biomedical knowledge graph described in Xu et al. (2020). The dataset comes as a relational database linked by a unique identifier, each article's identifier in PubMed, also known as the PMID. Those account for $31,929,000$ articles. Author information from each article, including first names, middle names, last names, and affiliations, has been extracted and disambiguated in separate tables. In addition, the disambiguated authors have a unique identifier of AIDs.

Table 1 provides statistics and a description of the PubMed relational database for essential tables. The original dataset contains 27 tables linked by PMID. Here we extract metadata from seven tables. In addition, we do not use 31 million articles for our dataset. Instead, we choose a subset of articles that have been submitted to journals between the dates of 2/1/2019 and 2/3/2019. This subset of the articles yielded 2,696 articles when queried on PubMed. We then use the 2,696 articles to extract a first-order citation network from the table *C04_ReferenceList*. The citation network produced 100,456 articles. Finally, for the 100,456 articles, we extracted the rest of the metadata from the tables listed above, which will be described later.

2.2. Indexing

Indexing is simply mapping unique vocab to documents or the opposite like the index at the back of a book. You can expand that definition and match the extracted unique vocab to a dictionary (Xu et al., 2020). The index here is the mapping between the limited unique vocab of the recognized entities and their respective documents which is enough for our task. The difference between our indexing strategy and a more generalized approach is that we did not expand the index to include all unique entities we just limited the index to the extracted biomedical terms. In addition in our case, we use the terms extracted during the named entity recognition as a limited index. Moreover, the table named *B10_BERN_Main* represents the names of drugs, genes, diseases, and species extracted using named entity recognition using the biomedical deep learning language model BioBERT (Lee et al., 2020) in the dataset presented in Xu et al. (2020), which acts as an index in addition to being part of the KG that we will describe its extraction later in the following subsection. In addition, the index will be used to match input user queries and expansion and create query vectors. More formally, each article $p \in P$ will contain a set of biological entity mentions $m \in M$. Each mention is part of a set of mentions that distinguish each unique biological entity $b \in B$ where $M\prime \subseteq M$ and $b \rightarrow |M\prime|$. In addition, each unique biological entity has a type that can be one of four types[drug, disease, gene, species] where $b(t) \in B(T)$ *and* $T = [drug, disease, gene, species]$. Hence the relationship becomes $p[b(t)] \in P[B(T)] \; \forall \; b \rightarrow |M\prime|$. Note that we only use extracted biological entities from the text of each article to index our corpus of articles instead of using MeSH terms or UMLS (Bodenreider, 2004) vocabulary, which is considered a standard approach in work that has been done before in biomedical information retrieval and text mining.

2.3. Knowledge graph extraction

KG extraction converts the relational database of the PubMed metadata to a graph of interconnected entities, as shown in Figure 2. For each article, we first extract all author names, names of drugs, genes, proteins, diseases, and species, and related MeSH terms and Chemical Substances terms from the tables described above. Then, the unique identifiers representing each entity create the KG. As described before, KGs are represented as a list of triples. For example, in our case, when we extract an author name for an article from the metadata database, we represent that information as ["article/pmid/86509", "isWrittenBy/wrote", "author/aid/6754"]. Similarly, when we extract a drug name from an article, that information is represented as ["bioentity/drug/1256", "isMentionedIn/mentions", "article/pmid/78456"]. In addition, if an NIH grant or project funded an article, that information will be represented as [" article/pmid/5678", "isFundedBy/funds", "nih_project/project_id/4123"]. Note that the relationships are represented equally as the data in this KG model compared with a relational model.

Accordingly, each article and associated metadata will be represented as a mini KG or a concept graph, as shown in Figure 2. Those mini KGs or concept graphs could be seen as subgraphs of a larger encompassing KG. In our case, we link all the subgraphs in two ways. First, we use the citation network

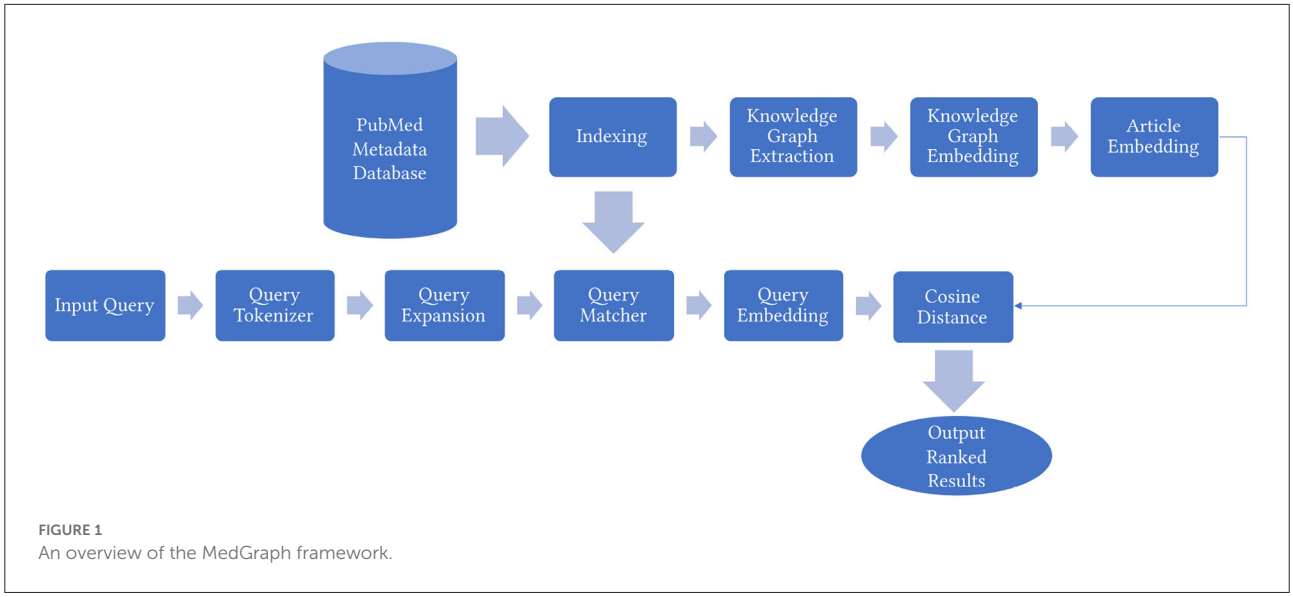


FIGURE 1
An overview of the MedGraph framework.

TABLE 1 A description of main tables in the downloaded PubMed dataset provided in Xu et al. (2020).

Table	No. of rows	No. of distinct entities	Description
A01_Articles	31,928,777	31,926,861	A table containing PubMed articles' bibliographic information.
A02_AuthorList	131,446,038	18,519,492	A table containing PubMed authors and their unique identifiers.
B10_BERN_Main	295,921,671	20,136,150	A table containing all types of extracted bio-entities by BioBERT are used in both building the Knowledge Graph and as an index.
C03_Affiliation_Merge	62,015,712	9,502,394	A table containing affiliations and extracted fine-grained items.
C05_NIH_PubMed	22,946,601	116,530	A table containing projects from NIH ExPORTER and mapping relation between PI_ID, PMID, and AND_ID.
C04_ReferenceList	633,401,975	23,856,949	A table containing reference relations between PMID and reference PMID. It was extracted from the Web of Sciences.

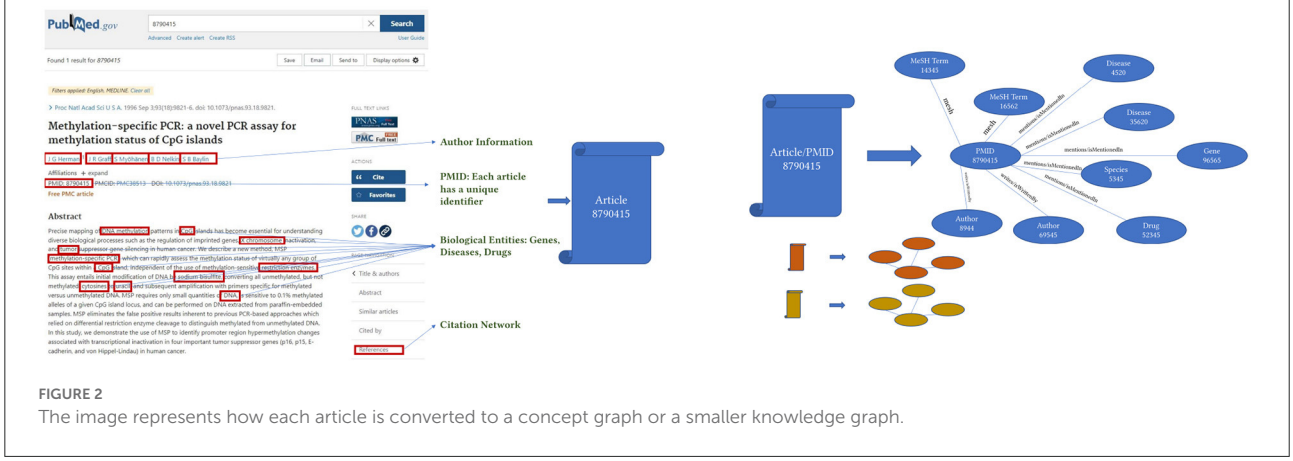


FIGURE 2
The image represents how each article is converted to a concept graph or a smaller knowledge graph.

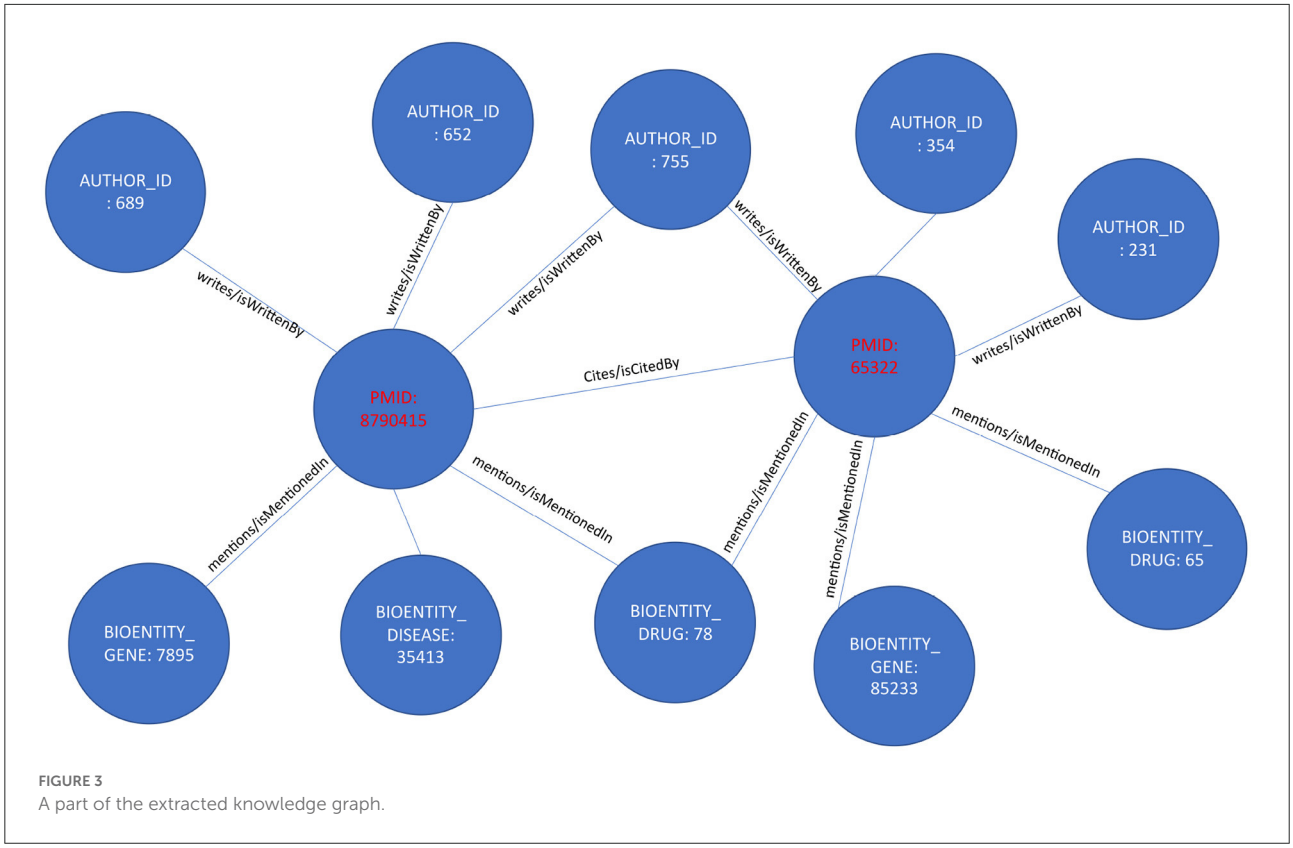


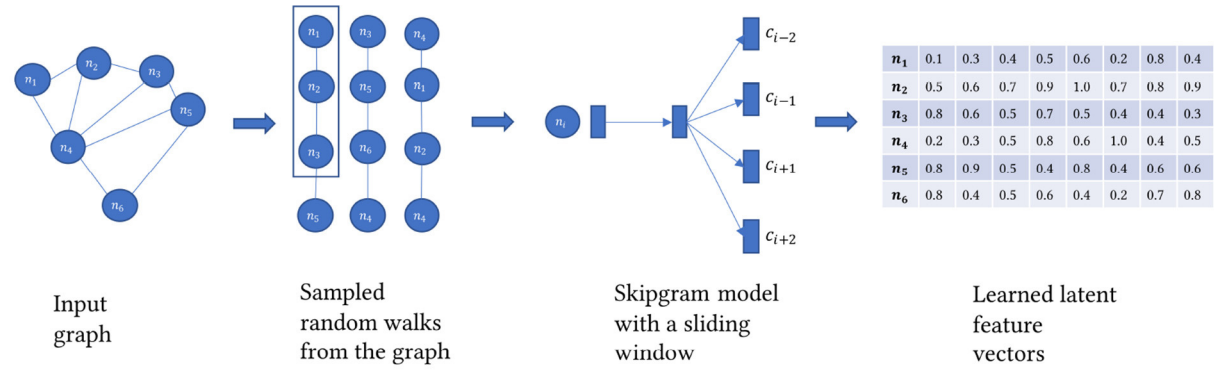
FIGURE 3
A part of the extracted knowledge graph.

provided in table *C04_ReferenceList*, representing extracted citation information from PubMed and Web of Science. The citation network provides the edges necessary to link most articles using the relationship "isCitedBy/cites." For example, two articles will be linked and represented in the knowledge graph as a triple ["article/pmid/652148," "isCitedBy/cites," "article/pmid/415923"]. Second, since the authors and the names of drugs, diseases, genes, and proteins are disambiguated and unique, if an author appears with multiple names across several articles, all the names they appeared with will have the same author identifier number. Similarly, they will have the same unique identifier if they occur with different names, such as Aspirin and NSAID for drugs, proteins, genes, and species. Moreover, we create a mini KG for each article using a unique identifier. The linked KG will also be semantically related because an author will appear in multiple articles, a drug name in various articles, and the citation network connects all articles. The final KG will be a semantically linked network representing articles, authors, NIH grants, drugs, diseases, and genes. Extracting a KG dataset as described above for the whole corpus of articles in PubMed is a daunting task. We extract only a small subset of articles with their citation information to prove the concept. KG extraction can be formalized by seeing each subject and object in the extracted triples $[v_i, r_k, v_j]$ as nodes v of type l in a KG $v(l) \in V(L)$ where each node has a type $l \in L$ where $L =$ ["article", "author", "gene/protein", "drug", "disease", "species", "nih project", "mesh term", "chemical substance"]. Edges in the KG are equivalent to verbs or predicates in the triple representation, as shown in Figure 3. Each edge $e(k) \in E(K)$ has a type $k \in K$ where $K =$ ["isCitedBy/cites", "isMentionedIn/mentions", "isFundedBy/funds", "mesh", "isRelatedTo/relates"]. Hence the triple relationship can be reformalized to $G = (V, E)$.

Regarding the validity of the extracted KG please refer to Xu et al. (2020). As mentioned before the authors extracted entities using BioBERT, a finely tuned state-of-the-art biomedical BERT model. The validation was done by comparing the results to a pretrained general BERT model on the general domain corpus. The relations were validated using multiple normalization models and dictionaries such as GNormPlus for Gene/Protein and Sieve-based entity linking for Diseases. Author disambiguation was validated using the NIH ExPORTER and NIH-funded research databases.

2.4. Knowledge graph embedding

Knowledge graph embedding models can be transductive as in learned from the structure of the graph itself (Perozzi et al., 2014; Tang et al., 2015; Grover and Leskovec, 2016). Or they



n_1	0.1	0.3	0.4	0.5	0.6	0.2	0.8	0.4
n_2	0.5	0.6	0.7	0.9	1.0	0.7	0.8	0.9
n_3	0.8	0.6	0.5	0.7	0.5	0.4	0.4	0.3
n_4	0.2	0.3	0.5	0.8	0.6	1.0	0.4	0.5
n_5	0.8	0.9	0.5	0.4	0.8	0.4	0.6	0.6
n_6	0.8	0.4	0.5	0.6	0.4	0.2	0.7	0.8

FIGURE 4
A representation of the Skip-gram model.

can be distance based by forcing a scoring function to evaluate the plausibility of the triples in the KG (Bordes et al., 2013; Lin et al., 2015). Or based on end-to-end graph-based deep learning models such as Graph Neural Networks (Kipf and Welling, 2016). More knowledge about graph embedding can be found in Wang et al. (2017). Here we aim to learn a set of feature vectors for each node or entity in the KG as shown in Figure 4. The feature vector needs to encode the structure of the graph. More formally, for the graph $G = (V, E)$ a matrix $X \in \mathbb{R}^d$ is learned *via* the function $f : v \epsilon V \rightarrow \mathbb{R}^d$. One of the constraints on the learned embedding matrix is that it can be decomposed to $X = Z_v^T Z_u$ so that X preserves the similarity between its component matrices where $v \in V$ and $u \in V$ and $Z_v \equiv X^T$ and $Z_u \equiv X$. Preserving the similarity is learned through predicting the probabilities of co-occurrence between 2 nodes in the same neighborhood within a specific context window C after sampling the graph using a random walk strategy to a size of a corpus sampled nodes, T.

$$P(v_1, v_2, v_3, \ldots, v_t) = \frac{1}{T} \sum_{T}^{t=1} \sum_{-c \le j \le c,\ j=0} \log P(v_{t+j} | v_t) \tag{1}$$

Where $c \in C$ and $t \in T$. $v_1, v_2, v_3, \ldots, v_t$ are sampled from the first order neighborhood N of a randomly chosen node v_i. To train matrix X, we approximate the probability $P(v_1, v_2, v_3, \ldots, v_t)$ over positively and negatively sampled and labeled nodes using a sliding window on the sampled chains of nodes from the graph as described in equation 1. Nodes within the context window are labeled 1, while nodes outside the context window are labeled 0. A sigmoid function is then used to normalize the parameters of the matrix X. A backpropagation phase then takes place to optimize the loss function:

$$J_t(\theta) = \log \sigma(u_0^T v_c) + \sum_{j = P(V)} \log \sigma(-u_j^T v_c) \tag{2}$$

Where u and $v \epsilon V$ and u_i and v_i are row vectors ϵX. The previously described algorithm is the Skip-gram model introduced in Mikolov et al. (2013). It is worth mentioning that first-order neighborhood means one edge at a time. It is different than the walk length. Other types of graph embedding algorithms might take into consideration 2nd and 3rd order. But in general, it is computationally impractical and intractable to take more than that. To extract KG embedding representations, we use Node2vec, the algorithm described in Grover and Leskovec (2016). Node2vec performs a modified version of the random walk strategy in Perozzi et al. (2014), including parameters p and q to control the sampling strategy. The p parameter controls the likelihood of the walk revisiting a node. The q parameter controls whether the search is constrained locally or globally. Given $q > 1$ and a random walk on an initial node, the random walk samples nodes closer to the initial node as in Breadth-First Search. Whereas, $q < 1$, random walk samples nodes further from the initial node like a Depth First Search. This customizability in search behavior allows the random walker to capture diverse structural and topological properties within the graph. The sampling strategy builds a corpus of walks starting from each node. The Skip-gram model trains on this corpus to generate a unique embedding vector for each node in the KG. Once the model finishes training, we get an embedding vector of size d for each node regardless of its type, whether an article, author, drug, disease, gene, NIH project, or MeSH term.

2.5. Article embedding

Our goal is to build a backend KG-based embedding model used by a front-end search engine to rank articles relevant to specific user queries. This step uses a pooling operation averaging all the node embedding vectors of all types of nodes connected to each article node in its first-order neighborhood.

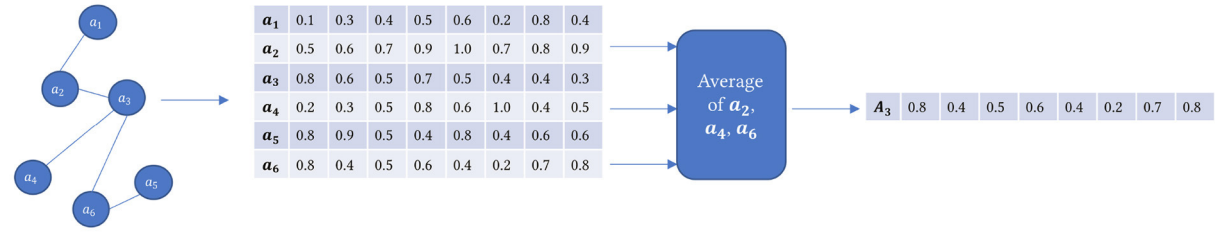


FIGURE 5
How article embeddings are generated.

We created the article embedding model in two stages. First, we performed the pooling operation of averaging all the nodes of the articles as described before mentioned in the citation network, which gives us 100,456 articles. Next, we did a second pooling operation where we averaged the first-order neighbors of articles for the 2,696 articles we intend to search.

In Figure 5, the graph on the left is our KG, where we only have article nodes along with other node types as shown in Figure 3. For example, suppose we want to calculate the embedding for article *a*3, one of the 2,696 articles, but it is also connected to other article nodes in the graph. So we average all the embedding vectors of the neighboring articles only, that is, *a*2, *a*4, and *a*6, and the resultant vector will be the one representing *a*3.

2.6. Query tokenizer

This module acts as an interface with the user. It takes user queries and parses them. The input queries are assumed to be in English and are tokenized by splitting over white spaces after removing punctuation, stop words, and verbs. For example, a query like "show me articles on depression and type 2 diabetes" after tokenization it will be reduced to ["articles," "depression," "type," "2," "diabetes"]. The output keywords will be passed to the query expansion module. Note that the assumption here is that the query should include keywords in the index.

2.7. Query matcher

The list of extracted keywords is then expanded using a sliding window of sizes 2, 3, and 4. The sliding window's function captures multiple tokens from the initial keyword list. It slides over the list of keywords and expands it. For example, our list of keywords ["articles," "depression," "type," "2," "diabetes"] will be expanded to ["articles," "depression," "type," "2," "diabetes," "articles depression," "depression type," "type 2," "2 diabetes," "articles depression type," "depression type 2," "type 2 diabetes"]. The expanded list of keywords is then matched using a Levenshtein string distance comparator to the index. The index contains all the extracted biological entities from the articles and their unique identifiers and locations. For the matched mentions in each article in the index, each biological entity's unique identifier will be extracted and passed to the next step. Similar to PubMed the system exits if the keywords are not found in the index.

2.8. Query embedding

This step aims to find all the nodes in the KG with the same identifiers as the identifiers returned by the query matcher. After identifying the nodes, their corresponding learned embedding vectors from the KG embedding step is extracted. All the vectors are averaged to a single vector in a pooling operation like Figure 5. The single vector becomes our query embedding vector.

2.9. Cosine distance and ranked results

In a Euclidean space, the cosine of angle θ between two vectors A and B is determined using the relationship:

$$similarity = \cos(\theta) = \frac{AB}{||A||\,||B||} \tag{3}$$

Since our KG has been embedded in Euclidean space, the similarity between two nodes is equivalent to the cosine of the angle between the two vectors representing the two nodes. So at this point, we have a query vector and a set of article vectors. A simple operation between the query vector and the article vectors would yield the list of articles relevant to the query vector. When sorted by the cosine score, the list of articles will be presented as ranked retrieved articles.

3. Evaluation

3.1. Dataset

For general tasks in information retrieval there exists multiple benchmark datasets (Thakur et al., 2021). However, in biomedical information retrieval, there is a lack of benchmark datasets specific to this particular task (Fiorini et al., 2018). That said it is not unusual in information retrieval for researchers to device their evaluation procedure and dataset like we did here. The only difference is that here we did not perform A/B testing with users. We used the common heuristic mean average precision (MAP) on a reproducible dataset. The contribution of this work lies in the fact that an information retrieval researcher can take this tested framework and apply it to another biomedical digital library or further test it on any digital library. Hence we extracted a proof of concept dataset from the PubMed database described in Xu et al. (2020) and available at http://er.tacc.utexas.edu/datasets/ped. The database contains 3, 190, 000 articles indexed from the year 1781 to December 2020. We extracted our target dataset of 2,696 articles submitted to journals between 02/01/2019 and 02/03/2019. We came about those dates by examining the number of articles that have been submitted to journals for each month in the past 5 years in PubMed. We then chose the month with the least number of articles submitted, February 2019. Still, the dataset at that point was too large. Note that we include extracted articles, but we also query the reference table to extract the first order citations of each article, so the number grows exponentially. Accordingly, we kept reducing the number of days where articles were submitted to their journals until we got a reasonable size dataset. The dataset was extracted by first querying the PubMed online search engine[6] for the articles that were submitted to their journals each month for each year since 2019:

((((("2019/*month*/01"[*Date* − *Completion*] : "2019/*month*/30"[*Date* − *Completion*]))))

Then the month with the least number of completed and submitted articles was chosen across all years. Then we adjusted to choose only 3 days since the size of the yielded citation network would have been beyond the scope of this study. We then settled for the dates mentioned above and queried PubMed with the query, which yielded 2,696 articles:

((((("2019/02/01"[*Date*−*Completion*] : "2019/02/03"[*Date*−*Completion*]))))

We then extracted the PMIDs of those articles. The extracted PMIDs were used to query the downloaded PubMed database to extract all the necessary metadata for each article. We first extracted the citation network of the 2,696 articles, which yielded 100,456 articles, including the 2,696 articles. For the 100,456 articles, all the metadata has been extracted, including author information, MeSH terms, Substances, NIH project involvement, extracted drug, disease, and protein names, and citation network from Table 1. The extracted metadata was used to create the KG as described in Figure 2. The final KG is a multi-undirected graph with the following description in Table 2. The total nodes in the graph were 578, 453, representing nine types of entities; authors, articles, NIH projects, MeSH terms, registered chemical substances, diseases, drugs, genes, and species. Most of those nodes were author nodes, followed by article nodes, then several NIH projects, MeSH term nodes, and extracted biological entities. Note that what defines a node in a graph is its identifier. Each node in the KG is identified by its original identifier concatenated to its type with a slash. For authors, identifiers are Author IDs (AIDs) in the database, PMIDs identify articles, Project IDs identify NIH projects, Header IDs identify MeSH terms, and extracted biological entities are identified by their unique Entity ID assigned by BioBERT in the original paper (Xu et al., 2020). For example, an article node will appear in the KG "*article*/*pmid*/652148." On the other hand, edges in the KG are identified by their edge type. Here we identify nine relationships represented with edge labels, as shown in Table 2.

TABLE 2 The description of node and edge types in the extracted knowledge graph.

Node/Edge type	Count
No. of nodes	578,453
No. of author	393,864
No. of article	100,456
No. of NIH projects	27,109
No. of MeSH terms	20,015
No. of chemical substances	9,686
No. of disease	9,594
No. of drug	8,762
No. of gene	6,094
No. of species	2,873
No. of edges	2,226,999
No. of article-relatedTo-MeSHTerm	1,049,789
No. of article-writtenBy-author	596,340
No. of article-mentions-disease	176,516
No. of article-mentions-drug	108,435
No. of article-cites-article	104,138
No. of article-mentions-species	70,694
No. of article-mentions-gene	56,337
No. of article-isFundedBy-NIHProject	54,751
No. of article-relatedTo-substances	9,999

3.2. Experimental setup

We then trained the resultant KG to extract node embedding vectors using a Node2vec (Grover and Leskovec, 2016) approach

6 https://www.ncbi.nlm.nih.gov/pmc/

implemented using Python 3.8 and the library Stellargraph (Data61, 2018). The algorithm first runs a biased random walk sampling algorithm on the graph to sample chains of nodes using the breadth-first bias parameter $q = 0.5$ and the depth-first bias parameter $p = 2$ with a walk length of 50 and 5 walks per node. The sampled corpus of node walks is then used to train a Skip-gram model as described in Figure 4. Next, we tuned the Skipgam model over multiple iterations to yield the best MAP value. The final model was trained using the vector size 128 chosen from a list of $[12, 24, 48, 64, 128, 256]$, context window size 5 chosen from the values $[3, 5, 7, 12]$ which are mostly commonly used in the literature, and the number of negative samples was 7 from the values $[7, 10, 20]$ also from the most commonly used values in the literature. The model was trained on a Windows PC with an Intel *i7* processor and 32 GB of RAM. We also implemented and trained a TF-IDF model on our corpus of 100,456 articles and then extracted the TF-IDF vectors for the 2,696 target articles to compare against our method. With the help of the Python library Gensim (Rehurek and Sojka, 2011), we first extracted a dictionary of unique tokens in the corpus and then trained a Bag of Words model. The Bag of Words model was then used to train the TF-IDF model, yielding a vectorized document matrix and unique vocabulary. We evaluated MedGraph to assess the quality of our KG embedding based on relevance ranking against PubMed's BestMatch algorithm as ground truth. We extracted a set of 15 queries from PubMed, and we applied the search to the articles that were completed between the dates of 2/1/2019 and 2/3/2019. The 15 queries were chosen randomly from the extracted index of biological entities as described in Section 2.2. They contained the names of diseases and drugs, as shown in Table 3. For example, for the query "type 2 diabetes," we use the following query to search PubMed and then download the resultant PMIDs of the ranked articles.

$$((((\text{"2019/02/01"}[Date-Completion] : \text{"2019/02/03"}[Date-Completion])))) \; AND \; (type2diabetes[TextWord])$$

Then for each query, we rank the articles based on the cosine distance metric by comparing the query vector to the article vectors described in Figure 1. We then prune the list of the resultant ranked retrieved articles by K. That means we choose the top K elements of the ranked retrieved articles from MedGraph. Then we compute the number of relevant articles, the number of retrieved articles, and the number of relevant articles retrieved. We then compute precision, recall, and F1-Score. Precision is the number of relevant articles retrieved over the total number of relevant articles. The recall is the number of relevant articles retrieved over the total number of retrieved articles. Moreover, the F1-Score is the harmonic mean of precision and recall.

We also compute the Mean Average Precision (MAP) across queries (Aslam and Yilmaz, 2006). MAP is a widely used metric in information retrieval to evaluate search engines. It focuses on precision since recall can be misleading in some cases. To compute MAP, we first calculate the average precision for each query. That is done by finding each retrieved article in the ground truth and for top K. Then computing precision at each article in the retrieved articles. That is followed by averaging the precision values across all retrieved articles K. Then averaging across all the queries.

TABLE 3 A description of the queries we used to evaluate the system against PubMed's BestMatch ranked results were used as a ground truth.

Query ID	Text	No. of relevant documents	No. of tokens
1	Alcohol	37	1
2	Amino acids	11	2
3	Bacterial infections	6	2
4	Basal cell carcinoma	3	3
5	Bipolar disorder	10	2
6	Cancer	320	1
7	Diabetes	59	1
8	Hepatitis c virus	3	3
9	Histamine	2	1
10	Insulin	25	1
11	Loss of muscle strength	1	4
12	Pediatric cancer	1	2
13	Trauma	22	1
14	Type 2 diabetes	22	3
15	Urinary tract infection	5	3

4. Results

Table 4 presents the results of the four metrics we described in the previous section. We ran 12 levels of K for both our method MedGraph and the standard TF-IDF (Ramos, 2003) approach for ranking relevant documents. Our results indicate that MedGraph has outperformed TF-IDF on the PubMed BestMatch dataset at various levels of K and across all queries and metrics. The only exception is that MAP at higher K levels was higher for TF-IDF. That might explain why TF-IDF returns more relevant documents but does not rank them higher, while MedGraph might retrieve less relevant documents more semantically related and ranked closely. In addition, both precision and recall for MedGraph were consistently higher. The recall increased exponentially with higher K, and precision decreased exponentially with higher K levels, as demonstrated in Figure 6.

MedGraph had higher MAP and F1-Scores across all K levels due to its higher recall and precision. The highest difference

TABLE 4 Results averaged across the 15 queries on different K levels.

Metric	Method	K = 1	K = 2	K = 5	K = 10	K = 25	K = 50	K = 75	K = 100	K = 150	K = 250	K = 500	K = 1,000
Recall	TFIDF	0.053	0.062	0.078	0.113	0.177	0.197	0.202	0.207	0.217	0.22	0.22	0.22
	MedGraph	**0.227**	**0.245**	**0.297**	**0.392**	**0.545**	**0.646**	**0.693**	**0.726**	**0.749**	**0.846**	**0.931**	**0.976**
Precision	TFIDF	0.467	0.4	0.307	0.293	0.248	0.171	0.136	0.117	0.1	0.064	0.032	0.016
	MedGraph	**0.867**	**0.667**	**0.507**	**0.453**	**0.336**	**0.248**	**0.202**	**0.172**	**0.134**	**0.103**	**0.064**	**0.034**
F1-Score	TFIDF	0.081	0.083	0.09	0.122	0.161	0.138	0.118	0.107	0.097	0.074	0.046	0.026
	MedGraph	**0.279**	**0.245**	**0.235**	**0.276**	**0.282**	**0.253**	**0.221**	**0.197**	**0.161**	**0.134**	**0.095**	**0.056**
MAP	TFIDF	0.467	0.383	0.272	0.221	0.184	0.177	0.175	0.174	0.174	0.173	0.173	0.173
	MedGraph	**0.867**	**0.55**	**0.284**	0.168	0.077	0.041	0.028	0.021	0.014	0.009	0.004	0.002

Bold values indicate the instances where MedGraph has outperformed TF-IDF on different metrics.

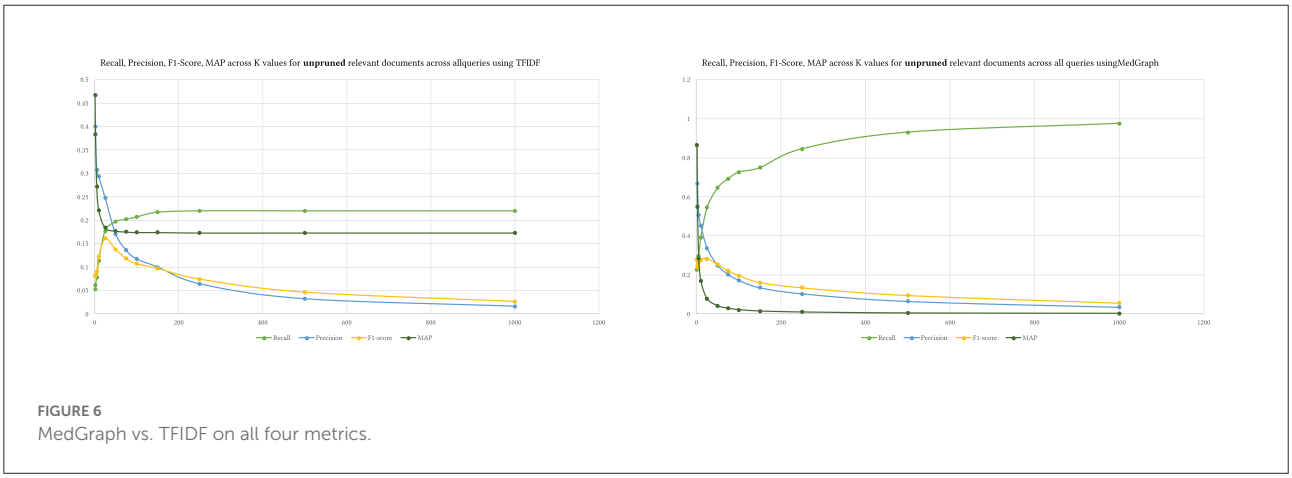


FIGURE 6
MedGraph vs. TFIDF on all four metrics.

between MedGraph and TF-IDF was at $K = 1$, indicating that the first document in the retrieved documents almost always existed in the ground truth dataset. However, recall was the lowest because most of the relevant documents did not exist in the first position in the retrieved documents. Of course, as we increase K, the recall increases, indicating that most of the relevant documents in the ground truth appeared in the retrieved documents. At $K = 10$, MedGraph started underperforming on MAP while TF-IDF stayed consistent at higher K levels. That is because MedGraph ranks a small number of the relevant documents highly, while many of the documents do not appear in MedGraph. The documents that appear in the retrieved documents are ranked closely and higher due to the semantic nature of the algorithm, while the documents that are not closely ranked and in the top are usually ranked lower and tend to be spread out.

Alternatively, in other words, MedGraph produces relevant articles that are closely ranked together due to the semantic nature of the algorithm. In contrast, TF-IDF has almost the same number of relevant articles but is not ranked closer together. Finally, we computed the four metrics by pruning the top K ground truth results from relevant documents from BestMatch.

We used the same K levels provided to prune the retrieved and relevant results. Figure 7 shows the difference between pruning the relevant ground truth articles and not pruning them. The values of recall and F1-Score do not differ between both approaches. Yet, precision and MAP are higher when the relevant documents are not pruned using K. Pruning perhaps provides a mechanism to control the ground truth dataset. We do not know how exactly BestMatch ranked it. The returned BestMatch articles from PubMed have different retrieved articles without explanation, as shown in Table 4. Hence pruning might make sense in some cases depending on the evaluation dataset.

That is also seen in Figure 8, where precision was much higher across queries with unpruned

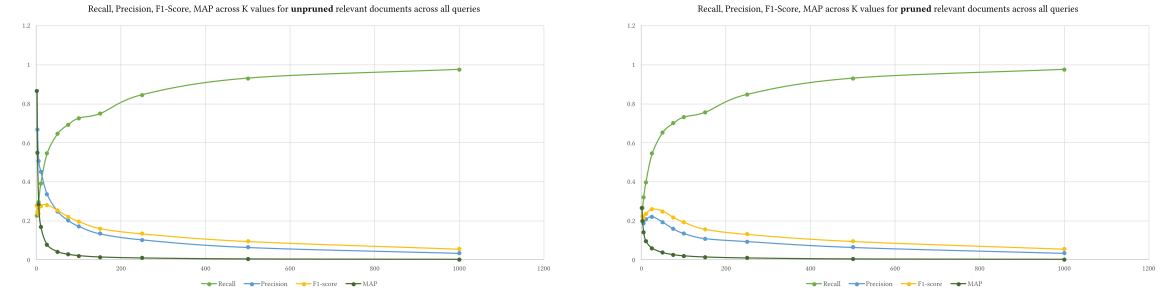


FIGURE 7
The difference between pruned and unpruned retrieved results.

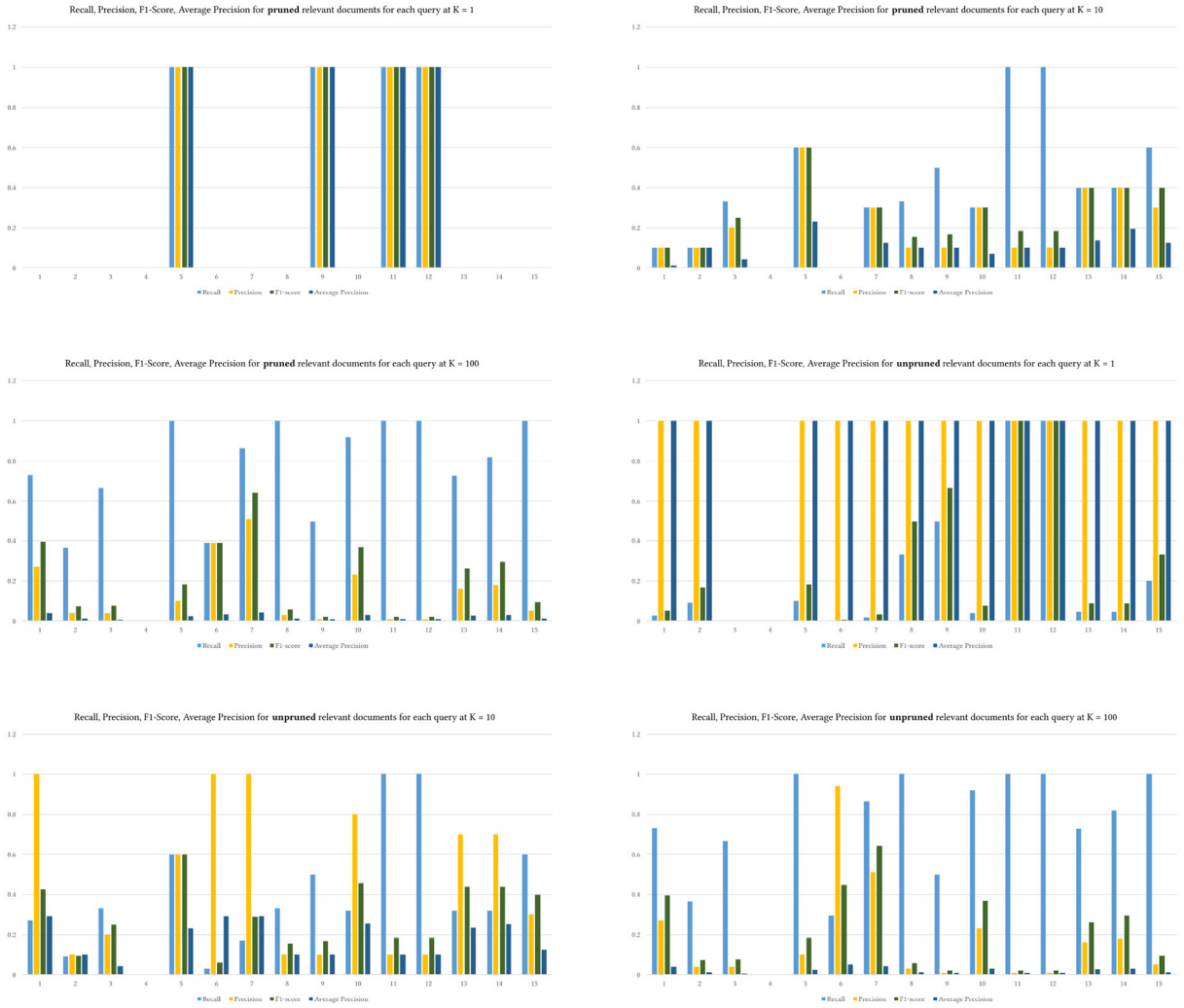


FIGURE 8
The four metrics across various levels of K over the 15 queries. Upper is pruned, and lower is unpruned relevant documents.

relevant results, indicating that MedGraph retrieved almost all of the relevant results compared with BestMatch.

5. Discussion

This work provided evidence of the utility and efficiency of KG-based methods in information retrieval, especially in the biomedical field. We highlighted the need for more techniques that rely on semantic understanding of queries and datasets to aid in automated knowledge discovery and information organization. KGs have been around for a while, yet they have not been fully utilized in search engines. Approaches such as BestMatch for PubMed are very efficient but do not understand semantics and are trained on user query logs that might change over time, requiring retraining. Traditional TF-IDF approaches do not rely on semantics and are almost outperformed by newly developed methods like ours. The results also indicated that MAP alone is not enough as an evaluation metric. The ranking is usually evaluated using A/B testing approaches involving user studies and metrics that would include users ranking relevance by hand and then computing metrics such as Normalized Discounted Cumulative Gain (Busa-Fekete et al., 2012). Precision as a metric is very informative in evaluating how many relevant articles were retrieved and, in our case, MedGraph. It highlighted its superiority. Nevertheless, metrics such as recall can be misleading. For example, if the system only retrieved one document, but that document is in the relevant documents no matter the rank, then recall shall be 100%. Precision acts as a self-assessment of the retrieved articles by MedGraph because it compares the numbers of retrieved relevant articles to the number of retrieved articles regardless of the number of relevant articles.

In our future work, we plan to conduct a user study where each user, typically a biomedical researcher or a medical student, will be invited and asked to rank documents based on specific queries. We will create our ground-truth dataset instead of relying on BestMatch as our ground truth. We also plan to expand the scope to extract a KG from the entire dataset of 30 million articles (Xu et al., 2020) and compare our model with BestMatch and TF-IDF using our ground truth. Node2vec represents a basic model incapable of encoding heterogeneity in KGs. Heterogeneity refers to a KG having more than one type of node and more than one type of edge or relationship. Hence more sophisticated embedding algorithms such as Wang Z. et al. (2014), which focuses on embedding not just the structure but also the relations in the KG could be used. In addition heterogeneous graph neural networks (Wu et al., 2020) could be also used and both might provide better results. In that light, we plan to experiment with various other KG embedding models (Wang et al., 2017) like GraphSAGE that are capable of handling dynamic KGs. In addition we will experiment with embedding models capable of capturing more semantics in the training of node embeddings, expanding our query matching capabilities to include more than four tokens, and handling out-of-context queries. Moreover, we plan to have even more metadata nodes in our KG with the potential of enriching the KG with other semantic datasets such as Chem2Bio2RDF (Chen et al., 2010). Moreover, we plan to experiment with different pooling operations in both article and query embeddings and present a full parameter sensitivity and ablation studies.

Its worth mentioning that to experiment on a huge KG of billions of nodes, we need a parallel large-scale heterogeneous embedding algorithm that could take in billions of nodes that would presumably be extracted from the whole PubMed corpus. Those models though exist and some of them are used in the industry they can be impractical in research. Most graph embedding algorithms work on a very limited amount of data. Our sample corpus here provides some evidence that this framework is effective and provides better search results than traditional methods opening the door to building a full-scale system. Finally even though this framework here does not address query intention particularly. Yet it considers semantics and relations between terms in the ranking. Semantics could be seen as a step toward future systems that consider query intentions.

6. Conclusion

In this article, we presented a proof-of-concept method to build a semantic search engine for the biomedical literature indexed in PubMed named MedGraph. We showed that our method is superior to more traditional approaches in relevance ranking and provided evidence that semantic methods in information retrieval are more needed. Furthermore, we performed a complete evaluation using various metrics on our approach using PubMed's BestMatch as a ground truth. We also presented an innovative way of converting relational databases to KGs. In the future, we hope to expand this work and provide a fully working model and system accessible by researchers to provide better ways to discover knowledge and advance science.

Data availability statement

Publicly available datasets were analyzed in this study. This data can be found at: The University of Texas at Austin, Domain Informational Vocabulary Extraction (DIVE), PubMed Knowledge Graph Datasets, http://er.tacc.utexas.edu/datasets/ped.

Author contributions

IE created this framework and wrote the article under the guidance of Dr. Elizabeth Pierce at The University of Arkansas at Little Rock.

Acknowledgments

The author would like to thank Dr. Elizabeth Pierce of the Department of Information Science at The University of Arkansas at Little Rock for her guidance and mentorship on this project. The author also would like to thank Dr. Ying Ding at the School of Information at The University of Texas at Austin for introducing him to the PubMed knowledge graph and metadata database while he was under her advisement.

Conflict of interest

The author declares 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.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fdata.2022.965619/full#supplementary-material

References

Aslam, J. A., and Yilmaz, E. (2006). "Inferring document relevance *via* average precision," in *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '06* (New York, NY: Association for Computing Machinery), 601–602. doi: 10.1145/1148170.1148275

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Sci. Am.* 284, 34–43. doi: 10.1038/scientificamerican0501-34

Blanco, R., and Lioma, C. (2012). Graph-based term weighting for information retrieval. *Inf. Retr.* 15, 54–92. doi: 10.1007/s10791-011-9172-x

Bodenreider, O. (2004). The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucl. Acids Res.* 32, D267–D270. doi: 10.1093/nar/gkh061

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. (2013). "Translating embeddings for modeling multi-relational data," in *Proceedings of the 26th International Conference on Neural Information Processing System (NIPS)*, eds J. Weston and O. Yakhnenko (New York, NY), 1–9.

Busa-Fekete, R., Szarvas, G., Élteto, T., and Kégl, B. (2012). "An apple-to-apple comparison of learning-to-rank algorithms in terms of normalized discounted cumulative gain," in *ECAI 2012-20th European Conference on Artificial Intelligence: Preference Learning: Problems and Applications in AI Workshop*, *Vol. 242* (Montpellier: IOS Press), 1–4.

Chen, B., Dong, X., Jiao, D., Wang, H., Zhu, Q., Ding, Y., et al. (2010). Chem2Bio2RDF: a semantic framework for linking and data mining chemogenomic and systems chemical biology data. *BMC Bioinformat.* 11, 255. doi: 10.1186/1471-2105-11-255

Data61, C. (2018). *StellarGraph Machine Learning Library*. GitHub Repository. Available online at: https://github.com/stellargraph/stellargraph

Ebeid, I. A., Hassan, M., Wanyan, T., Roper, J., Seal, A., and Ding, Y. (2021). "Biomedical knowledge graph refinement and completion using graph representation learning and top-k similarity measure," in *International Conference on Information* (Wuhan: Springer), 112–123. doi: 10.1007/978-3-030-71292-1_10

Farouk, M., Ishizuka, M., and Bollegala, D. (2018). "Graph matching based semantic search engine," in *Research Conference on Metadata and Semantics Research* (Madrid: Springer), 89–100. doi: 10.1007/978-3-030-14401-2_8

Fiorini, N., Canese, K., Starchenko, G., Kireev, E., Kim, W., Miller, V., et al. (2018). Best match: new relevance search for PubMed. *PLoS Biol.* 16:e2005343. doi: 10.1371/journal.pbio.2005343

Fricke, S. (2018). Semantic scholar. *J. Med. Lib. Assoc.* 106:145. doi: 10.5195/jmla.2018.280

Grover, A., and Leskovec, J. (2016). "node2vec: scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY: Association for Computing Machinery), 855–864. doi: 10.1145/2939672.2939754

Guo, Z.-H., You, Z.-H., Huang, D.-S., Yi, H.-C., Zheng, K., Chen, Z.-H., et al. (2021). Meshheading2vec: A new method for representing mesh headings as vectors based on graph embedding algorithm. *Brief. Bioinformat.* 22, 2085–2095. doi: 10.1093/bib/bbaa037

Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., Melo, G. d., Gutierrez, C., et al. (2021). "Knowledge graphs," in *Synthesis Lectures on Data, Semantics, and Knowledge 12* (Morgan & Claypool Publishers), 1–257.

Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *J. Document.*

Kipf, T. N., and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv [Preprint]*. arXiv:1609.02907.

Lassila, O., and Swick, R. R. (1998). *Resource Description Framework (RDF) Model and Syntax Specification*. World Wide and Web Consortium.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., et al. (2020). BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* 36, 1234–1240. doi: 10.1093/bioinformatics/btz682

Lin, Y., Liu, Z., Sun, M., Liu, Y., and Zhu, X. (2015). "Learning entity and relation embeddings for knowledge graph completion," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2181–2187. doi: 10.1609/aaai.v29i1.9491

Lofgren, P., Banerjee, S., and Goel, A. (2016). "Personalized pagerank estimation and search: a bidirectional approach," in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, WSDM '16* (New York, NY: Association for Computing Machinery), 163–172. doi: 10.1145/2835776.2835823

Ma, Q., Muthukrishnan, S., and Simpson, W. (2016). "App2vec: vector modeling of mobile apps and applications," in *2016 IEEE/ACM International Conference on*

Advances in Social Networks Analysis and Mining (ASONAM) (IEEE), 599–606. doi: 10.1109/ASONAM.2016.7752297

Matsuo, Y., Sakaki, T., Uchiyama, K., and Ishizuka, M. (2006). "Graph-based word clustering using a web search engine," in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing* (Sydney, NSW), 542–550. doi: 10.3115/1610075.1610150

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Montes-y Gómez, M., López-López, A., and Gelbukh, A. (2000). "Information retrieval with conceptual graph matching," in *International Conference on Database and Expert Systems Applications* (Springer), 312–321.

Motschall, E., and Falck-Ytter, Y. (2005). Searching the MEDLINE literature database through PubMed: a short guide. *Onkologie* 28, 517–522. doi: 10.1159/000087186

Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). *The PageRank Citation Ranking: Bringing Order to the Web*. Tech. rep., Stanford InfoLab.

Paulheim, H. (2017). Knowledge graph refinement: a survey of approaches and evaluation methods. *Semant. Web* 8, 489–508. doi: 10.3233/SW-160218

Perozzi, B., Al-Rfou, R., and Skiena, S. (2014). "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY), 701–710. doi: 10.1145/2623330.2623732

Pita, R., Menezes, L., and Barreto, M. E. (2018). *Applying Term Frequency-Based Indexing to Improve Scalability and Accuracy of Probabilistic Data Linkage*. Rio de Janeiro: LADaS@ VLDB. 65–72.

Ramos, J. (2003). "Using tf-idf to determine word relevance in document queries," in *Proceedings of the First Instructional Conference on Machine Learning* (Washington, DC: Citeseer), vol. 242, 29–48.

Rehurek, R., and Sojka, P. (2011). *Gensim—Python Framework for Vector Space Modelling*. NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic.

Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., and Mei, Q. (2015). "Line: large-scale information network embedding," in *Proceedings of the 24th International Conference on World Wide Web*, 1067–1077. doi: 10.1145/2736277.2741093

Thakur, N., Reimers, N., Rücklé, A., Srivastava, A., and Gurevych, I. (2021). BEIR: a heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.

Thirion, B., Robu, I., and Darmoni, S. J. (2009). "Optimization of the pubmed automatic term mapping," in *Medical Informatics in a United and Healthy Europe* (IOS Press), 238–242.

Wang, J. Z., Zhang, Y., Dong, L., Li, L., Srimani, P. K., and Yu, P. S. (2014). G-Bean: an ontology-graph based web tool for biomedical literature retrieval. *BMC Bioinformat.* 15, 1–9. doi: 10.1186/1471-2105-15-S12-S1

Wang, Q., Mao, Z., Wang, B., and Guo, L. (2017). "Knowledge graph embedding: A survey of approaches and applications," *IEEE Transactions on Knowledge and Data Engineering 29* (IEEE), 2724–2743.

Wang, Z., Zhang, J., Feng, J., and Chen, Z. (2014). "Knowledge graph embedding by translating on hyperplanes," in *Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 28*. doi: 10.1609/aaai.v28i1.8870

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Philip, S. Y. (2020). A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* 32, 4–24. doi: 10.1109/TNNLS.2020.2978386

Xiong, C., Power, R., and Callan, J. (2017). "Explicit semantic ranking for academic search *via* knowledge graph embedding," in *Proceedings of the 26th International Conference on World Wide Web*, 1271–1279. doi: 10.1145/3038912.3052558

Xu, J., Kim, S., Song, M., Jeong, M., Kim, D., Kang, J., et al. (2020). Building a PubMed knowledge graph. *Sci. Data* 7, 1–15. doi: 10.1038/s41597-020-0543-2

frontiers | Frontiers in Artificial Intelligence

TYPE Original Research
PUBLISHED 03 November 2022
DOI 10.3389/frai.2022.1028978

OPEN ACCESS

EDITED BY
Zhaohui Steve Qin,
Emory University, United States

REVIEWED BY
Lei Zhang,
Soochow University, China
Jinran Wu,
Queensland University of Technology, Australia

*CORRESPONDENCE
Hong-Wen Deng
hdeng2@tulane.edu
Chaoyang Zhang
chaoyang.zhang@usm.edu

SPECIALTY SECTION
This article was submitted to Medicine and Public Health, a section of the journal Frontiers in Artificial Intelligence

RECEIVED 26 August 2022
ACCEPTED 29 September 2022
PUBLISHED 03 November 2022

CITATION
Song M, Greenbaum J, Luttrell J, Zhou W, Wu C, Luo Z, Qiu C, Zhao LJ, Su K-J, Tian Q, Shen H, Hong H, Gong P, Shi X, Deng H-W and Zhang C (2022) An autoencoder-based deep learning method for genotype imputation.
Front. Artif. Intell. 5:1028978.
doi: 10.3389/frai.2022.1028978

An autoencoder-based deep learning method for genotype imputation

Meng Song[1], Jonathan Greenbaum[2], Joseph Luttrell IV[1], Weihua Zhou[3], Chong Wu[4], Zhe Luo[2], Chuan Qiu[2], Lan Juan Zhao[2], Kuan-Jui Su[2], Qing Tian[2], Hui Shen[2], Huixiao Hong[5], Ping Gong[6], Xinghua Shi[7], Hong-Wen Deng[2]* and Chaoyang Zhang[1]*

[1]School of Computing Sciences and Computer Engineering, University of Southern Mississippi, Hattiesburg, MS, United States, [2]Tulane Center of Biomedical Informatics and Genomics, School of Medicine, Tulane University, New Orleans, LA, United States, [3]College of Computing, Michigan Technological University, Houghton, MI, United States, [4]Department of Biostatistics, The University of Texas MD Anderson Cancer Center, Houston, TX, United States, [5]Division of Bioinformatics and Biostatistics, National Center for Toxicological Research, US Food and Drug Administration, Jefferson, AR, United States, [6]Environmental Laboratory, U.S. Army Engineer Research and Development Center, Vicksburg, MS, United States, [7]Department of Computer & Information Sciences, Temple University, Philadelphia, PA, United States

Genotype imputation has a wide range of applications in genome-wide association study (GWAS), including increasing the statistical power of association tests, discovering trait-associated loci in meta-analyses, and prioritizing causal variants with fine-mapping. In recent years, deep learning (DL) based methods, such as sparse convolutional denoising autoencoder (SCDA), have been developed for genotype imputation. However, it remains a challenging task to optimize the learning process in DL-based methods to achieve high imputation accuracy. To address this challenge, we have developed a convolutional autoencoder (AE) model for genotype imputation and implemented a customized training loop by modifying the training process with a single batch loss rather than the average loss over batches. This modified AE imputation model was evaluated using a yeast dataset, the human leukocyte antigen (HLA) data from the 1,000 Genomes Project (1KGP), and our in-house genotype data from the Louisiana Osteoporosis Study (LOS). Our modified AE imputation model has achieved comparable or better performance than the existing SCDA model in terms of evaluation metrics such as the concordance rate (CR), the Hellinger score, the scaled Euclidean norm (SEN) score, and the imputation quality score (IQS) in all three datasets. Taking the imputation results from the HLA data as an example, the AE model achieved an average CR of 0.9468 and 0.9459, Hellinger score of 0.9765 and 0.9518, SEN score of 0.9977 and 0.9953, and IQS of 0.9515 and 0.9044 at missing ratios of 10% and 20%, respectively. As for the results of LOS data, it achieved an average CR of 0.9005, Hellinger score of 0.9384, SEN score of 0.9940, and IQS of 0.8681 at the missing ratio of 20%. In summary, our proposed method for genotype imputation has a great potential to increase the statistical power of GWAS and improve downstream post-GWAS analyses.

KEYWORDS

genotype imputation, deep learning, autoencoder, paired sample *t*-test, GWAS

Introduction

Genotype imputation has become an essential step in genome-wide association study (GWAS). It is now widely used in a variety of applications in GWAS, such as boosting the power of association studies, increasing the possibility of identifying functional single-nucleotide polymorphisms (SNPs) or causal genetic variants, enhancing the resolution in fine-mapping studies, and discovering trait-associated loci in meta-analyses (Das et al., 2018). Although the cost of whole-genome sequencing (WGS) has decreased considerably during the past few years, it remains cost-prohibitive to perform WGS for a large number of samples. Currently, most GWAS samples are genotyped with low coverage genotyping approaches such as SNP arrays (Torkamaneh and Belzile, 2021). However, these low coverage approaches will inevitably generate incomplete datasets with missing values. Missing values in genotype data can considerably limit causal variants discovery or statistical inferences in meta-analysis. Therefore, it is a necessary step to impute untyped or missing variants before performing association studies.

The first two examples of GWASs facilitated by genotype imputation were a type 2 diabetes (T2D) study in Finns (Scott et al., 2007) and a joint GWAS with 2,000 cases and 3,000 controls from the UK for seven complex diseases such as coronary artery disease (CAD) and T2D (Burton et al., 2007). From then on, genotype imputation has become an important step in GWAS for human disease studies. Another recent example is a meta-analysis with 44,506 samples to identify genomic risk loci for bone mineral density (BMD) in an osteoporosis study (Greenbaum et al., 2022). For five independent GWAS array samples in this study, missing values were imputed by using Minimac2 (Fuchsberger et al., 2015, 2) with the Trans-Omics for Precision Medicine (TOPMed) (including $>$ 97,000 high coverage genomes with a mean depth of 30$\times$) as a reference panel.

The presence of missing values in SNP genotyping arrays is a common issue and can have various causes, such as assay failures, the design of different densities for genotyping platforms, and the detection of rare variants. Current genotype imputation methods can be divided into two classes: reference-based and reference-free approaches. The reference-based genotype imputation methods need a large-scale reference panel such as TOPMed and the assumption behind them is that individuals from the same or similar ancestor can share short stretches of DNA sequence between them (Song et al., 2020). Therefore, the observed genotypes from an SNP array can be used to match DNA segments shared between a target sample with missing values and a reference panel without missing values. Reference-based imputation methods include IMPUTE5 (Rubinacci et al., 2020), BEAGLE5 (Browning et al., 2018), Minimac4 (Das et al., 2016), MACH (Li et al., 2010), and fastPHASE (Scheet and Stephens, 2006). In recent years, web-based imputation tools appeared, such as the TOPMed Imputation Server (https://imputation.biodatacatalyst.nhlbi.nih.gov/), the Michigan Imputation Server (https://imputationserver.sph.umich.edu/), and the Sanger Imputation server (https://www.sanger.ac.uk/tool/sanger-imputation-service/). However, there are some challenges for these reference-based methods such as the computational cost of genotype calling for a large number of samples in a reference panel and the restrictive nature of obtaining consent for general research use (Das et al., 2018).

In contrast, reference-free imputation methods such as mean replacement, singular value decomposition (SVD), k-nearest neighbors (KNN), and random forest (RF) do not require a reference panel. In recent years, deep learning (DL) has had a great impact on many application areas, such as natural language processing, image processing, and bioinformatics because of its ability to accommodate large datasets and model highly non-linear relationships. By combining autoencoder (AE) and convolutional neural networks (CNNs), a reference-free approach, SCDA (Sparse Convolutional Denoising Autoencoder), was used for genotype imputation (Chen and Shi, 2019). It utilizes the advantages of convolutional layers to extract local data correlations within nearby variants in an AE model structure. However, the SCDA model was implemented sequentially, and has some limitations such as the inability to handle shared information with another layer except for its subsequent layer as well as the inability to build a model with multiple inputs and outputs. In addition, the training process for the SCDA model is based on minimizing a default average loss over batches and researchers are not able to implement a custom training loop, which may be needed to further improve the performance. Therefore, there is a need to modify the model and its implementation to improve the performance of genotype imputation.

In this paper, we present an improved one-dimensional (1D) convolutional AE model, inspired by SCDA, to perform genotype imputation. Instead of using sequential or functional methods to define the neural network architectures, we utilized the model subclassing method to build our AE model as it can be more easily extended to other omics data (e.g., gene expression data). Compared with sequential and functional methods, our model subclassing method is fully customizable and enables researchers to have control over every detail of the deep neural network and the whole training pipeline. With these advantages of the model subclassing method, we improved the training process by implementing a customized training loop and using a single batch loss. We evaluated our modified AE model with two public genotype datasets [yeast data and the human leukocyte antigen (HLA) data in the 1,000 Genomes Project (1KGP)] and our own genotype data generated from the Louisiana Osteoporosis Study (LOS) project. Compared with the

SCDA model, our AE model achieved a comparable or better concordance rate (CR), Hellinger score, scaled Euclidean norm (SEN) score, and imputation quality score (IQS).

Materials and methods

Dataset sources and data preprocessing

We used three genotype datasets in this study, including the yeast data (Chen and Shi, 2019), HLA data (Chen and Shi, 2019), and LOS data (Greenbaum et al., 2022). We selected the first two publicly available datasets to benchmark the performance of our imputation approach. Then we applied our model to the LOS data, which was recently collected at Tulane University and aimed to investigate the molecular mechanisms of osteoporosis by integrating multi-omics data.

Yeast data

The yeast genotype data (Bloom et al., 2015; Chen and Shi, 2019) from the SCDA model has 28,220 variants from 4,390 samples. There are two strains of yeast: an isolate from a vineyard (RM) encoded with −1 and a laboratory strain (BY) encoded with 1, respectively. We replaced all RM variants of −1 with 2 to make sure that there were no negative values when calculating the categorical cross entropy (CCE) loss function for the model.

HLA data

The aim of the 1KGP was to provide researchers with a comprehensive open data source of human genetic variation by using technologies such as microarray genotyping, low coverage WGS with a mean depth of 7.4×, and deep exome sequencing with a mean depth of 65.7× (Auton et al., 2015; Zheng-Bradley and Flicek, 2017). The phase 3 of 1KGP (released in 2005) included 2,504 individuals from 26 multiple populations. Given the high quality of genotype data from 1KGP, it can serve as a reference panel for reference-based genotype imputation methods such as IMPUTE5 (Rubinacci et al., 2020) and BEAGLE5 (Browning et al., 2018). Specifically, HLA genes from the major histocompatibility complex (MHC) region at 6p21.3 are considered to contribute to a wide range of complex human diseases (Naito et al., 2021) and the genotypes in this HLA region are more diverse and heterogeneous (Chen and Shi, 2019). The HLA region from the 1KGP contains 28,583 genotypes from 2,504 individuals across five populations including Americans, Europeans, Africans, East Asians, and Southern Asians (Auton et al., 2015; Chen and Shi, 2019). After removing multi-allelic SNPs with Bcftools for the HLA data, there were 27,209 SNPs remaining across 2,504 individuals that are used in this study.

TABLE 1 Distribution of LOS samples based on gender and ethnicity.

Gender/ethnicity	African American	Caucasian	Total
Male	1,124	1,357	2,481
Female	986	1,519	2,505
Total	2,110	2,876	4,986

LOS data

LOS is an ongoing research study that has recruited >17,000 individuals since 2011 and aims to investigate the genetic risk factors of osteoporosis and other complex diseases (Greenbaum et al., 2022). Table 1 shows a summary of the gender and ethnicity for the available subjects in the LOS data until June 2022. In total, there are 4,986 unrelated subjects including 2,110 African Americans and 2,876 Caucasians randomly selected (stratified by sex and race groups) from the whole LOS cohort. WGS of the blood samples was conducted on a BGISEQ-500 sequencer (BGI Americas Corporation, Cambridge, MA, USA) with 350 bp paired-end reads at an average sequencing depth of 22× (Greenbaum et al., 2022). By using the Burrows-Wheeler Aligner software, sequence reads were aligned to the human reference genome (version GRCh38/hg38) (Li and Durbin, 2009). Single-nucleotide variants (SNVs) and small Insertion–deletion mutations (InDels) were detected with the HaplotypeCaller of the Genome Analysis Toolkit (GATK) (McKenna et al., 2010). Variant quality score recalibration (VQSR) was applied to filter out potential sequencing artifacts and obtain high confidence variant calls (McKenna et al., 2010).

The pipeline for quality control of the LOS genotype data is illustrated in Figure 1. Taking chromosome 20 as an example, it has 3,098,112 SNVs from 4,986 samples. We first performed sample filtering with PLINK (Purcell et al., 2007) to exclude samples with more than 95% of the genotype missing. We then used Bcftools to remove multiallelic variants (Danecek et al., 2021). To solve the unknown strand issue, we aligned the strands of genotype data to the latest version of 1KGP reference panel with a mean depth of 30× (GRCh38/hg38) (Aganezov et al., 2022) by using Genotype Harmonizer (GH) (Deelen et al., 2014). GH automatically aligns ambiguous A/T and G/C SNPs to the reference by using linkage disequilibrium (LD) patterns without prior knowledge of the strands. Next, we corrected the strand swaps with the fixref library of Bcftools and excluded any remaining unmatched SNPs for the reference genome with "Bcftools norm -check-ref x" (Danecek et al., 2021). Then, we used Vcftools (Danecek et al., 2011) to perform SNP filtering with the following criteria: missing ratio > 0% (removing all missing values), Hardy-Weinberg equilibrium (HWE) p-value $< 10^{-6}$, and minor allele frequency (MAF) $< 0.1\%$. After the above quality control steps, we retained 162,027 SNPs from 4,985 samples for chromosome 20 in LOS

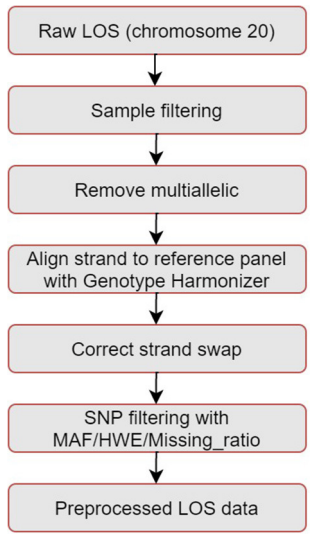


FIGURE 1
Pipeline of quality control for the LOS data. Taking chromosome 20 as an example, it has 3,098,112 SNVs from 4,986 samples. After quality control, we retained 162,027 SNPs from 4,985 samples.

data. Figure 2 visualizes a subset of the preprocessed LOS data (chromosome 20) with a heatmap at the missing ratio of 10%.

One-hot encoding of preprocessed data

Table 2 shows a summary of the three datasets after preprocessing. For the encoding of the HLA and LOS data, we first added one to all the original genotype values of 0 (0|0), 1 (0|1 or 1|0) and 2 (1|1), which represents the number of non-reference alleles. Therefore, the corresponding new genotype values are 1, 2, and 3, respectively. The purpose for doing this is to use zeros to represent fake missing values for evaluating the imputation performance (Chen and Shi, 2019). Then, we utilized one-hot encoding for these genotype values with 0 encoded as (1,0,0,0), 1 as (0,1,0,0), 2 as (0,0,1,0), and 3 as (0,0,0,1). As for the yeast data, we conducted similar processing procedures with 0 encoded as (1,0,0), 1 as (0,1,0), and 2 as (0,0,1).

In addition, to determine the impact of minor allele frequency (MAF) on the imputation accuracy for the preprocessed LOS data (chromosome 20), we divided the SNPs into four groups according to their MAFs (as shown in Table 3): MAF $>$ 5% (38,872 SNPs), 1% $<$ MAF $<$ 5% (59,579 SNPs), 0.5% $<$ MAF $<$ 1% (33,899 SNPs), and 0.1% $<$ MAF $<$ 0.5% (29,677 SNPs). The thresholds of missingness and HWE for the quality control remain the same.

AE model architecture

An AE is an unsupervised artificial neural network that learns a low-dimensional latent space representation from high-dimensional input data and then reconstructs the output data from the learned representation (Goodfellow et al., 2016). It consists of two components: an encoder and a decoder. The structure of an AE can be defined as:

$$\hat{x} = \mathcal{D}(\mathcal{E}(x)) \tag{1}$$

where x is the input, $\hat{x}$ is the output, $\mathcal{E}$ is the encoder sub-network of the AE, and $\mathcal{D}$ is the decoder sub-network of the AE. The decoder usually has an inverted symmetric structure to the encoder. The number of nodes for the stacked layers in the encoder usually decreases while the number of nodes for the decoder increases back to the number of the AE's input. The loss function for an AE can be defined as:

$$L(x, \hat{x}) \tag{2}$$

Among the different types of AE structures, a denoising AE receives corrupted data by injecting some noise into the original input and predict the uncorrupted output. If we corrupt the input genotype data with some missing values, the denoising AE is able to recover these missing values for genotype imputation. On the other hand, CNNs have been widely used for two-dimensional image classification problems. Similarly, they can be applied to 1D genotype data with one-hot encoding for human data. The equation for a 1D CNN can be described as follows (González-Muñiz et al., 2020):

$$x_l^{(m)} = \delta(b_l^{(m)} + \sum_{c=1}^{C} W_l^{(c,m)} * x_{l-1}^{(c)}) \tag{3}$$

where (*) is the convolution operator between the input $x_{l-1}^{(c)}$ and the weight of the m-th filter $W_l^{(c,m)}$ at the c-th channel, C denotes the number of channels, l represents the number of layers, $b_l^{(m)}$ is the bias for the m-th filter at layer l, δ is the activation function (such as rectified linear unit (ReLU) and sigmoid), and $x_l^{(m)}$ is the output.

We implemented a 1D convolutional AE model to perform genotype imputation (Figure 3). Taking the LOS data (chromosome 20) as an example, we selected the first 162,024 SNPs out of the total 162,027 SNPs according to their positions to ensure that the number of SNPs is divisible by 4. Here the number 4 is determined by the product of the number of convolutional layers (e.g., 2) of the encoder and the pool size (e.g., 2). The input SNPs data have been converted from two dimensions (4,985 samples, 162,024 SNPs) into three dimensions (4,985 samples, 162,024 SNPs, 4 channels) with one-hot encoding. The first two 1D convolutional layers for the SNP encoder have 32 and 64 filters, respectively. Each of them is

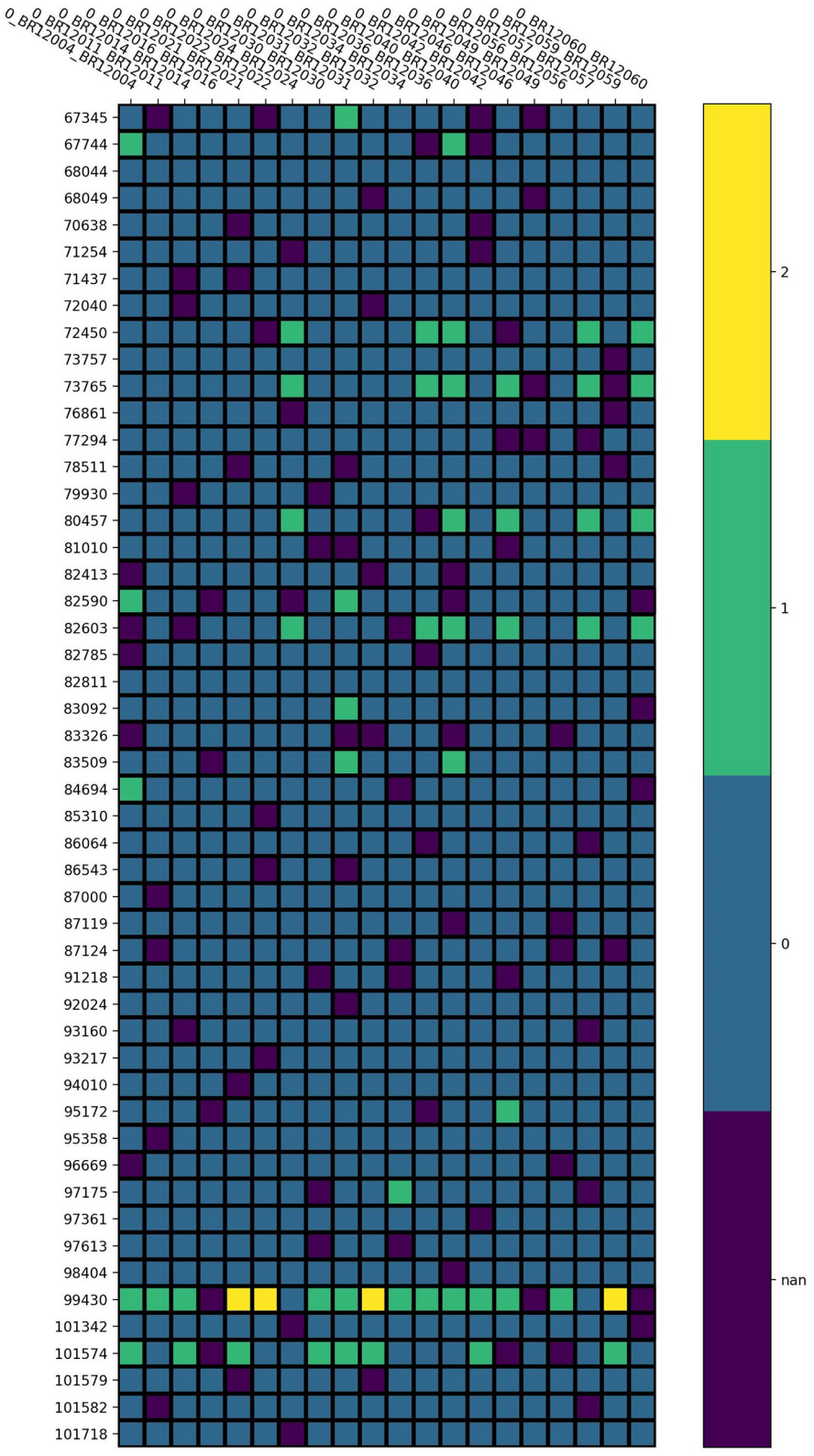


FIGURE 2
Visualization of the preprocessed LOS data (chromosome 20) with a heatmap at the missing ratio of 10%. Rows represent the position of each SNP and columns indicate different samples. Different colors represent different genotype values: purple for missing value, blue for 0, green for 1, and yellow for 2.

TABLE 2 Summary of three genotype datasets after preprocessing.

Data	Samples	Number of SNPs
Yeast	4,390	28,220
HLA	2,504	27,209
LOS (chromosome 20)	4,985	162,027

TABLE 3 SNPs of LOS (chromosome 20) data with different MAFs.

MAF	Number of SNPs
MAF > 5%	38,872
1% < MAF < 5%	59,579
0.5% < MAF < 1%	33,899
0.1% < MAF < 0.5%	29,677

followed by a max-pooling layer (with a pool size of 2) and a dropout (with a dropout rate of 0.2) layer. The embedding layer of the AE model is a 1D convolutional layer with 40,506 features and 128 filters. The SNP decoder has an inverted symmetry structure with the encoder. The first two 1D convolutional layers for the SNP decoder have 64 and 32 filters, respectively. Each of them is followed by an up-sampling layer (with a factor of 2) and a dropout (with a dropout rate of 0.2) layer. Finally, we used a 1D convolutional layer with 162,024 features, 4 channels, and the "Softmax" activation function as the output layer. All the convolutional layers of the AE model have a filter size of five and each of them has an L1 regularization factor of 0.0001.

Loss function

The loss function of the AE model can be defined as a reconstruction error between the input and imputed output such as CCE for discrete values or mean squared error (MSE) for continuous values. Since genotype data are discrete values, we utilized CCE as the loss function:

$$L_{CCE}(x, \hat{x}) = -\frac{1}{N}\sum_{i=0}^{N}\sum_{j=0}^{C} x_{ij} * \log(\hat{x}_{ij}) \tag{4}$$

where N is the total number of data points (i.e., the product of the number of samples and the number of SNPs), C is the number of channels, x_{ij} is the input SNP with one-hot encoding for the i-th sample of the j-th channel, and $\hat{x}_{ij}$ is the probability of imputed SNP. Then we defined a weighted CCE to train the AE model:

$$L\left(x, \hat{x}\right) = \alpha L_{CCE}\left(x, \hat{x}\right) \tag{5}$$

where α is the weight of CCE loss. In our AE model, we set α as 1.

SCDA model for baseline comparison

The SCDA model is based on a general denoising AE framework for genotype imputation (Chen and Shi, 2019). To capture the LD patterns among nearby genetic markers, it utilizes the CNNs in an AE structure. In total, the SCDA model has six convolutional layers with the number of filters set as 32, 64, 128, 128, 64, and 1, respectively. The size of all the filters is 5×1 and an L1 regularization ($\lambda = 0.0001$) was introduced to each convolutional layer to add a sparsity constraint for the high dimensional genotype data. Two max-pooling layers with a pool size of 2 were deployed in the encoder network to reduce the dimension of the input features, whereas two up-sampling layers with a factor of 2 were used in the decoder network to restore the dimension for the imputed output features. In addition, the SCDA model uses dropout layers (with a rate of 0.25) to prevent overfitting. For the input genotype data, it uses the one-hot encoding technique. The loss function for the model is CCE.

Model training strategy

We implemented the proposed AE model with TensorFlow v2.4.1. We utilized the model subclassing method in the Keras framework to implement our AE model as it is more flexible and can be easily extended to other omics data (e.g., gene expression). At the same time, the model subclassing method offers us the opportunity to have full control of the model. Thus, it enables us to implement a custom training loop and improve the training process by using a single batch loss rather than the average loss over batches.

We first divided the preprocessed genotype data into training, validation, and test data by randomly splitting the samples with the proportion of 64%, 16%, and 20%, respectively. Next, to compare the performances between our AE model and the SCDA model, we generated three datasets by randomly masking with enforced missing rates of 0%, 10%, and 20% after data splitting. This process replaced random values in the original genotype datasets with zeros to create missing values for each of the preprocessed genotype datasets including yeast, HLA, and LOS. A summary of the hyperparameter settings for our AE model is shown in Table 4. For example, we set the batch size as 32 and the number of epochs as 100. During the training process, we used the Adam optimizer with an initial learning rate of 0.001.

Evaluation metrics

We evaluated our AE model in terms of the evaluation metrics CR, Hellinger score, SEN score, and IQS for all the experiments as well as the Pearson correlation coefficient (PCC) in the LOS genotype imputation experiment (Stahl et al.,

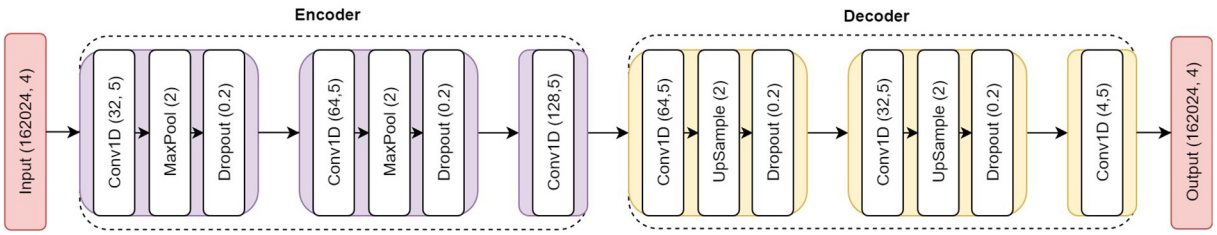


FIGURE 3
Overview of the AE structure for genotype imputation. The input is the fake missing SNPs with one-hot encoding and the output is the probability of the imputed SNPs. The six consecutive 1D convolutional layers have 32, 64, 128, 64, 32, and 4 filters, respectively. All of them have a filter size of five and L1 regularization factor of 0.0001.

TABLE 4 Summary of the hyperparameter settings for our AE model.

Hyperparameters	Values
Epochs	100
Batch size	32
Initial learning rate	0.001
Dropout	0.2
L1 regularization	0.0001
Max-pooling size	2
Up-sampling size	2
Filter size	5
Number of filters	(32, 64, 128, 64, 32, 4)
Strides	1
Padding	Same
Optimizer	Adam
Activation function	ReLU except the output layer (Softmax)

2021). The CR is the ratio of correctly imputed SNPs out of all SNPs. The Hellinger score is a measure of the distance between two probability distributions, while the SEN score is the scaled Euclidean distance between the true dosage (the expectations of the observed distribution) and the imputed dosage (the expectations of the imputed posterior distribution) (Roshyara et al., 2014). Both the Hellinger score and SEN score are calculated per SNP and per sample. The IQS is calculated based on the observed proportion of agreement and the chance agreement (Lin et al., 2010). The details for the definition of the equations for these metrics are shown in Supplementary material.

The above four evaluation metrics were based on the comparison between imputed genotypes and the ground truth of the sequenced genotypes (Stahl et al., 2021). For the calculation of the CR, we first calculated the values across SNPs for each sample, and then determined the mean value for all samples. As for the IQS, we first calculated the values across samples for each SNP, and then obtained the mean values for all SNPs. Since the Hellinger score and SEN score are calculated per SNP and per sample, we needed to accumulate them (e.g., the mean and the minimum) across samples for each SNP, and then determine the mean values for all SNPs. The minimum of the Hellinger score and the minimum of the SEN score can be viewed as the lower bound of the imputation quality. The range of all the evaluation metrics is from 0 to 1, and a score close to 1 indicates a higher imputation quality.

A paired sample *t*-test (Ross and Willson, 2017) was used to compare the evaluation metrics between our AE model and the SCDA model and to determine if there is a significant difference between them. Since we selected the same random seed for data splitting on both models and ensured the same test samples for each comparison, we chose to perform a paired *t*-test rather than a standard two sample *t*-test for comparing the mean evaluation metrics between the two models.

Experimental setup

We trained and tested the AE model and then compared it with the SCDA model on both our Seahawk server and the Tulane BIZON HPC server. Seahawk consists of four NVIDIA GP102 Titan X (Pascal) GPUs, an Intel Xeon CPU E5 1650 V4, and 98 GB system memory. The Tulane BIZON HPC server consists of two NVIDIA RTX A6000 48 GB GPUs, an AMD Ryzen Threadripper 3970X CPU, and 128 GB DDR4 system memory.

Results

To evaluate the performance of our AE model, we compared it to the SCDA method with three different genotype datasets including yeast, HLA, and LOS. We chose the CR, the Hellinger score, the SEN score, and the IQS as evaluation metrics and calculated the average value and standard deviation (SD) as well as the corresponding *p*-value by running the models three times at three different missing ratios (0%, 10%, and 20%). In addition, we visualized the results of evaluation metrics between these two

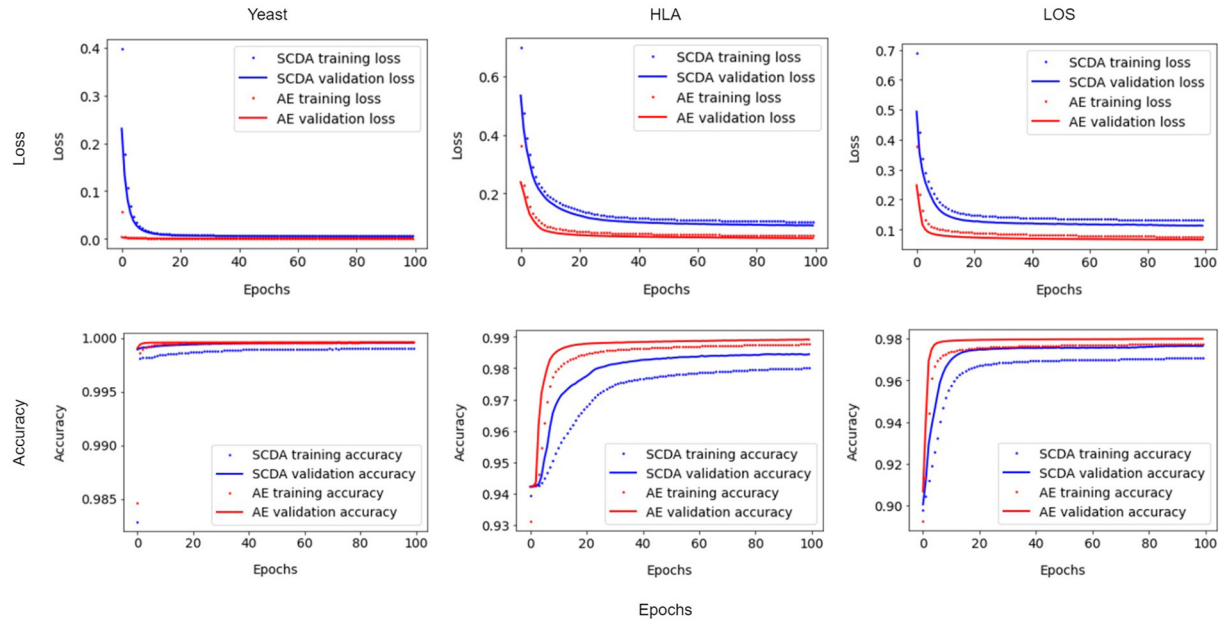


FIGURE 4
Comparison of loss **(Top)** and accuracy **(Bottom)** curve between the SCDA (blue color) and the AE (red color) model during training (dotted line) and validation (solid line) processes on the yeast, HLA, and LOS data (from left to right) at the missing ratio of 20%.

models with violin plots and histograms. Lastly, we assessed the impact of MAFs on the imputation quality with the LOS data.

Impacts of the improved training processes

We implemented a customized training loop and modified the training process by using a single batch loss rather than the running average loss over batches. Since the results of running the training process between our AE model and the SCDA model over three attempts were very similar, we chose the first instance as an example. Figure 4 shows the improvements for the loss and accuracy curve of our AE model compared to the SCDA model during training and validation processes on three different training and validation datasets, especially for the HLA and LOS genotype data, at the missing ratio of 20%. The results show that our AE model converges faster than the SCDA model and achieves comparable or higher accuracy than the SCDA model for both training and validation processes.

Imputation performance comparison

Table 5 shows the performance comparison of evaluation metrics between our AE model and the SCDA model on three test datasets at different missing ratios. We observed that our AE model achieved overall better or comparable imputation performance than the SCDA model in all metrics.

First, for the yeast data, our AE model achieved slightly better or at least comparable performance than the SCDA model in terms of the evaluation metrics CR, Hellinger score, SEN score, and IQS. Both models achieved almost the same performance regardless of the missing ratios. In contrast, for the minimum of the Hellinger score and the minimum of the SEN score, our model achieved considerably better results than the SCDA model on the data with three different missing ratios. The performance of these two metrics for both models declined with increasing missing ratios. Second, for the HLA data, our AE model had better performance than the SCDA model in all of the metrics at three different missing ratios, except one case of the minimum of the Hellinger score with the missing ratio of 10%, which shows no significant difference based on the paired sample t-test (p-value $= 0.1395$). On the other hand, even though the imputation performances for both models declined when the missing ratios increased, our AE model still outperformed the SCDA model. Finally, for the LOS data, our AE model performed better than the SCDA model in all the metrics at three different missing ratios except one case of the IQS with the missing ratio of 0% showing no significant improvement (p-value $= 0.0730$). Although the performance of both models decreased with the increase of missing ratios, our AE model yielded better performance than the SCDA model.

TABLE 5 Performance results (mean, SD, and *p*-value with respect to different evaluation metrics) between the AE and the SCDA model on three different test datasets at different missing ratios (0%, 10% and 20%).

Metrics	Data	Model	Missing ratio					
			0%		10%		20%	
			mean (SD)	*p*-value	mean (SD)	*p*-value	mean (SD)	*p*-value
CR (accuracy)	Yeast	SCDA	0.9999 (0.0)	0.0090	0.9979 (0.0)	0.0422	0.9970 (0.0)	0.4710
		AE	1.0000 (0.0)		0.9980 (0.0)		0.9979 (0.0)	
	HLA	SCDA	0.9947 (0.0008)	0.0103	0.9421 (0.0003)	0.0019	0.9416 (0.0002)	0.0014
		AE	1.0000 (0.0)		0.9468 (0.0001)		0.9459 (0.0001)	
	LOS	SCDA	0.9983 (0.0003)	0.0141	0.9005 (0.0007)	0.0393	0.8999 (0.0007)	0.0166
		AE	0.9999 (0.0)		0.9011 (0.0006)		0.9005 (0.0007)	
Hellinger score	Yeast	SCDA	0.9979 (0.0007)	0.0540	0.9974 (0.0005)	0.0312	0.9963 (0.0006)	0.0240
		AE	1.0000 (0.0)		0.9995 (0.0001)		0.9991 (0.0)	
	HLA	SCDA	0.9843 (0.0016)	0.0056	0.9506 (0.0010)	0.0004	0.9192 (0.0017)	0.0001
		AE	0.9995 (0.0)		0.9765 (0.0005)		0.9518 (0.0012)	
	LOS	SCDA	0.9880 (0.0010)	0.0037	0.9404 (0.0028)	0.0048	0.9043 (0.0007)	0.0004
		AE	0.9993 (0.0001)		0.9687 (0.0001)		0.9384 (0.0005)	
Minimum Hellinger score	Yeast	SCDA	0.7180 (0.0049)	0.0002	0.6477 (0.0056)	0.0006	0.6151 (0.0076)	0.0017
		AE	0.9765 (0.0004)		0.8389 (0.0018)		0.7551 (0.0020)	
	HLA	SCDA	0.8245 (0.0157)	0.0047	0.5596 (0.0118)	0.1395	0.5022 (0.0092)	0.0391
		AE	0.9826 (0.0003)		0.5753 (0.0042)		0.5119 (0.0072)	
	LOS	SCDA	0.7058 (0.0175)	0.0021	0.1626 (0.0015)	0.0054	0.1049 (0.0019)	0.0002
		AE	0.9471 (0.0048)		0.2079 (0.0033)		0.1413 (0.0012)	
SEN score	Yeast	SCDA	1.0000 (0.0)	0.0007	0.9999 (0.0)	0.0018	0.9999 (0.0)	0.0070
		AE	1.0000 (0.0)		1.0000 (0.0)		0.9999 (0.0)	
	HLA	SCDA	0.9981 (0.0003)	0.0130	0.9952 (0.0004)	0.0156	0.9929 (0.0002)	0.0014
		AE	1.0000 (0.0)		0.9977 (0.0)		0.9953 (0.0001)	
	LOS	SCDA	0.9995 (0.0001)	0.0227	0.9958 (0.0)	0.0003	0.9925 (0.0)	0.0014
		AE	1.0000 (0.0)		0.9970 (0.0)		0.9940 (0.0)	
Minimum SEN score	Yeast	SCDA	0.9795 (0.0003)	0.0001	0.9636 (0.0007)	0.0010	0.9526 (0.0013)	0.0038
		AE	0.9976 (0.0001)		0.9821 (0.0001)		0.9700 (0.0002)	
	HLA	SCDA	0.9702 (0.0043)	0.0106	0.8436 (0.0036)	0.0027	0.8195 (0.0048)	0.0045
		AE	0.9991 (0.0001)		0.8591 (0.0026)		0.8301 (0.0041)	
	LOS	SCDA	0.9681 (0.0062)	0.0188	0.6389 (0.0007)	0.0019	0.5751 (0.0035)	0.0015
		AE	0.9976 (0.0006)		0.6719 (0.0025)		0.6059 (0.0019)	
IQS	Yeast	SCDA	0.9998 (0.0)	0.0090	0.9993 (0.0)	0.0001	0.9990 (0.0)	0.0107
		AE	1.0000 (0.0)		0.9996 (0.0)		0.9991 (0.0)	
	HLA	SCDA	0.9604 (0.0056)	0.0099	0.9115 (0.0067)	0.0129	0.8678 (0.0057)	0.0113
		AE	0.9996 (0.0001)		0.9515 (0.0002)		0.9044 (0.0022)	
	LOS	SCDA	0.9899 (0.0040)	0.0730	0.9145 (0.0023)	0.0064	0.8470 (0.0023)	0.0079
		AE	0.9997 (0.0001)		0.9355 (0.0003)		0.8681 (0.0005)	

To test if there is a significant improvement between our AE model and the SCDA model, we calculated the mean and SD of each metric as well as the corresponding *p*-values (Table 5). We observed that most of the *p*-values were below a significance level of 0.05, which indicated a significant improvement between our AE model and the SCDA model.

We noticed that the imputation performance on yeast genotype data was much better than that on human genotype datasets including HLA and LOS data with different missing ratios, especially with high missing ratios (e.g., 20%). As discussed in the SCDA paper (Chen and Shi, 2019), the correlation patterns among nearby genetic markers in yeast genotype data are considerably stronger than those among

human genotype data, which led to a higher imputation performance with the yeast data. Compared with yeast, human genotypes are highly dispersed and heterogeneous, leading to more difficulty for the human genotype imputation than for yeast data.

Visualization of metrics with violin plots

A violin plot depicts not only the distribution of the numeric data (same as a box plot) but also its probability density. In other words, it shows summary statistics (e.g., median, interquartile range, and distribution except for outliers) and density of each variable (wider regions of a violin plot indicate values will occur more frequently, while narrower regions indicate values will occur less frequently). The results of evaluation metrics gathered three times between our AE model and the SCDA model on three different test datasets at the missing ratio of 20% are visualized in violin plots in Figure 5. We observed that our AE model had relatively higher metrics including the CR, the Hellinger score, the SEN score, and the IQS compared with the SCDA model.

Distribution of metrics with histogram

Figure 6 shows the frequency distribution of the metrics between our AE model and the SCDA model on three different test datasets at the missing ratio of 20%. We chose the histogram of the first run as an example because the results across all three runs were very similar. From this figure, we can see that for both the HLA and LOS data, our AE model achieved comparable distributions of the CR and SEN score to the SCDA model, while it had distributions closer to the right than the SCDA model (i.e., 1, indicating a higher imputation quality) for metrics including the Hellinger score and IQS. As for the yeast data, our AE model achieved comparable distributions of the CR and Hellinger score to the SCDA model, whereas it had the distributions closer to the right (i.e., 1) for metrics such as the SEN score and IQS compared with the SCDA model.

Imputation quality with different MAFs

Table 6 shows the performance comparison in terms of evaluation metrics including the CR, the PCC, the Hellinger score, the minimum of the Hellinger score, the SEN score, the minimum of the SEN score, and the IQS between the AE and SCDA models on the test dataset of LOS data with four ranges of MAFs (e.g., MAF $>$ 5%, 1% $<$ MAF $<$ 5%, 0.5% $<$ MAF $<$ 1%, and 0.1% $<$ MAF $<$ 0.5%) at a missing ratio of 20%. The AE model achieved overall better or comparable performance than the SCDA model in all four different ranges of MAFs, especially for the range of 0.1% $<$MAF $<$0.5% where our model demonstrated a considerably better IQS value (0.8767) than that of the SCDA model (0.0042). Based on the paired sample *t*-test, our model significantly outperformed the SCDA model in most scenarios.

We also observed two opposite trends of evaluation metrics for both our AE model and the SCDA model. With the increasing MAF, some metrics, including the CR, the Hellinger score, and the SEN score, declined. On the contrary, the PCC and IQS of the SCDA model increased when MAF was increased. This phenomenon is consistent with previous studies (Buckley et al., 2022; Kai-li et al., 2022). This is because the CR does not consider the correct genotype imputation with a random guess, especially for the rare variants. When MAF is increased, the probability of correct genotype imputation by chance decreased. On the other hand, the PCC is less sensitive to MAFs and the IQS adjusts for chance agreement and controls for allele frequencies. Therefore, both the PCC and MAF are more useful for the evaluation of imputation performance for rare variants. Interestingly, the PCC and IQS of our AE model have not shown a large difference for the four different ranges of MAFs, which means that our AE model is more robust to the impact of MAF in terms of the PCC and IQS metrics. As the LOS dataset is a multiethnic cohort including both Caucasian and African American samples, one of the advantages of our AE model compared with the SCDA model is that it can enhance the PCC and IQS imputation performance for rare variants, especially for African American data, which have more complicated genome structures and more rare variants.

Discussion

In summary, we implemented a 1D convolutional AE model for genotype imputation and increased the imputation performance by improving the learning process. The evaluation results on the three genotype datasets revealed that our AE model achieved better (or at least comparable) imputation performance measured with metrics including the CR, the Hellinger score, the minimum of the Hellinger score, the SEN score, the minimum of the SEN score, and the IQS when compared with the reported SCDA model.

As our AE imputation is a reference-free genotype imputation method, we did not compare our model with reference-based methods such as IMPUTE5, BEAGLE5, and Minimac4. However, we did compare it with the reference-free genotype imputation SCDA model. For the other basic reference-free methods including average, KNN, and SVD, Chen and Shi (Chen and Shi, 2019) have already made a comprehensive investigation between their proposed SCDA model and these popular imputation methods, and the comparison results showed that the SCDA model achieved better imputation accuracy than these popular methods. Therefore, we did not include them for the comparison with our AE approach.

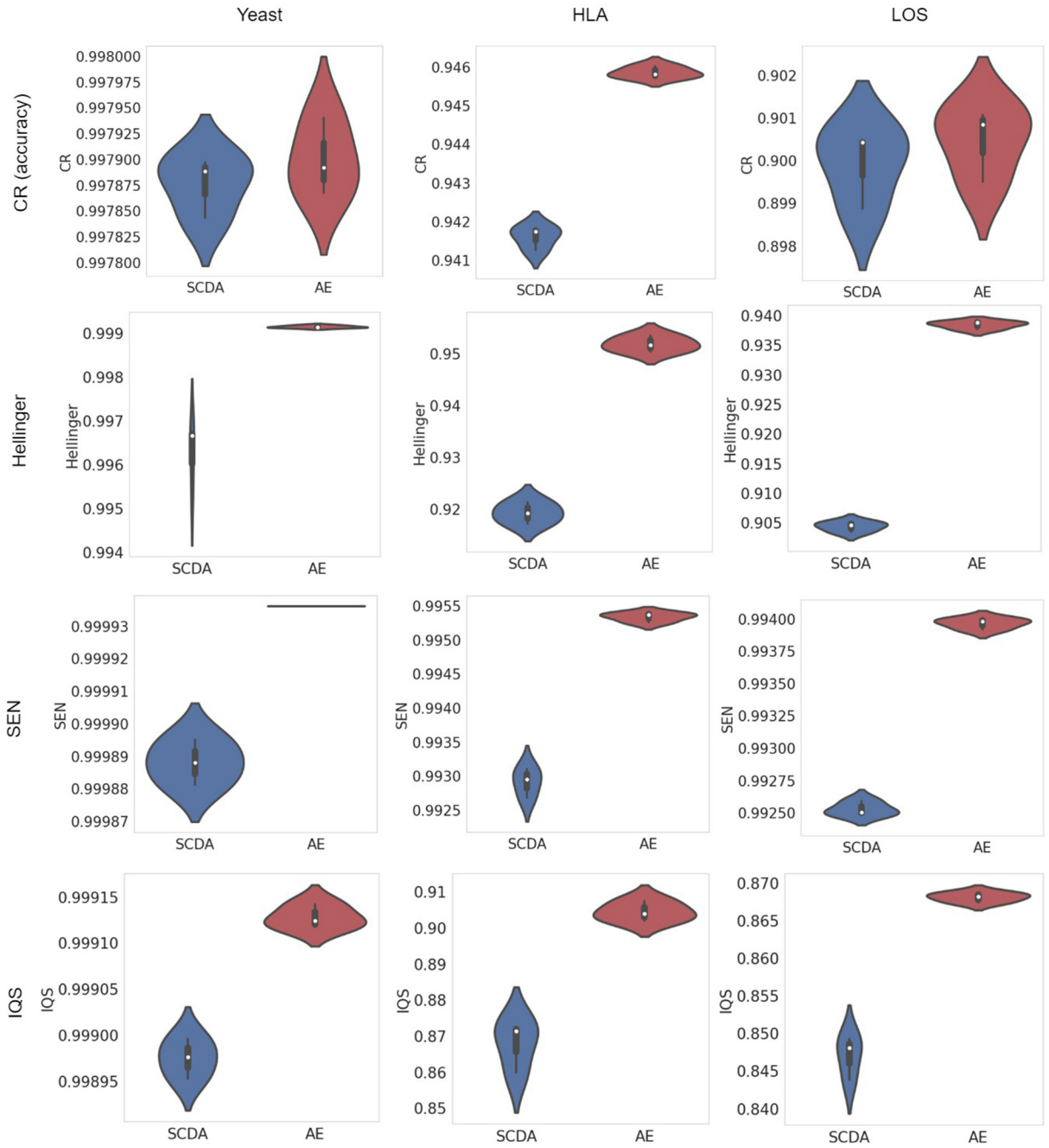


FIGURE 5
Violin plots of metrics (including the CR, the Hellinger score, the SEN score, and the IQS, from top to bottom) between the SCDA (blue color) and the AE (red color) model on the test datasets of yeast, HLA, and LOS data (from left to right) at the missing ratio of 20%.

In the comparison of imputation performance between our AE model and the SCDA model, we used the same parameters in the model structure (such as number of epochs, fake missing ratio, batch size, the learning rate, L1 regularization, dropout rate, number of filters, kernel size of the 1D convolution window, pooling size, and random seed for data splitting) except for the training strategies which were different. In other words, we implemented a customized training loop in our AE model and improved the training process by using a single batch loss rather than the average loss over batches used by the SCDA model. As shown in Figure 4, the losses decrease smoothly for all cases with insignificant fluctuations. We found that minimizing the losses corresponding to two batches separately is more effective than minimizing the average loss over two batches because the

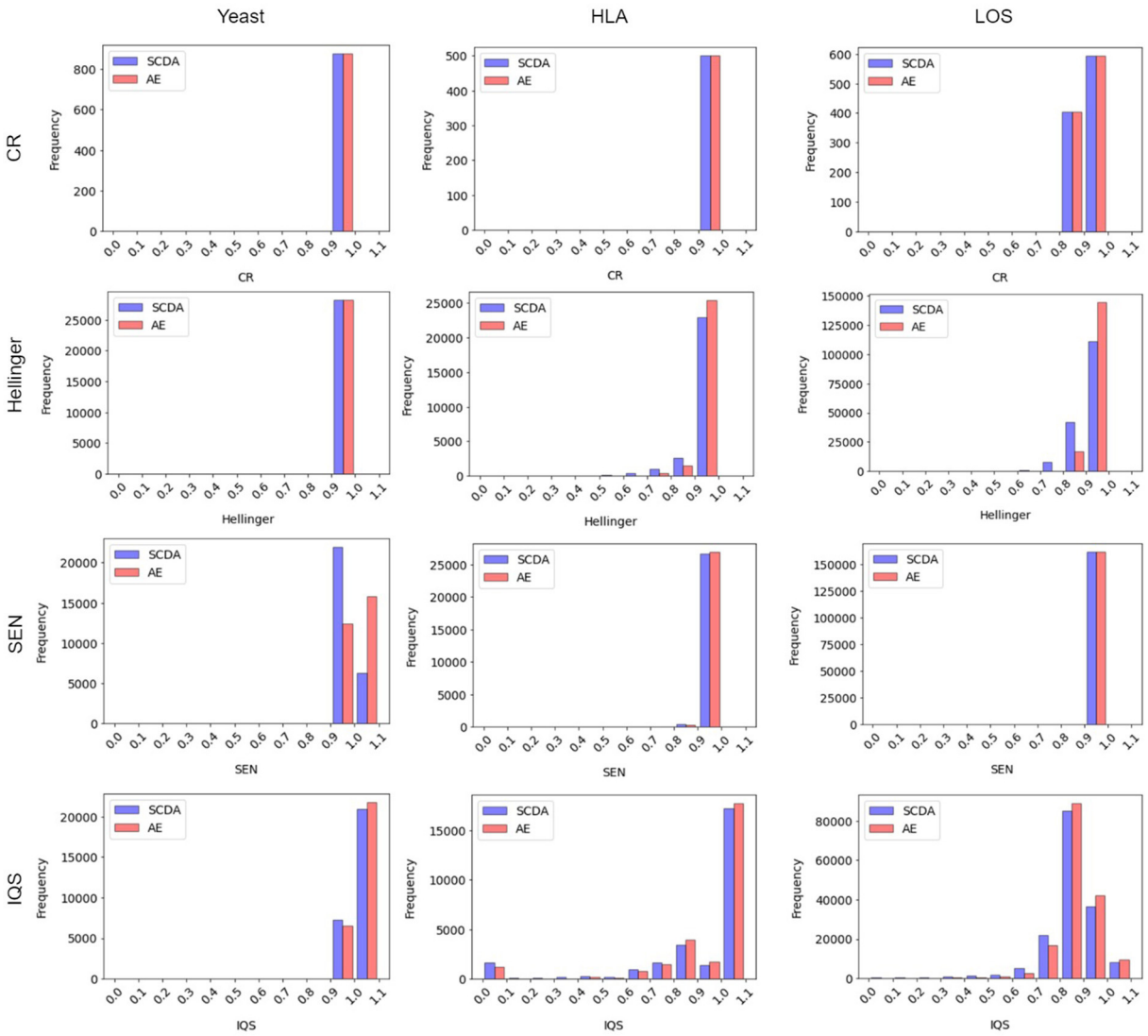


FIGURE 6
Histogram of metrics (including the CR, the Hellinger score, the SEN score, and the IQS, from top to bottom) between the SCDA (blue color) and the AE (red color) model on the test datasets of yeast, HLA, and LOS data (from left to right) at the missing ratio of 20%.

minimization of the loss for the second batch is based on the loss that has already been minimized for the first batch. Therefore, the improved imputation performance mainly resulted from the model subclassing method of the training process implemented in our proposed AE model.

There are several limitations for our AE model. First, compared with reference-based imputation methods, which can impute small sample size from a large reference panel, our AE model may not be able to handle the imputation of small sample sizes as effectively since it needs more data to train the model sufficiently. Second, since our AE model does not require a reference panel, it lacks the ability to utilize key genetic characteristics such as mutations, linkage patterns, and recombination hotspots in the reference panels of hundreds of thousands of individuals. Lastly, the imputation accuracy can be affected by several factors such as sample size, sequencing coverage, population structure, and MAF. In our current study, we only considered the effect of different MAFs on imputation accuracy and did not investigate the impacts of other factors.

For the future work, based the imputed genotype data from the AE model, we will further perform downstream analyses on LOS data for GWAS, such as identifying novel causal variants in a fine-mapping study to verify the power of genotype imputation. In addition, to overcome the drawback of lack of the interpretability of the DL methods, we will integrate prior biological information into our DL model and define biologically plausible connections within the architecture of a deep neural network. Another interesting direction of genotype

TABLE 6 Performance results (mean, SD, and *p*-value with respect to different evaluation metrics) between the AE and the SCDA model on the test dataset of LOS data with different MAFs at the missing ratio of 20%.

Metrics	Model	MAF							
		0.1% < MAF < 0.5%		0.5% < MAF < 1%		1% < MAF < 5%		MAF > 5%	
		mean (SD)	*p*-value	mean (SD)	*p*-value	mean (SD)	*p*-value	mean (SD)	*p*-value
CR (accuracy)	SCDA	0.9931(0.0002)	0.6730	0.9854(0.0001)	0.0063	0.9544(0.0011)	0.0701	0.7371(0.0011)	0.0028
	AE	0.9932(0.0001)		0.9859(0.0)		0.9560(0.0005)		0.7422(0.0009)	
PCC (dosage)	SCDA	0.5901(0.0108)	0.0006	0.7682(0.0945)	0.1907	0.8334(0.0021)	0.0005	0.8872(0.0003)	0.0003
	AE	0.8915(0.0002)		0.8980(0.0003)		0.9011(0.0002)		0.8987(0.0002)	
Hellinger score	SCDA	0.9763(0.0023)	0.0132	0.9624(0.0010)	0.0064	0.9296(0.0072)	0.0205	0.8472(0.0029)	0.0011
	AE	0.9876(0.0005)		0.9827(0.0013)		0.9646(0.0010)		0.8867(0.0024)	
Minimum Hellinger score	SCDA	0.1336(0.0067)	0.0023	0.0871(0.0032)	0.0096	0.0670(0.0070)	0.0169	0.1239(0.0015)	0.0168
	AE	0.2572(0.0047)		0.1255(0.0043)		0.1152(0.0034)		0.1420(0.002)	
SEN score	SCDA	0.9988(0.0)	0.0008	0.9987(0.0002)	0.0285	0.9961(0.0001)	0.0015	0.9845(0.0)	0.0017
	AE	0.9997(0.0)		0.9993(0.0)		0.9979(0.0)		0.9864(0.0001)	
Minimum SEN score	SCDA	0.7760(0.0023)	0.0014	0.7045(0.0104)	0.0126	0.5908(0.0199)	0.0439	0.3789(0.013)	0.1708
	AE	0.8207(0.0010)		0.7664(0.0013)		0.6614(0.0035)		0.4089(0.0070)	
IQS	SCDA	0.0042(0.0017)	0.0	0.7280(0.0614)	0.0689	0.7848(0.0105)	0.0056	0.8551(0.0007)	0.0013
	AE	0.8767(0.0008)		0.8844(0.0002)		0.8836(0.0001)		0.8699(0.0004)	

imputation is the imputation for low-coverage (e.g., 1× coverage or less) WGS data, which can be seen as an alternative approach to SNP arrays. Methods for low-coverage WGS imputation include GLIMPSE (Rubinacci et al., 2021), QUILT (Davies et al., 2021), and GeneImp (Spiliopoulou et al., 2017). All of these methods are based on large reference panels. Therefore, we will have the advantage to extend our reference-free AE model to low-coverage WGS imputation.

Conclusions

To address the problem of missing values in genotype data with deep learning methods, we implemented a convolutional AE imputation model with an improved learning strategy by using a single batch loss rather than the average loss over batches. We first evaluated our AE model with two public genotype datasets including the yeast data and the HLA data and then applied it to our own LOS data. Our modified AE imputation model outperformed the reported SCDA model in terms of the performance metrics CR, Hellinger score, SEN score, and IQS. Furthermore, our AE model significantly improved the IQS for rare variants, especially for the data from African Americans. We believe that our proposed method has a great potential to increase the statistical power of GWAS and enrich downstream GWAS analyses.

Data availability statement

The yeast (https://github.com/work-hard-playharder/SCDA/tree/master/data) and HLA (https://www.internationalgenome.org/) data can be found online. The LOS data presented in this article is not readily available due to patient confidentiality. Requests to access to it should be directed to the figshare repository (https://figshare.com/) under a DOI: https://doi.org/10.6084/m9.figshare.21441078. The code of AE model is available from: https://github.com/mengsong28/Autoencoder_imputation.

Ethics statement

The studies involving human participants were reviewed and approved by the Tulane University Institutional Review Board. The patients/participants provided their written informed consent to participate in this study.

Author contributions

H-WD, CZ, and PG conceived and supervised this project. MS and JG designed and implemented the model and drafted the manuscript. XS provided the yeast and HLA data. ZL, CQ, LZ, K-JS, and QT contributed to the LOS data curation. CZ, HH, PG, H-WD, JL, and K-JS revised the first draft manuscript. JL, WZ, CW, HH, XS, and HS provided valuable insights into the original and revised manuscript. All authors have read and agreed to the final version of the manuscript for publication.

Funding

This project was funded in part by grants from the U.S. National Institutes of Health (P20GM109036, R01AR069055,

U19AG055373, R01AG061917, AR-27065, and M01 RR00585) and a grant awarded by the U.S. Engineer Research and Development Center (W912HZ20P0023).

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.

Author disclaimer

This article reflects the views of the authors and does not necessarily reflect those of the U.S. Food and Drug Administration and U.S. Army Corps of Engineers.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frai.2022.1028978/full#supplementary-material

References

Aganezov, S., Yan, S. M., Soto, D. C., Kirsche, M., Zarate, S., Avdeyev, P., et al. (2022). A complete reference genome improves analysis of human genetic variation. *Science* 376, eabl3533. doi: 10.1126/science.abl3533

Auton, A., Abecasis, G. R., Altshuler, D. M., Durbin, R. M., Abecasis, G. R., Bentley, D. R., et al. (2015). A global reference for human genetic variation. *Nature* 526, 68–74. doi: 10.1038/nature15393

Bloom, J. S., Kotenko, I., Sadhu, M. J., Treusch, S., Albert, F. W., and Kruglyak, L. (2015). Genetic interactions contribute less than additive effects to quantitative trait variation in yeast. *Nat. Commun.* 6, 8712. doi: 10.1038/ncomms9712

Browning, B. L., Zhou, Y., and Browning, S. R. (2018). A one-penny imputed genome from next-generation reference panels. *Am. J. Hum. Genet.* 103, 338–348. doi: 10.1016/j.ajhg.2018.07.015

Buckley, R. M., Harris, A. C., Wang, G.-D., Whitaker, D. T., Zhang, Y.-P., and Ostrander, E. A. (2022). Best practices for analyzing imputed genotypes from low-pass sequencing in dogs. *Mamm. Genome* 33, 213–229. doi: 10.1007/s00335-021-09914-z

Burton, P. R., Clayton, D. G., Cardon, L. R., Craddock, N., Deloukas, P., Duncanson, A., et al. (2007). Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature* 447, 661–678. doi: 10.1038/nature05911

Chen, J., and Shi, X. (2019). Sparse convolutional denoising autoencoders for genotype imputation. *Genes* 10, 652. doi: 10.3390/genes10090652

Danecek, P., Auton, A., Abecasis, G., Albers, C. A., Banks, E., DePristo, M. A., et al. (2011). The variant call format and VCFtools. *Bioinformatics* 27, 2156–2158. doi: 10.1093/bioinformatics/btr330

Danecek, P., Bonfield, J. K., Liddle, J., Marshall, J., Ohan, V., Pollard, M. O., et al. (2021). Twelve years of SAMtools and BCFtools. *GigaScience* 10, giab008. doi: 10.1093/gigascience/giab008

Das, S., Abecasis, G. R., and Browning, B. L. (2018). Genotype imputation from large reference panels. *Annu. Rev. Genom. Hum. Genet.* 19, 73–96. doi: 10.1146/annurev-genom-083117-021602

Das, S., Forer, L., Schönherr, S., Sidore, C., Locke, A. E., Kwong, A., et al. (2016). Next-generation genotype imputation service and methods. *Nat. Genet.* 48, 1284–1287. doi: 10.1038/ng.3656

Davies, R. W., Kucka, M., Su, D., Shi, S., Flanagan, M., Cunniff, C. M., et al. (2021). Rapid genotype imputation from sequence with reference panels. *Nat. Genet.* 53, 1104–1111. doi: 10.1038/s41588-021-00877-0

Deelen, P., Bonder, M. J., van der Velde, K. J., Westra, H.-J., Winder, E., Hendriksen, D., et al. (2014). Genotype harmonizer: automatic strand alignment and format conversion for genotype data integration. *BMC Res. Notes* 7, 901. doi: 10.1186/1756-0500-7-901

Fuchsberger, C., Abecasis, G. R., and Hinds, D. A. (2015). minimac2: faster genotype imputation. *Bioinformatics* 31, 782–784. doi: 10.1093/bioinformatics/btu704

González-Muñiz, A., Díaz, I., and Cuadrado, A. A. (2020). DCNN for condition monitoring and fault detection in rotating machines and its contribution to the understanding of machine nature. *Heliyon* 6, e03395. doi: 10.1016/j.heliyon.2020.e03395

Goodfellow, I., Bengio, Y., and Courville, A. (2016). "Autoencoders," in *Deep Learning* (Cambridge, MA: MIT Press).

Greenbaum, J., Su, K.-J., Zhang, X., Liu, Y., Liu, A., Zhao, L.-J., et al. (2022). A multiethnic whole genome sequencing study to identify novel loci for bone mineral density. *Hum. Mol. Genet.* 31, 1067–1081. doi: 10.1093/hmg/ddab305

Kai-li, Z., Xia, P., Sai-xian, Z., Hui-wen, Z., Jia-hui, L., Sheng-song, X., et al. (2022). A comprehensive evaluation of factors affecting the accuracy of pig genotype imputation using a single or multi-breed reference population. *J. Integr. Agric.* 21, 486–495. doi: 10.1016/S2095-3119(21)63695-X

Li, H., and Durbin, R. (2009). Fast and accurate short read alignment with Burrows–Wheeler transform. *Bioinformatics* 25, 1754–1760. doi: 10.1093/bioinformatics/btp324

Li, Y., Willer, C. J., Ding, J., Scheet, P., and Abecasis, G. R. (2010). MaCH: using sequence and genotype data to estimate haplotypes and unobserved genotypes. *Genet. Epidemiol.* 34, 816–834. doi: 10.1002/gepi.20533

Lin, P., Hartz, S. M., Zhang, Z., Saccone, S. F., Wang, J., Tischfield, J. A., et al. (2010). A new statistic to evaluate imputation reliability. *PLOS ONE* 5, e9697. doi: 10.1371/journal.pone.0009697

McKenna, A., Hanna, M., Banks, E., Sivachenko, A., Cibulskis, K., Kernytsky, A., et al. (2010). The genome analysis toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res.* 20, 1297–1303. doi: 10.1101/gr.107524.110

Naito, T., Suzuki, K., Hirata, J., Kamatani, Y., Matsuda, K., Toda, T., et al. (2021). A deep learning method for HLA imputation and trans-ethnic MHC fine-mapping of type 1 diabetes. *Nat. Commun.* 12, 1639. doi: 10.1038/s41467-021-21975-x

Purcell, S., Neale, B., Todd-Brown, K., Thomas, L., Ferreira, M. A. R., Bender, D., et al. (2007). PLINK: a tool set for whole-genome association and population-based linkage analyses. *Am. J. Hum. Genet.* 81, 559–575. doi: 10.1086/519795

Roshyara, N. R., Kirsten, H., Horn, K., Ahnert, P., and Scholz, M. (2014). Impact of pre-imputation SNP-filtering on genotype imputation results. *BMC Genet.* 15, 88. doi: 10.1186/s12863-014-0088-5

Ross, A., and Willson, V. L. (2017). "Paired samples T-test," in *Basic and Advanced Statistical Tests: Writing Results Sections and Creating Tables and Figures*, eds. A. Ross and V. L. Willson (Rotterdam: SensePublishers), 17–19. doi: 10.1007/978-94-6351-086-8_4

Rubinacci, S., Delaneau, O., and Marchini, J. (2020). Genotype imputation using the Positional Burrows Wheeler Transform. *PLOS Genet.* 16, e1009049. doi: 10.1371/journal.pgen.1009049

Rubinacci, S., Ribeiro, D. M., Hofmeister, R. J., and Delaneau, O. (2021). Efficient phasing and imputation of low-coverage sequencing data using large reference panels. *Nat. Genet.* 53, 120–126. doi: 10.1038/s41588-020-00756-0

Scheet, P., and Stephens, M. (2006). A fast and flexible statistical model for large-scale population genotype data: applications to inferring missing genotypes and haplotypic phase. *Am. J. Hum. Genet.* 78, 629–644. doi: 10.1086/502802

Scott, L. J., Mohlke, K. L., Bonnycastle, L. L., Willer, C. J., Li, Y., Duren, W. L., et al. (2007). A genome-wide association study of type 2 diabetes in Finns detects multiple susceptibility variants. *Science* 316, 1341–1345. doi: 10.1126/science.1142382

Song, M., Greenbaum, J., Luttrell, J. I., Zhou, W., Wu, C., Shen, H., et al. (2020). A review of integrative imputation for multi-omics datasets. *Front. Genet.* 11, 570255. doi: 10.3389/fgene.2020.570255

Spiliopoulou, A., Colombo, M., Orchard, P., Agakov, F., and McKeigue, P. (2017). GeneImp: fast imputation to large reference panels using genotype likelihoods from ultralow coverage sequencing. *Genetics* 206, 91–104. doi: 10.1534/genetics.117.200063

Stahl, K., Gola, D., and König, I. R. (2021). Assessment of imputation quality: comparison of phasing and imputation algorithms in real data. *Front. Genet.* 12, 724037. doi: 10.3389/fgene.2021.724037

Torkamaneh, D., and Belzile, F. (2021). "Accurate imputation of untyped variants from deep sequencing data," in *Deep Sequencing Data Analysis Methods in Molecular Biology*, ed N. Shomron (New York, NY: Springer US), 271–281. doi: 10.1007/978-1-0716-1103-6_13

Zheng-Bradley, X., and Flicek, P. (2017). Applications of the 1000 genomes project resources. *Briefings in Functional Genomics* 16, 163–170. doi: 10.1093/bfgp/elw027

frontiers | Frontiers in Big Data

TYPE Original Research
PUBLISHED 04 November 2022
DOI 10.3389/fdata.2022.1016606

OPEN ACCESS

EDITED BY
Prashanti Manda,
University of North Carolina at Greensboro, United States

REVIEWED BY
Emre Sefer,
Özyegin University, Turkey
Zhi-Ping Liu,
Shandong University, China

*CORRESPONDENCE
Jake Y. Chen
jakechen@uab.edu

SPECIALTY SECTION
This article was submitted to Medicine and Public Health, a section of the journal Frontiers in Big Data

RECEIVED 11 August 2022
ACCEPTED 14 October 2022
PUBLISHED 04 November 2022

CITATION
Nguyen T, Yue Z, Slominski R, Welner R, Zhang J and Chen JY (2022) WINNER: A network biology tool for biomolecular characterization and prioritization.
Front. Big Data 5:1016606.
doi: 10.3389/fdata.2022.1016606

WINNER: A network biology tool for biomolecular characterization and prioritization

Thanh Nguyen[1,2], Zongliang Yue[1], Radomir Slominski[1], Robert Welner[3], Jianyi Zhang[2] and Jake Y. Chen[1]*

[1]Informatics Institute in School of Medicine, The University of Alabama at Birmingham, Birmingham, AL, United States, [2]Department of Biomedical Engineering, The University of Alabama at Birmingham, Birmingham, AL, United States, [3]Comprehensive Arthritis, Musculoskeletal, Bone and Autoimmunity Center (CAMBAC), School of Medicine, The University of Alabama at Birmingham, Birmingham, AL, United States

Background and contribution: In network biology, molecular functions can be characterized by network-based inference, or "guilt-by-associations." PageRank-like tools have been applied in the study of biomolecular interaction networks to obtain further the relative significance of all molecules in the network. However, there is a great deal of inherent noise in widely accessible data sets for gene-to-gene associations or protein-protein interactions. How to develop robust tests to expand, filter, and rank molecular entities in disease-specific networks remains an ad hoc data analysis process.

Results: We describe a new biomolecular characterization and prioritization tool called Weighted In-Network Node Expansion and Ranking (WINNER). It takes the input of any molecular interaction network data and generates an optionally expanded network with all the nodes ranked according to their relevance to one another in the network. To help users assess the robustness of results, WINNER provides two different types of statistics. The first type is a node-expansion *p*-value, which helps evaluate the statistical significance of adding "non-seed" molecules to the original biomolecular interaction network consisting of "seed" molecules and molecular interactions. The second type is a node-ranking *p*-value, which helps evaluate the relative statistical significance of the contribution of each node to the overall network architecture. We validated the robustness of WINNER in ranking top molecules by spiking noises in several network permutation experiments. We have found that node degree–preservation randomization of the gene network produced normally distributed ranking scores, which outperform those made with other gene network randomization techniques. Furthermore, we validated that a more significant proportion of the WINNER-ranked genes was associated with disease biology than existing methods such as PageRank. We demonstrated the performance of WINNER with a few case studies, including Alzheimer's disease, breast cancer, myocardial infarctions, and Triple negative breast cancer (TNBC). In all these case studies, the expanded and top-ranked genes identified by WINNER reveal disease biology more significantly than those identified by other gene prioritizing software tools, including Ingenuity Pathway Analysis (IPA) and DiAMOND.

Conclusion: WINNER ranking strongly correlates to other ranking methods when the network covers sufficient node and edge information, indicating a high network quality. WINNER users can use this new tool to robustly evaluate a list of candidate genes, proteins, or metabolites produced from high-throughput biology experiments, as long as there is available gene/protein/metabolic network information.

KEYWORDS

gene prioritization, network expansion, network statistical analysis, pathway analysis, network biology

Introduction

Gene prioritization from large-scale omics projects is a central topic in disease biology (Huang H. et al., 2009). Manual searches of the literature and publicly annotated databases (Gene Ontology et al., 2013; Kanehisa et al., 2017; Tyner et al., 2017) for genes associated with a particular disease or biological process can be biased, because they are limited to existing knowledge. Sifting hundreds and thousands of gene or genetic variations associated with genes from genomic studies can also be daunting (Moreau and Tranchevent, 2012), e.g., even for a user to search for genes associated with cardiac arrhythmia (Rajab et al., 2010) within a 2-Mb region of chromosome 17 may return 77 candidate genes. For many biologists, the lack of ranking of genes based on biological relevance of disease context is an experience analogous to the pre-Google days of Internet search of web content. With influx of data from large-scale sequencing projects (Schlotterer et al., 2014), bioinformatics users increasingly count on good gene prioritization to help them generate biological hypotheses (Chen et al., 2006a; Hale et al., 2012), find potential disease biomarkers (Saha et al., 2008; Zhang and Chen, 2010, 2013), and identify candidate drug targets (Chen et al., 2006b, 2013; Li et al., 2009; Muhammad et al., 2017). However, as datasets continue to become larger and more heterogeneous, statistical (Subramanian et al., 2005; Aerts et al., 2006; Cantor et al., 2010) and text-mining (Krallinger et al., 2008; Liu et al., 2015; ElShal et al., 2016) approaches to gene prioritization lack sufficient precision in the biological knowledge context. For example, surveys of PAGER (Yue et al., 2018) for genes associated with the response of breast cancer to doxorubicin treatment may retrieve more than 2,000 statistically significant genes with MSigDB (Liberzon et al., 2015), or 234 candidate genes with the online text-mining platform Beegle (ElShal et al., 2016). The use of statistical *p*-values to prioritize retrieved genes can mislead biology users who assume statistical significance in samples equate the gene's true biological significance against one another in the experiment (Kim and Bang, 2016).

To overcome the limitations gene prioritization in practice, bioinformatics researchers have developed gene network models with which they perform knowledge-based gene prioritization and novel candidate genes identification (Chen et al., 2006a; Cowen et al., 2017). A molecular network consists of nodes (e.g., proteins) linked by edges that represent the pairwise interactions between nodes, forming a convenient computational model that is easy to interpret and has been widely used to discover (and rediscover) disease-specific genes and potential targets for treatment (Chen et al., 2009; Wu et al., 2009; Erten et al., 2011; Gottlieb et al., 2011; Guney and Oliva, 2012; Singh-Blom et al., 2013; Smedley et al., 2014; Peters et al., 2017; do Valle et al., 2018). Network-based methods also enable researchers to integrate data from a wide variety of sources, including analyses of gene-gene similarity (Alvarez-Ponce et al., 2013), proteomic interactions (Rolland et al., 2014), and regulatory pathways (Li and Campos, 2015); however, the results of prioritization strongly depend on the input gene list (Antanaviciute et al., 2015), and the list is often derived from existing databases that may lack important genes because of statistical errors or human errors during annotation. For example, acetylcholinesterase (ACHE), which is commonly associated with β-amyloid plaques and neurofibrillary tangles in the brains of patients with Alzheimer's Disease (AD; Talesa, 2001), is not among the annotated genes for AD in the KEGG database (Kanehisa et al., 2017). Input lists may also be compromised by redundancy, which can be generated from at least two sources: (1) the inclusion of genes that were falsely identified during the statistical analysis of an experiment (Yu et al., 2017), and (2) when, in an attempt to increase comprehensiveness, the list is expanded to include the gene for a "hub" protein that interacts with dozens, or even hundreds, of other proteins [e.g., ubiquitin C binds to 4,658 other molecules (Chen et al., 2017)] and, consequently is unlikely to be specific for the phenotype of interest. Furthermore, the statistical significance of a ranking is typically calculated *via* comparison to the rankings from a randomized version of the original network, but since the randomized network is often created by adding or deleting a small number of gene-gene interactions (i.e., increasing noise), or *via* total network permutation (Xie et al., 2015; Guala and Sonnhammer, 2017), much of the topology of the original network may be lost.

Related works

According to Bromberg (2013), molecular-interaction-based disease gene prioritization started in the early 2000's by pioneering techniques such as G2D (Perez-Iratxeta et al., 2002). In principle, statistical analysis of the patients' genetic data yields 100's of disease-associated genes. These genes often belong to an interaction network (Sun and Zhao, 2010), which is also called a "disease pathway." Assume that the disease phenotypes occur due to a disturbance at any point of the pathway, then disturbing the "most influential" genes is the most likely reason leading to the disease. Then, having a good disease pathway, network ranking algorithms, especially the eigenvector-based [Random Walk (Smedley et al., 2014) and PageRank (Page et al., 1999)] and centric-based [betweenness centrality (Newman, 2005)] can be used to prioritize the genes. Also, this idea can be applied to analyze key regulators in non-disease-specific biological processes. However, the pathways are usually incompleted: new disease regulators are still not discovered or some interaction among disease-associated genes are not yet shown (Bromberg, 2013). Therefore, the ranking techniques are required to extend the interaction network beyond the known disease-associated genes. Recent gene prioritization techniques have this ability. For example, DIAMOnD (Ghiassian et al., 2015) built a large network comprising genes related to 70 diseases, clustered the large network into multiple network modules, then assigned the network module to a disease; here, in the same module, genes not related to the disease module are added (extended) into the disease-specific network-module for prioritization. Ingenuity Pathway Analysis (Kramer et al., 2014) extended the disease-specific pathway by statistically estimating the likelihood of how a new gene interacts with the known disease-related gene. In Node2Vec (Grover and Leskovec, 2016; Peng et al., 2019), a "global gene network," which includes the known disease-specific genes, their direct interacting genes, and indirect interacting ones (optionally) was constructed; then, each gene is represented by a numerical vector having a fixed-length dimension to allow computing the cosine similarity between a known disease-specific gene and another gene; so, the extension can be made by choosing the genes having high cosine similarity to any of the disease-specific ones. Or, in GenePANDA (Yin et al., 2017), given a "global gene network" (similar to Node2Vec), for a specific gene, the average distance between itself and any other gene in the "global" network was subtracted by the average distance between itself and the known disease-specific genes; then, this difference was used to rank the genes.

Besides the network-based approach, gene prioritization could be performed using text mining and similarity profiling approaches (Yin et al., 2017). In the text mining approach, it is hypothesized that important genes are more likely to be mentioned in an article than non-important ones. Therefore, text mining tools, such as aBandApart (Van Vooren et al., 2007) and Gene Prospector (Yu et al., 2008), emphasize efficient queries in MEDLINE and other large literature collections to find important disease-specific genes. However, these approaches may not find important genes when the disease is not yet well-researched or when a new disease model (i.e., a new cell line or new organoid) is built to represent the disease. On the other hand, similarity profiling defines the similarity among the genes according to the disease-related information; then, if a novel gene shares a high similarity with genes that are known to be important, the novel gene will be ranked highly. For example, Endeavor (Aerts et al., 2006) and ToppGene (Chen et al., 2009) integrated multiple disease-omic databases by a machine-learning model; the model was trained to classify between the known-important genes and non-important genes; the model will produce a ranking score reflecting how important a novel gene is, respecting the already known ones. Meanwhile, the disease-specific gene expression and correlation matrix can be clustered or latent-based represented, such as in Pinta (Nitsch et al., 2011), Maxlink (Guala et al., 2014), and Genefriends (van Dam et al., 2012), where the well-known disease-specific genes are expected to concentrate in one or a few clusters/latent modules, and the novel genes in these clusters or modules would be ranked highly.

Here, we introduce a new ranking method, Weighted In-Network Node Expansion and Ranking (WINNER), that addresses many of the current limitations of network-based gene prioritization methods. As with PageRank (Winter et al., 2012) and many other gene prioritization techniques, the ranking engine of WINNER uses random-walk principles (Zhao et al., 2015). However, WINNER was designed to address the following three specific network biology tasks: (1) perform gene prioritization in a weighted biomolecular association network, (2) identify upstream regulators and targeted genes (i.e., "upstream" ranking), or (3) identifying downstream effector molecules that are specific for a particular disease or phenotype ("downstream" ranking). WINNER can generate a ranking score for each input gene, derive optional genes that are "expanded" from the original seed gene lists, and provide two different statistic for users (1) the gene expansion p-value (p_e) for adding a gene to the network, which addresses both incomprehensiveness and redundancy; and (2) the gene ranking p-value (p_r), which represents the significance of the ranking when compared to the randomized network. Furthermore, we found that compared to total network permutation (Xie et al., 2015; Guala and Sonnhammer, 2017), preserving the modularity randomization (Cowen et al., 2017) produces a randomized network that is topologically similar to the original network and yields a more normal distribution of ranks (Espinoza, 2012). We further demonstrated the benefit of WINNER in omics study result interpretations with the following case studies: (1) ranking genes that are genetically associated with Alzheimer's disease (AD); (2) ranking breast-cancer survival-related genes (Lanczky et al., 2016); (3) ranking differentially expressed genes involved in

myocardial injury in pigs for their potential roles in myocardial regeneration (Eschenhagen et al., 2017). In all these studies, we discuss how our prioritization score and statistic associated with high-ranked genes enable biology users to derive new insights and hypotheses worth further experimental investigations.

Methods

For this work, we postulated (1) that the seeded (i.e., input) genes consist of (but are not limited to) differentially expressed genes identified in a wet-lab experiment, genes in a well-curated pathway, and phenotype-associated genes mined from the literature; and (2) that genes added to the expanded network (i.e., "expansion genes") would have significantly more interactions with seeded genes (i.e., "seeded interactions") than with non-seeded genes. WINNER begins with the set of seeded genes and a collection of gene-gene interactions, iteratively applies network ranking for gene prioritization, and expands the ranked list of genes one gene at a time (Supplementary Figure 1). Each gene-gene interaction has a confidence score (scaled between 0 and 1), which is commonly included in interactome databases (Chatr-Aryamontri et al., 2013; Szklarczyk et al., 2015); however, if a confidence score is not available, then the confidence score is set to 1 for all interactions. Network ranking is first applied to the seeded genes and the interactions among them (S_0 metric, Equation 1); then, genes adjacent to the seeded genes are filtered for significant interactions with the seeded genes (p_e) to identify candidates for the expanded network. The identified candidate is added to the ranked list, and network ranking is re-applied to initiate the next iteration of the cycle. A more detailed description of each step is provided below.

Ranking genes in the network by WINNER

Undirected networks

Given a gene-gene association network, the genes are ranked as in Supplementary Video 1. First, WINNER assigns an initial score (S_0) to the genes, according to Yue et al. (2017):

$$S_0(i) = e^{2\ln(w(i)) - ln(I(i))} \quad (1)$$

where i represents the gene index, $w(i)$ is the sum of the confidence scores (normalized to between 0 and 1) for all gene-gene interactions associated with i, and $I(i)$ is the number of gene-gene interactions associated with i. Here, larger confidence scores imply stronger associations. Second, WINNER iteratively updates the gene score by applying the Random Walk technique (Page et al., 1999):

$$S_t(i) = (1-\sigma) \times S_0(i) + \sigma \times \sum_{\forall j} \frac{c(j,i) \times S_{t-1}(j)}{w(j)} \quad (2)$$

where s is the random walk damping parameter [set to $s = 0.85$ as described (Page et al., 1999)], $c(j, i)$ represents the confidence score of the interaction between gene i and gene j, and t is the index of iteration (starting at 1); $S = 0$ for genes that are outside the network but appear in the collection of gene-gene interactions. PageRank theory (Page et al., 1999) demonstrates that S_t converges ($|S_t - S_{t-1}| \overset{\text{®}}{} 0$) if t is large enough, so the iterative cycle was continued until $|S_t - S_{t-1}| < 0.001$.

Directed networks

Directed networks, such as networks of regulatory pathways, include more annotation than undirected networks. Thus, we adapted the definitions of terms in Equations 1, 2 so that WINNER could be used to (for example) infer upstream regulatory and downstream effector genes (Kramer et al., 2014). For "upstream" ranking, i is the regulatory gene and j is the gene regulated by i; thus, $w(i)$ is the sum of the confidence scores for all gene-gene relationships that i regulates, $I(i)$ is the number of gene-gene relationships regulated by i, and $c(j, i)$ is the confidence score for the regulation of j by i. For "downstream" ranking, i is the regulated gene and j is the gene that regulates i; thus, $w(i)$ is the sum of the confidence scores for all gene-gene relationships in which i is regulated, $I(i)$ is the number of gene-gene relationships in which i is regulated, and $c(j, i)$ is the confidence score for the regulation of i by j.

Statistical significance of gene ranking

To evaluate the statistical significance (p-value) of the gene ranking, we determined how likely the converging result of S (by default, S_{200}) in Equations 1, 2 is higher than in random networks. Randomization was performed in Matlab with degree-preservation (Espinoza, 2012; Tiong and Yeang, 2019) to maintain the topological characteristics of the original gene-gene network; however, the technique only generates unweighted relationships, so weights were randomly assigned from the distribution of relationship weights in the original network. One thousand random networks were generated, and the ranking scores (S_{200}) of the genes in the random networks were normally distributed (as validated *via* the Chi-square goodness-of-fit test). Thus, the ranking p-value (p_r) for each gene i was calculated by using the normal distribution [$m(i)$, $s(i)$] parameter estimation (Bowman and Azzalini, 1997):

$$p_r(i) = \begin{cases} \int_{-\infty}^{S_{200}(i)} \frac{1}{\sigma(i)\sqrt{2\pi}} e^{-\frac{(x-\mu(i))^2}{2\sigma^2}} dx & if \ S_{200}(i) < \mu(i) \\ \int_{S_{200}(i)}^{\infty} \frac{1}{\sigma(i)\sqrt{2\pi}} e^{-\frac{(x-\mu(i))^2}{2\sigma^2}} dx & if \ S_{200}(i) > \mu(i) \end{cases} \quad (3)$$

which is equivalent to computing the two-tailed p-value for a normal distribution.

Filtering candidates for expansion

We chose two hypergeometric tests that are common practice in annotation (Huang et al., 2009). First, we tested the likelihood of the candidate expansion gene having a seeded interaction relative to its total number of interactions. Second, we tested the likelihood of the candidate expansion gene having seeded interactions relative to the seeded interactions of its most similar seeded gene, with similarity determined by node degree. Thus, we calculated two p-values for each expansion gene j from the "overrepresented" point of view (Beissbarth and Speed, 2004; terms are defined in Supplementary Figure 2):

Test 1:

$$p_{1e}(j) = \sum_{l=k(j)}^{\min(n,K)} \frac{\binom{K}{l}\binom{N-K}{n-l}}{\binom{N}{K}} \tag{4}$$

Test 2:

$$\begin{cases} p_{2e}(j) = \sum_{l=k(j)}^{\min(n,K)} \frac{\binom{K}{l}\binom{N-K}{n-l}}{\binom{N}{K}} & \text{if } N > K \\ p_{2e}(j) = 1 - \sum_{l=0}^{\min(n,K)} \frac{\binom{N}{l}\binom{K-N}{k-l}}{\binom{K}{N}} & \text{if } N < K \end{cases} \tag{5}$$

in which the double-line bracket operator represents the combination operator:

$$\binom{N}{K} = \frac{N(N-1)(N-2)\ldots(N-K+1)}{K(K-1)(K-2)\ldots 1} \tag{6}$$

Genes for which both $p_{1e}(j) < 0.05$ and $p_{2e}(j) < 0.05$ were chosen as candidates for expansion. Thus, the expansion p-value (p_e) for each gene j is defined by the equation $p_e(j) = \max[p_{1e}(j), p_{2e}(j)]$.

Selecting one candidate for expanded ranking

Since there will likely be more than one candidate expansion gene remaining after filtration, WINNER estimates which of the candidates should be added to the network by calculating an expansion score (e) from the confidence score of the interaction between the candidate gene and the ranked genes, and the ranking score (S) of the ranked genes:

$$e(i) = \sum \frac{c(i,j)\,S(j)}{W(j)} \tag{7}$$

Where i is the candidate expansion gene, j represents all seeded genes that interact with the candidate expansion gene, and $W(j)$ is the sum of the confidence scores for all interactions involving all seeded genes. Note that $W(j)$ differs from $w(j)$ in Equation 2, because $w(j)$ is restricted to interactions among ranked genes.

Informatics databases and benchmarking metrics

Correlations among WINNER, PageRank (Winter et al., 2012), dual node-edge ranking (Wang et al., 2015), eigenvector centrality, betweenness centrality, node degree, and clustering coefficient (Newman, 2008) were evaluated by computing the linear correlation coefficients and p-values with Matlab (Neupane and Kiser, 2018).

For analyses of upstream and downstream genes (directed network), genes were distributed into layers *via* the breadth-first-search approach, and groups of genes that formed a self-contained cycle were treated as a single node. Results were visualized with boxplots. In each pathway, the gene rank numbers were converted into percentile format: the first rank (number 1) was converted to 100% percentile, while the last rank was converted to 0% percentile. The percentile format allowed boxplot aggregation from multiple pathways, where the different pathways had different number of genes.

Experiments demonstrating the general topological and biological significance of the WINNER ranking were conducted with the small gene set associated with AD from KEGG release 50 (2009) (Kanehisa et al., 2010) and with undirected gene-gene interactions from HAPPI version 1.0 (Chen J. Y. et al., 2009). Rankings of upstream regulators and downstream effectors were conducted with all cancer disease pathways in KEGG release 85 (Kanehisa et al., 2017; Tessier et al., 2018) and gene-gene regulatory relationships from STRING v.10.5 (Szklarczyk et al., 2017).

The effectiveness of WINNER for identifying network-expansion genes was evaluated by using KEGG release 50 [stored in PAGER 1.0 (Yue et al., 2015)] as the input with interactions of all types (without directionality) from HAPPI v.2.0 whose confidence scores exceeded 0.75 (Chen et al., 2017), and then determining how closely the expanded network matched the updated KEGG release 85 (Kanehisa et al., 2017). An analogous experiment was conducted with Ingenuity Pathway Analysis (IPA), which (in theory) can be used for both upstream and downstream expansion and HAPPI v.2.0 (Kramer et al., 2014) for comparison. Precision, recall, and F1 scores were calculated *via* the following equations:

$$precision = \frac{|E \cap U|}{|E|} \tag{8}$$

$$recall = \frac{|E \cap U|}{|E|} \tag{9}$$

$$F1 = \frac{2 \times precision \times recall}{precision + recall} \quad (10)$$

where E is the set of expansion genes determined by Winner or IPA and U is the set of genes present in KEGG release 85 but not in KEGG release 50.

The biological relevance of our rankings was evaluated by (1) determining whether the top-ranked genes from WINNER ranking of the KEGG breast cancer pathway (Kanehisa et al., 2017; https://www.genome.jp/kegg-bin/show_pathway?hsa05224) were included among the genes correlated with survival in 3951 Breast Cancer patients (Gyorffy et al., 2010); and (2) by ranking the set of differentially expressed genes from a study of myocardial regeneration in neonatal pigs (Zhu et al., 2018) with WINNER and determining whether the top-ranked genes could contribute to cardiac repair and cardiomyocyte proliferation. For the analysis of breast-cancer survival genes, we calculated the ratio of the number of genes that were both significant (survival p-value < 0.05) in the breast cancer study (Gyorffy et al., 2010) and highly ranked by WINNER (i.e., scored above a defined threshold) to the number of highly-ranked genes.

Network randomization and testing for ranking normal distribution in random networks

In WINNER, given a network (also called the original network), we examined the following network randomization approaches to evaluate which network randomization approach was the most suitable for computing the ranking p-value for each gene:

- Total rewiring (also called total network permutation; Waksman, 1968). To implement this approach, for each interaction (edge) in the original network, we randomly changed the two genes (node) connecting through this edge. Therefore, this approach preserves the number of interactions, yet it totally changes the network and gene topology.
- Randomly drawing a new network such that each gene's degree is the same to what it is in the original network (also called preserving degree; Rao et al., 1996). A gene degree, in simple description, is the number of other genes connecting to the gene in the network.
- Randomly drawing a new network with the same modularity to the original network (also called preserving modularity). We implemented this strategy according to the network modularity definition in Newman (2006). Modularity measures likely the network can be partitioned into clusters of interacting genes.
- Randomly adding 5% new interactions into the original network. These interactions were not reported in the gene-gene interaction databases.
- Randomly removing 5% of the interactions from the original network.

For each network randomization approach, starting from the same original network, we repeated 10,000 times, yielding 10,000 different random networks. Then, applying WINNER (and other ranking algorithms) yielded 10,000 random ranking results for each gene. We tested whether these random rankings followed a normal distribution using chi-square goodness of fit test (chi2gof)[1] in Matlab. In this test, the smaller chi-square (chi2) indicates that the rankings are more naturally distributed.

Literature validation using co-citations from PubMed

Important disease-specific genes are often co-mentioned in a research article. Therefore, to demonstrate the significance of the genes related to a disease, we applied a co-citations from the NCBI e-utils application programming interface (API; Sayers, 2008) that implements semantic searches of PubMed abstracts to report biomedical literature citations (https://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?). We applied "pubmed" as input of database and the concatenated string of the candidate gene and the disease name as input of terms. To identify the co-citation support for the winner scores, we separated the genes into two categories, with literature co-citation ($k = 0$) or without literature co-citation ($k > 0$) to find the differences between the winner scores. We applied the Kruskal-Wallis test to report p-values.

Biomedical case studies, data, and preprocessing

Cardiac regeneration dataset

For the cardiac regeneration case study, the bulk-RNA expression dataset was obtained from Zhang et al. (2020). Briefly, two groups of pig hearts were sent for sequencing when they reached postnatal days (P) 7, 14, and 28. In the first group, the pigs underwent myocardial infarction (a heart attack model) on the postnatal day 1, then their heart fully recovered to normal cardiac functionality with no scar. In the second group, the pig did not undergo injury (sham control). For each group at each day (P7, P14, or P28), three pigs were sequenced. The bulk-RNA data were processed by applying trim-galore (Krueger, 2015) for trimming the fastQ read, then STAR package v2.5.2 for mapping to Pig genome (Dobin et al., 2013), then the RNA transcripts

1 chi2gof: Chi-square goodness-of-fit test [https://www.mathworks.com/help/stats/chi2gof.html].

were counted using HtSeq version 0.6.1 (Anders et al., 2015). The gene expression was normalized, and fold-change was calculated using Deseq2 software (Love et al., 2014). Due to the small sample size ($n = 3$), the p-values for differentially expressed genes, compared between two groups at P7, P14, and P21, were calculated using the approach in Bian et al. (2021). After calculating and comparing two groups at these three different postnatal time points, this process yielded 276 seed genes as input for WINNER. Then, these genes were queried in HAPPI v2 database (Chen et al., 2017) to build their interacting network. These gene lists, their interaction, and WINNER results were summarized in Supplementary Tables 1, 2.

Data processing of triple negative breast cancer (TNBC)

Triple negative breast cancer (TNBC) has been found in 15% of breast cancer cases and is characterized by the tumor cells lacking the expression of the following: epidermal growth factor 2 (HER2), estrogen receptor (ER), and progesterone receptor (PR; Liu et al., 2014; Ueda et al., 2019). Unfortunately, because of its nature, TNBC has a poorer prognosis than other types of breast cancers and treatment options are limited (Xia et al., 2014; Eltohamy et al., 2018; Lu et al., 2020). While TNBC markers are already well-studied, finding the key disease regulators and promising targeted genes is still challenging (Nedeljkovic and Damjanovic, 2019). Therefore, we applied WINNER to explore novel answers for this question.

We took the triple negative breast cancer candidate genes from the University of Alabama at Birmingham Cancer data analysis Portal (UALCAN) database (Chandrashekar et al., 2022). In the comparison between the 116 triple negative breast cancer samples and 114 normal samples, UALCAN provided the top 250 up-regulated genes and 250 down-regulated genes selected by the t-test p-value. Next, we retrieved the Protein-Protein Interaction (PPI) using the medium confidence (score ≥ 0.4) and extended 100 genes using the STRING database. We performed WINNER and generated the gene ranking and p-values (Supplementary Tables 3, 4).

PubMed co-citation analysis of the WINNER ranked genes

We hypothesize that important disease-specific genes are often co-mentioned in a research article (Olsen et al., 2014); if so, WINNER high-ranking genes tend to be more co-cited in the literature than the low-ranking ones. Therefore, to demonstrate the significance of the genes related to a disease, we applied co-citations from the NCBI e-utils application programming interface (API; Sayers, 2008) that implements semantic searches of PubMed abstracts to report biomedical literature citations (https://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?). We applied "pubmed" as an input of the database and the concatenated string of the candidate gene and the disease name as input of terms. To identify the co-citation support for the winner scores, we separated the genes into two categories, the WINNER significant ranked genes (p-value ≤ 0.05) or WINNER non-significant ranked genes (p-value > 0.05) to find the differences between the co-citations. We applied the Kruskal-Wallis test to report p-values to test differences of co-citations between significant and non-significant genes.

Pathway level assignment

We retrieved significantly enriched pathways from PAGER 2.0 database (Yue et al., 2018) using WINNER highly ranked genes with p-values ≤ 0.05. We applied the parameter set as follows. The data sources were KEGG, WikiPathway, BioCarta, NCI-Nature Curated, Reactome, Protein Lounge, and Spike, the similarity was set to be 0.05, and FDR was set to be 0.01. We constructed the regulatory (r-type) PAG-to-PAG network using the default r-type relationship score cutoff (=1). We performed a 5-step procedure in the pathway level assignment. Firstly, we calculated shortest paths among the pairwise r-type PAG-PAG relationships. Secondly, we extracted the longest shortest path and assigned levels of pathway from the upstream to the downstream pathway using 1 to n. Thirdly, we expanded the level assignment to the using shortest distances, such as the current pathway is level m, the shortest distance between the expanded pathway in the upstream to the current pathway is 2, the expanded pathway level will be assigned by m-2. Fourthly, we took the average of the levels assigned to pathways. Fifthly, we repeated the steps three and four until all the pathways had been assigned.

The correlation analysis of WINNER ranking and the enriched pathways using the exponential scale of top gene bins

Firstly, we segregated the WINNER significant genes into 2^x bins. Secondly, we took the top 2^x bins (x is [1, X]) and merge the genes to perform the enrichment analysis. Thirdly, we had the pathways enriched in the top 2^x gene bins minus the pathways enriched in $2^1, \ldots, 2^{x-1}$ to seek the add-on pathways enriched in the top 2^x gene bins. Fourthly, we mapped the levels from the r-type pathway-to-pathway relationships to the add-on enriched pathways in each top 2^x gene bins, and plotted the curve of pathway levels vs. the gene bins. Meanwhile, we performed the Pearson correlation analysis to report the correlation coefficient between the pathways' levels and gene bins.

Results

Characteristics of WINNER ranking

WINNER ranking of undirected networks

When genes in the KEGG [release 50, stored in the PAGER 1.0 database (Yue et al., 2015)] AD pathway

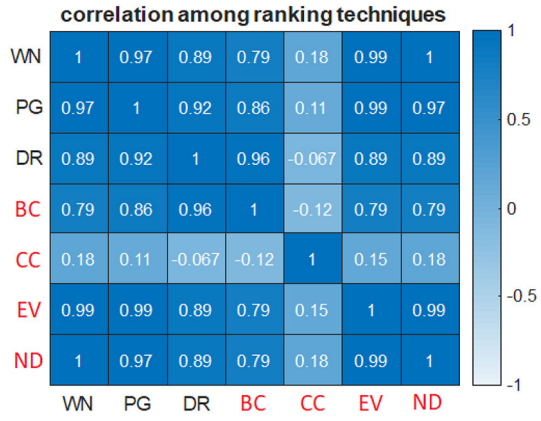


FIGURE 1
WINNER gene prioritization is well-correlated with other ranking techniques and most network topological metrics. Genes in the KEGG AD pathway were ranked *via* WINNER (WN), PageRank (PG), Dual Node-edge Rank (DR), Betweenness Centrality (BC), clustering coefficient (CC), eigenvector centrality (EV), and node degree (ND); then, the correlation coefficients for all pairwise comparisons between ranking methods were calculated *via* Pearson's correlation.

(Supplementary Figure 3) were ranked *via* WINNER gene prioritization, our results were strongly correlated with those obtained *via* analyses of both eigenvector (Newman, 2008; $p = 1.45 \times 10^{-39}$) and node-betweenness (Newman, 2008; $p = 1.67 \times 10^{-11}$) centrality, but not with the clustering coefficient (Newman, 2008; $p = 0.22$). Similar patterns of correlation were obtained with two other state-of-the-art network-based ranking techniques, PageRank (Winter et al., 2012), eigenvector (Newman, 2008), betweenness centrality (Newman, 2005), and dual node-edge ranking (dual rank; Wang et al., 2015) (Figure 1), and all three ranking techniques were strongly correlated with node degree. Notably, the clustering coefficient, but no other metric or technique, failed to identify some of the most important markers for Alzheimer's, including Amyloid Beta Precursor Protein (A4 or APP; Jonsson et al., 2012), Caspase 8 (CASP8; Wei et al., 2002), Caspase 3 (CASP3; D'Amelio et al., 2011), and Presenilin 1 (PSN1; La Bella et al., 2004). Thus, WINNER was at least equivalent to other network topological metrics and well-established prioritization techniques for ranking genes in undirected biological networks.

The strong correlation between the WINNER and node-degree rankings prompted us to preserve the node degree and modularity during randomization. Examining the AD-associated genes network, the pairwise rank differences between the original network and the total-permutation random network were significantly large (Figure 2A). When the difference between the random ranking and the original ranking is too large, the random network topology would be too different from the original network topology; thus, the random ranking may not be suitable to test statistical significance of the original ranking. Besides, when compared to other randomization techniques (total network permutation, preserving modularity, or adding/removing 5% of edges), the distribution of rankings of AD-associated genes in the degree-preserved randomized network was significantly more normally-distributed (Figure 2B). Furthermore, when examining the ranking distributions of two important AD-associated genes A4 and Presenilin 1 (PSN1; Figures 2C,D), it was clear that their distributions had the bell-shape. Thus, rather than relying on the empirical p-value (Cornish et al., 2018) for gene rankings, we generated 1,000 node-preserved randomized networks and calculated a ranking p-value (p_r) for all genes in all KEGG pathways. Notably, the rankings were much less likely to change in response to the addition of noise for genes with $p_r < 0.05$ than for genes with $p_r \geq 0.05$, especially as the amount of noise increased (Figure 3). These observations suggest that when randomized networks are generated with node-degree preservation, fewer randomizations may be required to achieve adequate precision, and fewer noise simulation may be necessary to evaluate the robustness of the rankings.

The accuracy of WINNER gene prioritization was evaluated by ranking genes in the KEGG breast cancer pathway (https://www.genome.jp/kegg-bin/show_pathway?hsa05224) and then determining whether the top-ranked genes correlated with the genes' effect on survival for patients with breast cancer, as estimated with an online Kaplan-Meier (Bland and Altman, 1998) tool that calculates the breast-cancer survival rates associated with more than 6,000 genes (Gyorffy et al., 2010). The KEGG breast cancer pathway contains 146 genes [annotated by UniProt Consortium (2018)], 62% of which significantly influenced patient survival, and a greater proportion of the most highly ranked genes were significantly associated with breast-cancer survival when prioritized with WINNER than with other gene prioritization techniques (PageRank and dual node-edge ranking; Figure 4). Furthermore, the precision of WINNER for retrieving survival-related genes (i.e., the proportion of retrieved genes that were significantly related to breast cancer survival) was even greater when restricted to genes with a ranking p-value of $p_r < 0.05$.

WINNER ranking of directed networks

WINNER ranking of directed networks was evaluated *via* WINNER upstream prioritization with all cancer disease pathways in KEGG release 85 (Kanehisa et al., 2017; KEGG, 2022) and the gene-gene regulatory relationships in STRING v.10.5 (Szklarczyk et al., 2017). Genes were distributed into layers using the breadth-first search approach (Wang et al., 2012) with genes coding for proteins that function further upstream in the pathways assigned to the lower-numbered layers. Thus, genes in the lowest-numbered layers tend to encode master regulatory molecules/receptors and first/second messengers, which are

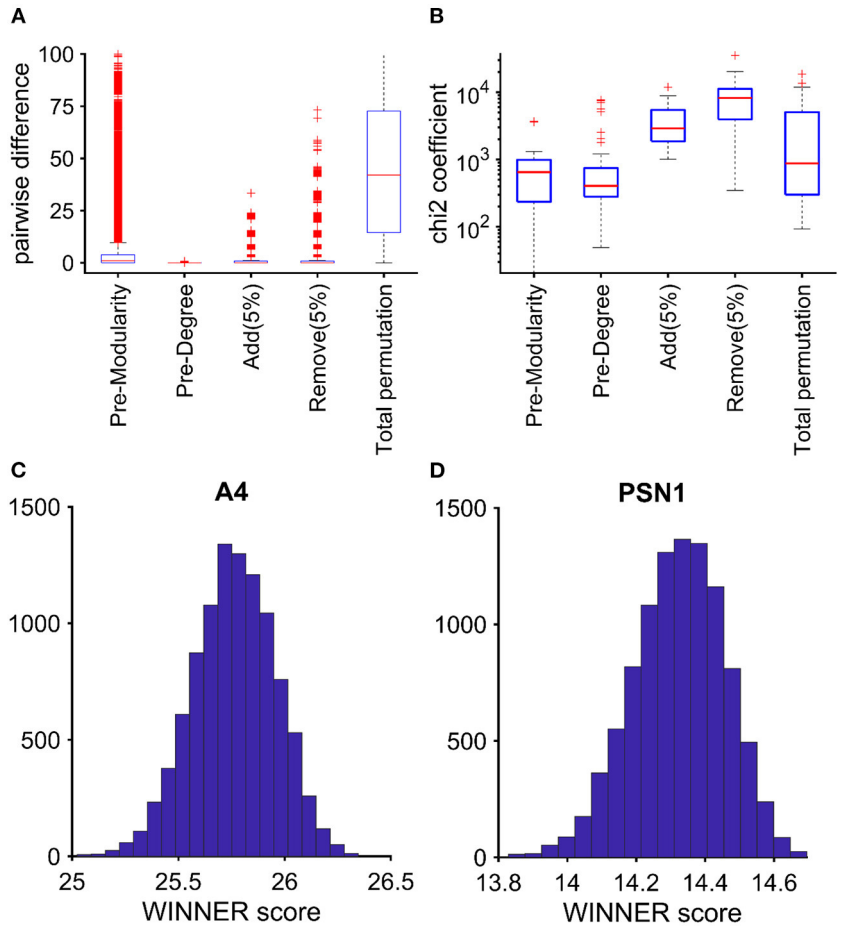


FIGURE 2
With WINNER, Node-degree–preservation and modularity preservation yields more normally distributed randomized networks. Genes in the KEGG AD pathway were ranked *via* WINNER; then, the ranked networks were randomized *via*: preserving node degree (Pre-Degree), preserving modularity (Pre-Modularity), adding 5% interactions [Add (5%)], removing 5% of the interactions [Remove (5%)], and total network permutation. **(A)** The (pairwise) difference between the original network ranking score and the random network ranking score; smaller difference implies the random network approach is more likely to preserve the original network topology. **(B)** Chi-square (chi2coef) coefficient in chi2gof test (https://www.mathworks.com/help/stats/chi2gof.html). Smaller chi2coef implies that the random ranking is more normally distributed. The (+) signs in the boxplots imply outliners (outside 2 and 98% percentiles). Under random network by preserving node degree, WINNER ranking distributions are in bell-shape for two important AD-related genes: A4 **(C)** and PSN1 **(D)**.

located where the signaling cascade originates (e.g., near the cell membrane; Koschmann et al., 2015), while genes with the highest layer numbers tend to encode downstream effector molecules that are closely associated with a specific disease phenotype, such as drug resistance in breast cancer (Johnston, 2006). Our results indicated that using WINNER, layer 1–3 genes, which were the upstream layers in the pathways, were consistently ranked at higher percentiles than genes at other layers (more downstream; Figure 5). But this consistency was not observed when the genes were prioritized *via* equivalent (directed-network ranking) analyses with PageRank (Winter et al., 2012) and dual node-edge ranking (Wang et al., 2015). WINNER upstream overestimated the ranking of genes in layer 8, but this can likely be attributed to noise, because the layer contained only 12 ranked genes.

WINNER network expansion and ranking upstream regulators

We demonstrated how WINNER could identify upstream regulators of two cancer pathways, Chronic Myeloid Leukemia (CML; https://www.genome.jp/kegg-bin/show_pathway?hsa05220) and hepatocellular carcinoma (https://www.genome.jp/pathway/hsa05225), that were missing from the existing pathways in KEGG but were present in the KEGG database itself. WINNER upstream prioritization distributed genes into five different layers for each pathway, and WINNER expansion added several highly ranked genes to both networks. Additions to the CML network (Figure 6) included JAK1/2/3 and proteins that participate in IL-2 (IL2, IL2RA, and IL2RB), IL-3 (IL-3, IL-3RA, and IL-3RB), and GM-CSF (CSF2) signaling, which is consistent with the JAK2/STAT5 pathway's status as one of the

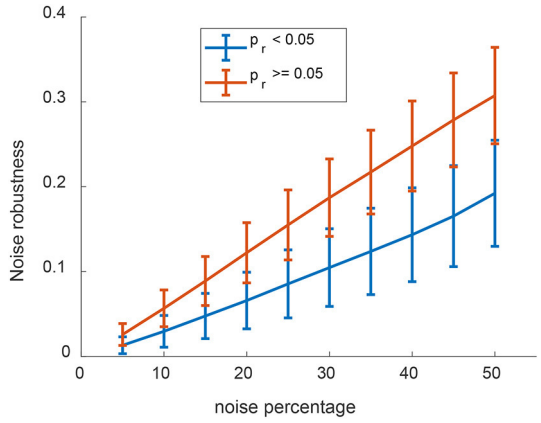


FIGURE 3
The WINNER ranking p-value (p_r) is robust to the addition of noise (STATS?). Genes in all KEGG pathways were ranked *via* WINNER, and WINNER ranking p-values (p_r) were calculated, after varying degrees of noise were added to the network; then, noise robustness was compared for genes with $p_r < 0.05$ and $p_r \geq 0.05$ by determining the likelihood that the gene's ranking changed by 10 or more upon the addition of noise.

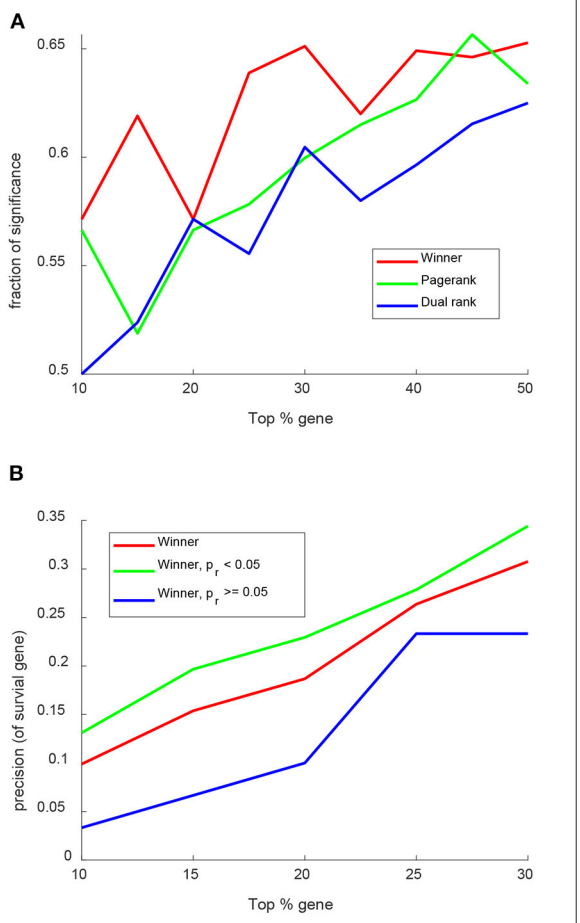


FIGURE 4
WINNER gene prioritization more accurately identifies the relationship between breast-cancer genes and patient survival. Genes in the KEGG breast-cancer pathway were ranked *via* WINNER, PageRank, and Dual Rank, and the significance of each gene's relationship to patient survival was determined with an online Kaplan-Meier plotting tool. **(A)** The proportion of genes that were significantly ($p < 0.05$) related to breast-cancer survival was determined for the top 0-50% of ranked genes. **(B)** The precision of the WINNER ranking of genes for breast-cancer survival (Bland and Altman, 1998) was compared for the top 0–30% of ranked genes with $p_r < 0.05$ and $p_r \geq 0.05$.

primary targets for treatment of CML (Valent, 2014), as well as evidence that STAT5 is phosphorylated by IL-2 (Kobayashi et al., 2014; Valent, 2014) and IL-3 (Jiang et al., 1999) signaling, and that GM-CSF is a crucial growth factor for myeloid cells; notably, several of these molecules are currently being investigated as therapeutic targets for CML treatment (Hercus et al., 2012; Broughton et al., 2014; Kobayashi et al., 2014). For the hepatocellular carcinoma pathway (Figure 7), WINNER expansion added KC1G2, a serine-threonine kinase that can activate TGF-β1/Smad signaling (Guo et al., 2008); TMED4, WLS, and PRCN, which mediate Wnt/β-catenin signaling (Guo et al., 2008; Martin-Orozco et al., 2019; Bland et al., 2021); and several genes for proteins in the FGF signaling pathway (FRS2, FRS3, KLB, and PLCG1; Gotoh, 2008; Gyanchandani et al., 2013; Wang et al., 2020), of which KLB is particularly important, because it functions as a co-receptor for the binding of FGF-19/21 to FGFR-1/4 (Yang et al., 2012). Thus, the genes added to the KEGG CML and hepatocellular carcinoma pathways by WINNER expansion have strong, well-established links to multiple binding partners that participate in the mechanisms associated these diseases.

Besides, WINNER ranking correlation with other ranking techniques, including Ingenuity Pathway Analysis (IPA; Kramer et al., 2014), DIAMOnD (Ghiassian et al., 2015), Random Walk (Smedley et al., 2014), Node2Vec (Grover and Leskovec, 2016; Peng et al., 2019), and GenePANDA (Yin et al., 2017), vary from −0.83 (negatively correlated) to −0.05 (insignificant correlation), then to 0.74 (moderate-positively correlated; Figure 6C). This result suggests that the major difference between WINNER and other techniques' ranking appears when the network expands beyond the seed genes. Thus, a good benchmark among WINNER and other techniques can be performed by a network-expansion scenario.

Benchmarking WINNER ranking by retrieving newly updated genes in KEGG pathways

Gene prioritization algorithms are benchmarked by information retrieval experiments, such as in Guala and Sonnhammer (2017) and Zhang et al. (2021), where some

important regulators are labeled "unknown," and the algorithms are executed to rank these "unknown-labeled" gene such that these regulators are top-ranked. Thus, to benchmark WINNER, we setup the KEGG Pathway retrieval experiment. Here, WINNER took a KEGG pathway release 50 (2009 version; Kanehisa et al., 2010) as the seed genes and gene-gene interactions (expanded network) in HAPPI database (Chen et al., 2017) as the input; the WINNER expansion p-value (p_e) and WINNER score were calculated for candidate genes to include in the KEGG release 50 pathway networks; then, the highly-ranked non-seed (expanded genes) was compared to the same updated pathway network in KEGG release 85 (Ogata et al., 1999; Kanehisa et al., 2017; 2017 version) as the ground-truth. In this experiment, WINNER performance, quantified by precision, recall, and the F1 score, was compared with Ingenuity Pathway Analysis (IPA; Kramer et al., 2014), DIAMOnD (Ghiassian et al., 2015), Random Walk (Smedley et al., 2014), Node2Vec (Grover and Leskovec, 2016; Peng et al., 2019), and GenePANDA (Yin et al., 2017); these techniques were chosen according to Zhang et al. (2021). The same experiment was executed with each KEGG pathway, and the results were aggregated into error bars.

Our results indicated that the WINNER predictions had greater precision but less recall (i.e., the proportion of newly incorporated genes that were retrieved by the prediction) than the predictions generated *via* other comparing methods (Figure 8). The WINNER predictions were also associated with a higher F1 score, which incorporates both precision and recall into a global measure of accuracy, when more than 60% of the extension candidates were examined. Besides, Figure 8 shows that the retrieval recall rate is low (usually <0.2) in all of the algorithms. Precision should be prioritized in comparing the performance among these expansion algorithms.

WINNER ranking of differentially expressed genes in biological case-studies

WINNER ranking of genes involved in apoptosis and cell-cycle activity

The use of WINNER for prioritizing genes involved in cellular processes was evaluated with the KEGG apoptosis and cell-cycle pathways and node-degree–preserved network randomization. WINNER ranking p-values were highly significant for genes that participate in some of the most essential mechanisms of apoptosis, such as Phosphatidylinositol 4,5-bisphosphate 3-kinase catalytic subunit alpha isoform (PIK3CA) ($p_r = 5.01\times10^{-13}$); the Phosphatidylinositol 3-kinase regulatory subunit alpha (P85A; $p_r = 1.34 \times 10^{-12}$) and Cytokine receptor common subunit beta (IL3RB; $p_r = 4.60 \times 10^{-12}$); and genes for several proteins of the cytoskeleton (actin, $p_r = 1.94\times10^{-104}$; Tubulin, $p_r = 1.94\times10^{-104}$; B4DZT3, $p_r = 8.71\times10^{-87}$; Lamin A/C, $p_r = 8.17\times10^{-87}$; Lamin B1, $p_r = 8.17\times10^{-87}$; actin-G, $p_r = 5.15\times10^{-63}$), which is substantially reorganized to produce the characteristic shrunken morphology of apoptotic cells; notably, actin and actin-binding proteins also initiate and regulate apoptosis (Desouza et al., 2012). However, the KEGG apoptosis pathway also includes genes for a number of proteins that participate IL-3– and NGF-signaling (IL-3, IL-3R, and NGF), which are nonessential (or even irrelevant) for apoptosis, and the ranking p-values calculated for these genes were not significant ($p_r = 0.18$). Similarly, genes in the KEGG cell-cycle pathway that encode proteins directly involved in DNA replication and cell division had highly significant ranking p-values (Cell Division Cycle 14B, $p_r = 9.5\times10^{-297}$ and 14A, $p_r = 2.28\times10^{-22}$) whereas the ranking p-values for genes that participate in TGF-β signaling were nonsignificant (TGF-β, $p_r = 0.29$; SMAD2, $p_r = 0.29$; SMAD3, $p_r = 0.29$; SMAD4, $p_r = 0.29$), which is consistent with the role of TGF-β in cell-proliferation: it interacts with many components of the cell cycle pathway but generally inhibits proliferation in non-mesenchymal cells. Collectively, these observations demonstrate that the WINNER ranking p-value can be a useful guide for distinguishing between genes that are essential or nonessential participants in a particular cellular process.

WINNER ranks important signaling pathway markers in mammalian pig heart regeneration

The hearts of adult mammals cannot regenerate myocardial tissues that are lost to injury; however, when myocardial infarction (MI) was induced in the hearts of one-day-old piglets, the animals recovered with no significant loss of cardiac function and little evidence of myocardial scarring (Zhu et al., 2018). Thus, to identify genes that may contribute to mammalian cardiac regeneration, we used WINNER to rank the list of differentially expressed genes from piglets that had or had not undergone surgically induced MI on postnatal day 1 for a previous report (Zhang et al., 2020; Figure 9, Supplementary Table 1). Here, we used HAPPI version 2 database (Chen et al., 2017) to build the network connecting these genes. The two top-ranked genes (FN1 and JAK3) encoded fibronectin, which is required for cardiac regeneration in zebrafish (Wang et al., 2013), and Janus kinase 3 (JAK3), which has been shown to protect against ischemia-reperfusion injury (Kubin et al., 2011); notably, JAK3 also interacts with oncostatin-M, which is encoded by the tenth-highest WINNER-ranked gene (OSM) and is a primary factor in cardiomyocyte dedifferentiation and remodeling (Singh et al., 2016; Doll et al., 2017). Also among the top 10 were genes encoding subunits of the essential matrix proteins integrin alpha (ITGA8) and beta (ITGB4), which are differentially expressed in adult and fetal cardiac fibroblasts and involved in chamber specification of zebrafish hearts (Singh et al., 2016; Doll et al., 2017), while the 11th-ranked gene, THBS3, encodes another extracellular matrix protein, thrombospontin

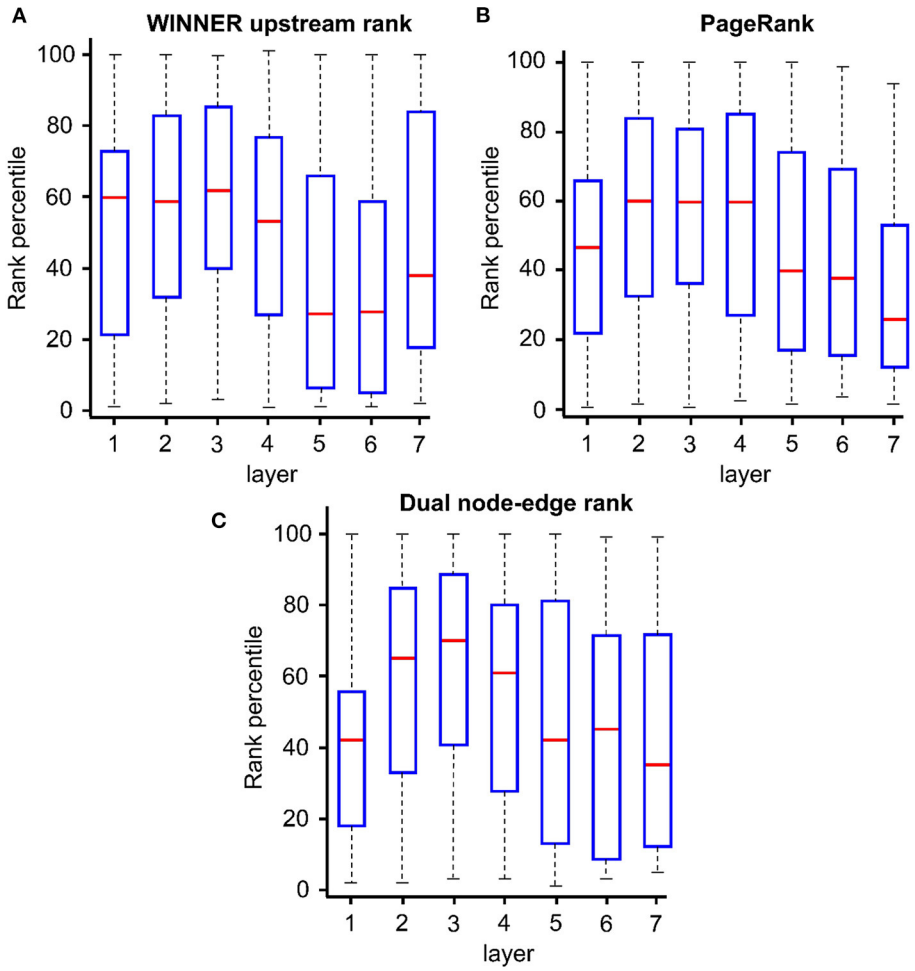


FIGURE 5
WINNER upstream prioritization more accurately identifies the relative position of genes in a pathway. Gene-gene regulatory relationships from STRING v.10.5 were used to distribute genes from all KEGG cancer pathways into 7 layers *via* WINNER (customized for upstream ranking), PageRank, and Dual Rank; genes coding for proteins that function further upstream in the pathways were assigned to the lower-numbered layers. Layers 1–3 are the most upstream layers, usually correspond to the kineases, grow factors, and receptors. Layers 4–7 are downstream, usually correspond to signaling hubs, phospholization, transcription factors, and inside-nucleus genes. The y axis indicates the ranking scores, which were converted into percentile so that the rankings across different pathways could be combined into one boxplot. The red cross implies boxplot outliners (beyond 2 and 98% percentiles). **(A)** WINNER upstream rank. **(B)** PageRank. **(C)** Dual node-edge rank.

3, which is a critical [and clinically relevant (Mustonen et al., 2013)] regulator of cell-cell and cell-matrix signaling that appears to impede integrin function and contribute to injury-induced cardiomyopathy in mice (Costa et al., 2014; Porrello and Olson, 2014; Puente et al., 2014). Other genes ranked among the top 20 by WINNER included the nitrous-oxide–related genes NCF2 and NCF4, and the gene for vasopressin 2 (AVPR2), which collectively modulate the cellular environment to promote cardiac regeneration (Costa et al., 2014; Porrello and Olson, 2014; Puente et al., 2014); ERBB3, which encodes a tyrosine kinase that appears to be crucial for embryonic development (Erickson et al., 1997); and genes for a dynamin protein (DNM1) and a Rho GTPase (RND2), which suggests that at least some of the mechanisms of mammalian myocardial regeneration are mediated by vesicle-based signaling.

WINNER ranking reflects the important genes supported by co-citations and reveals the upstream events in the r-type pathway-to-pathway network in triple negative breast cancer (TNBC) study

We found 72 significant genes ranked by WINNER using p-value ≤ 0.05 with the WINNER score ranging from 7.4 to 92.5, and the left nonsignificant genes' WINER score ranges from 0 to 68.7. The co-citations analysis shows that the "triple negative breast cancer" co-citations between the significant

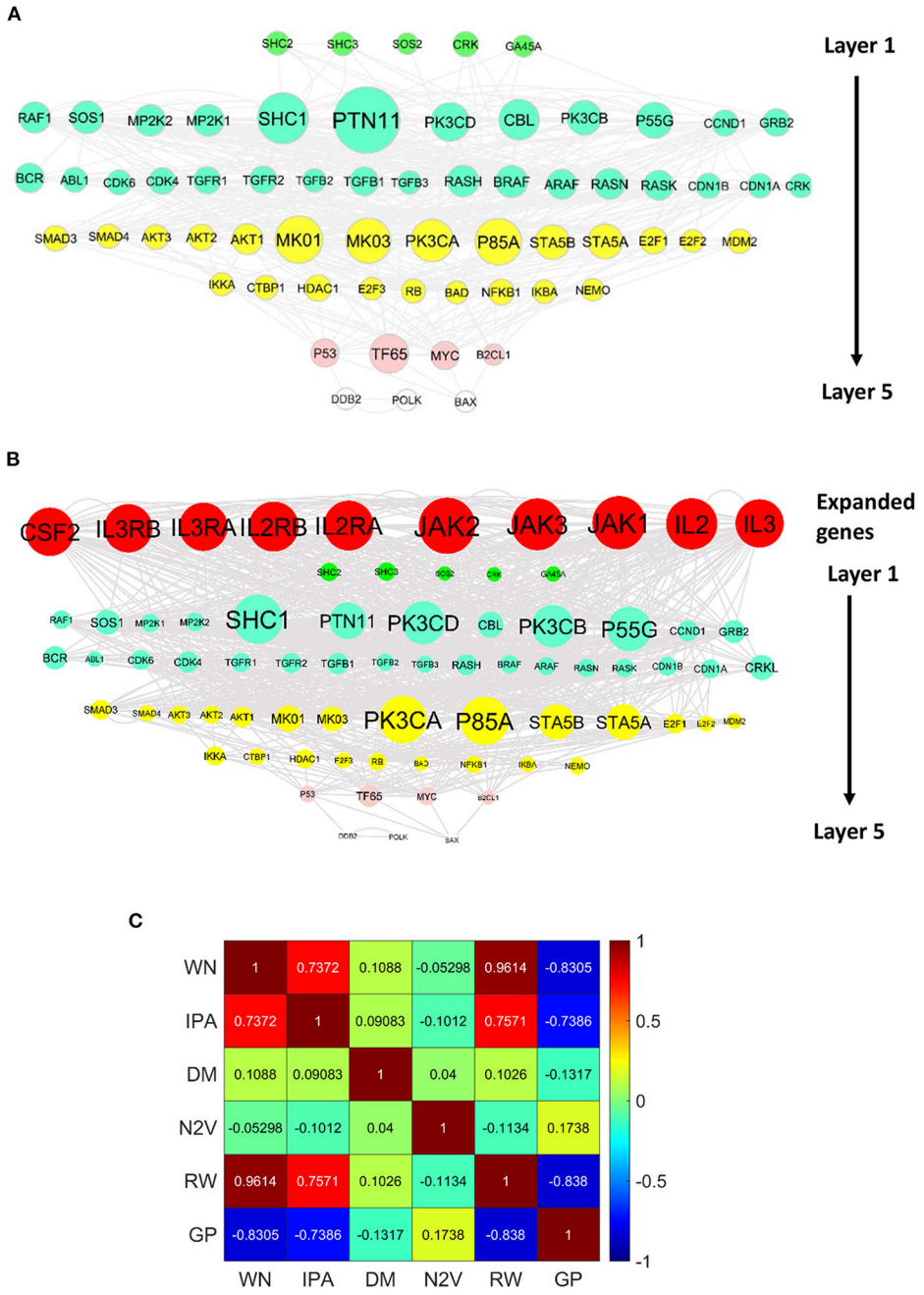


FIGURE 6
WINNER upstream ranking and expansion can identify genes that are missing from established chronic myeloid leukemia (CML) networks. Genes in the KEGG CML pathways were distributed into layers *via* WINNER upstream, and genes that were missing from the networks were identified *via* WINNER expansion. Genes in the same layer are displayed in the same color, and the size of the node represents the WINNER score. **(A)** WINNER ranking without expansion. **(B)** WINNER ranking with expanded genes. **(C)** Correlation among WINNER (WN), Igenunity Pathway Analysis (IPA), DIAMOnD (DM), Node2Vec (ND), Random Walk (RW), and GenePANDA (GP) ranking.

ranked genes and the nonsignificant ranked genes have significant difference with Kruskal Wallis test's p-value = 0.027 (Figure 10). The result suggests that WINNER's high-rank genes are more likely lead to biological insights than the WINNER's low-rank genes.

To explore new insights among the high-ranking genes, we performed pathway analysis and built the pathway-to-pathway regulatory networks from these genes using PAGER tool (Yue et al., 2018). The WINNER significantly ranked genes regulated many implicated pathways and processes for TNBC. Thus, we

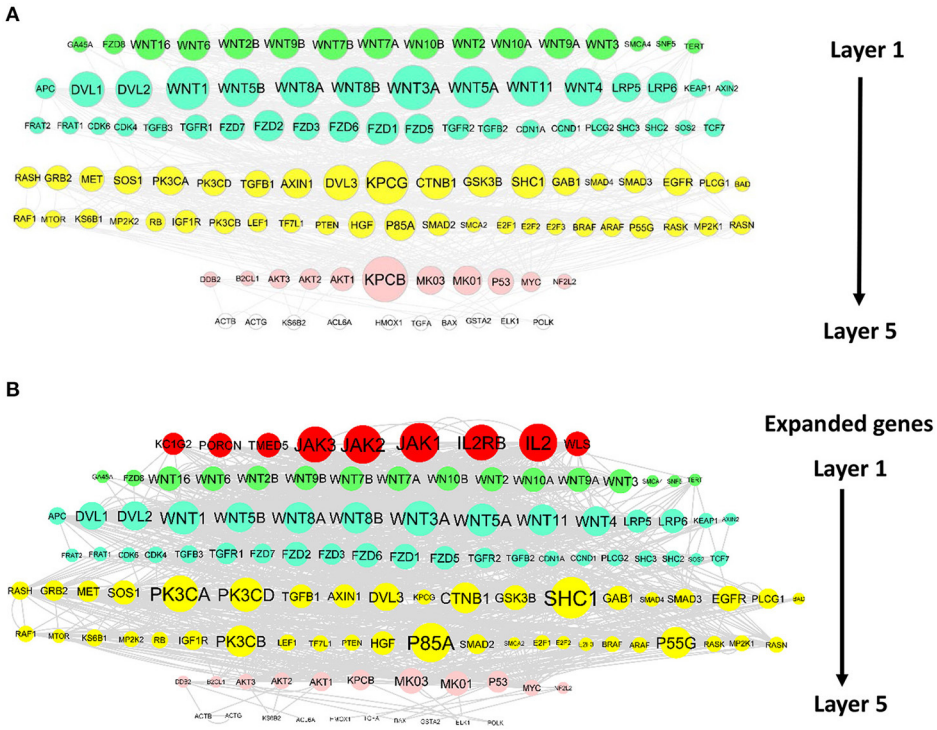


FIGURE 7
WINNER upstream ranking and expansion can identify genes that are missing from established hepatocellular carcinoma networks. Genes in the KEGG hepatocellular carcinoma pathways were distributed into layers *via* WINNER upstream, and genes that were missing from the networks were identified *via* WINNER expansion. Genes in the same layer are displayed in the same color, and the size of the node represents the WINNER score. **(A)** WINNER ranking without expansion. **(B)** WINNER ranking with expanded genes.

observed the higher ranked gene enriched pathways are more likely to be at upstream side of the regulatory (r-type) enriched pathway-to-pathway network. In general, the add-on pathway levels were positive correlated to the ranked gene bins with Pearson correlation coefficient equal to 0.74 (Figure 11).

We found that the top ranked genes, TOP2A, CDK1, PLK1, and UBE2C, were enriched in the cell cycle related pathways, such as "Phosphorylation of Cyclin B1 in the CRS domain," "Regulation of mitotic cell cycle," "Mitotic Metaphase and Anaphase," and "Free APC/C phosphorylated by Plk1."

Topoisomerase II a (TOP2A) can be a useful gene in determining whether TNBC patients would have a good response to anthracycline therapy, which is the mainstay treatment in TNBC cancer (Brase et al., 2010; Di Leo et al., 2011; Eltohamy et al., 2018). Both Eltohamy et al. and Di Leo et al. found that patients with aberrant expression of TOP2A have better response to anthracycline treatment (Di Leo et al., 2011; Eltohamy et al., 2018).

Cyclin dependent kinase 1 (CDK1) play a critical role how the cell cycle is regulated, specifically during mitosis. Liu et al. used nanoparticles with siRNA to target CDK1, and it has been found to successfully inhibit the TNBC cell line that has been injected in mice (Liu et al., 2014). Xia et al. has found that the CDK1 inhibitor can inhibit the growth of the TNBC cells by arresting them in the G2/M cell phase (Xia et al., 2014).

Polo like kinase-1 (PLK1) has been found to be one of the key regulators in the cell cycle. Targeting and knocking out of PLK1 has been found to cause the TNBC tumor cells to be arrested in the G2-M cell cycle (Ueda et al., 2019; Zhao et al., 2021; Patel et al., 2022). Morray et al. found that a nanoparticle with siRNA targeting PLK1 can inhibit growth in the TNBC tumor cell line (Morry et al., 2017). Patel et al. used the allosteric inhibitor RK-10 to target the PLK1 in TNBC cell lines, and it has inhibited growth through the S phase and G2/M (Patel et al., 2022).

Overexpression of Ubiquitin-conjugated enzyme (UBE2C) can play a role in the pathogenesis of TNBC (Chou et al., 2014; Kim et al., 2019). Chou et al had found that UBE2C has been highly expressed in cancer tissue cells, and that when UBE2C has been targeted with siRNA, the tumor cells have stopped proliferating (Chou et al., 2014).

Discussion and conclusion

In this paper, we introduce WINNER, a new network-based ranking tool that addresses several of the limitations associated with other gene prioritization techniques. Our

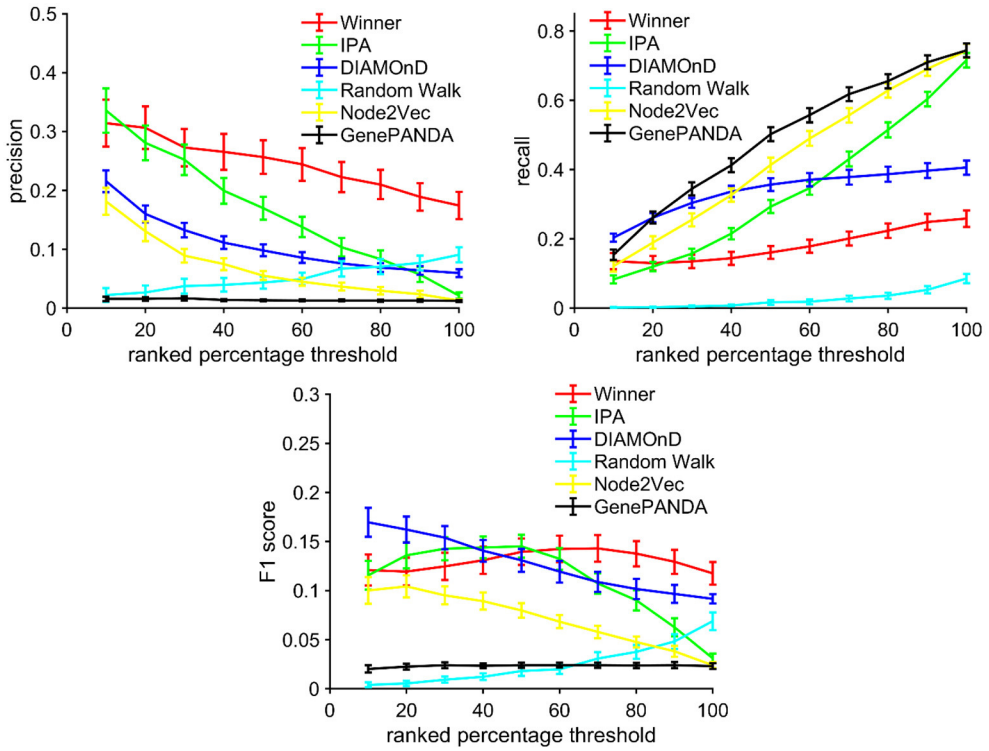


FIGURE 8
Benchmark: WINNER expansion more accurately identifies the addition of new genes to established networks. The pathway networks in KEGG (https://www.genome.jp/kegg/network.html) release 50 was expanded *via* WINNER (i.e., calculation of the WINNER expansion p-value), Ingenuity Pathway Analysis (IPA), DIAMoND, Random Walk, Node2Vec, and GenePANDA. Then, the expanded networks were compared to the updated network in KEGG release 85 to determine the precision, recall, and F1 scores for each expansion technique.

novel use of node-degree–preserved and modularity-preserved randomization produced randomized networks that retained some of the original network topology and were more normally distributed, which increased the precision and robustness of our ranking p-value (p_r) calculations, while the expansion p-value (p_e) better accommodated the incomprehensiveness and redundancy of the input gene list. However, WINNER rankings were not well-correlated with the clustering coefficient, which represents the presence of network cliques (Newman, 2008; i.e., semi-isolated groups of genes that collectively function like a single node), which suggests that WINNER ranking may be somewhat compromised in dense networks, such as those containing families of proteins, where the scale-free property (Timar et al., 2016) does not apply. Nevertheless, many biological networks are scale-free (Khanin and Wit, 2006), and since degree-preserved randomization tends to produce near-normal ranking distributions, the WINNER p_r value is likely more accurate than the empirical p-value, even for networks that are not perfectly scale-free.

WINNER network ranking belongs to the "eigenvector ranking" (Newman, 2008) class of algorithm. Therefore, it has the same "big-O" computational cost to PageRank [$O(N^3)$, where N is the number of network genes] if implemented using iterative matrix multiplication. However, this class of algorithm can be implemented in parallel, which significantly reduced the computational time in practice.

The performance of gene network prioritization significantly depends on the disease (Zhang et al., 2021), or the biological case-study. Therefore, we demonstrate WINNER's performance in various disease and biological study scenarios. The comprehensive KEGG pathway results reflect the case when lacking biological samples and expression data. Then, prioritization needs to be performed only using the domain-knowledge available network to generate hypotheses. Cardiac regeneration, which focuses on cardiomyocyte proliferation, case-study is an example when a significant biological process, not a disease, that does not naturally happen in matured mammals (Porrello et al., 2011; Lam and Sadek, 2018; Ye et al., 2018; Zhu et al., 2018; Zhao et al., 2020; Nakada et al., 2021; Nguyen et al., 2022). In this case, the focus is finding the regulating mechanism to create new cells and to apply this knowledge in biomedical engineering research. Cancer and other disease case studies (leukemia, TNBC, and Vitamin D) are directly related to the disease, and targeted therapies to kill cells are available or proposed. In this case, the focus is to find

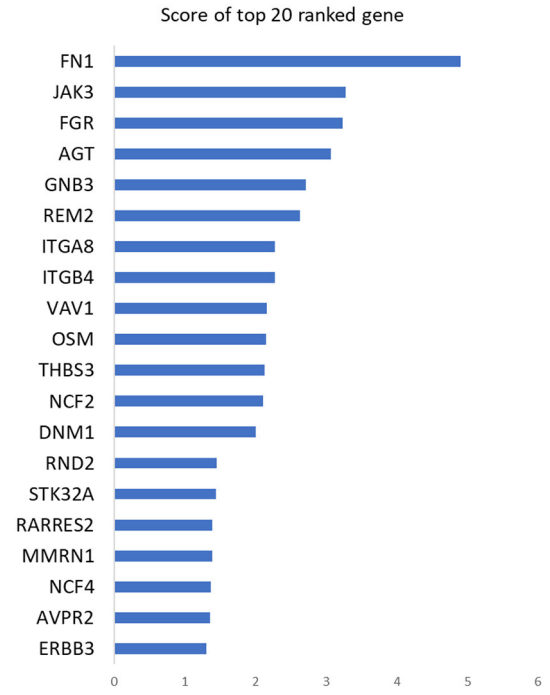


FIGURE 9
WINNER can identify genes that contribute to cardiac regeneration from a list of differentially expressed genes. RNA-sequencing analyses of gene expression in the hearts of piglets that had or had not undergone surgically induced myocardial infarction on the 1st day after birth for a previous report (Zhu et al., 2018) were compared to generate a list of differentially expressed genes; then their gene-gene interactions were queried from HAPPI v2 database; then, the list was ranked *via* WINNER gene prioritization to determine which genes likely contributed to myocardial regeneration. The 20 top-ranked genes are displayed with their corresponding WINNER scores.

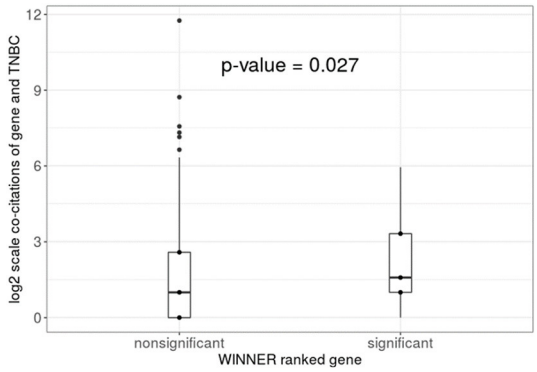


FIGURE 10
The literature validation of triple negative breast cancer genes using co-citations from PubMed. The co-citations of gene and TNBC are grouped by the WINNER reported *p*-values. The non-significant gene *p*-values are larger than 0.05 in WINNER, and the significant gene *p*-values are ≤0.05 in WINNER. The Kruskal Wallis test *p*-value is 0.027.

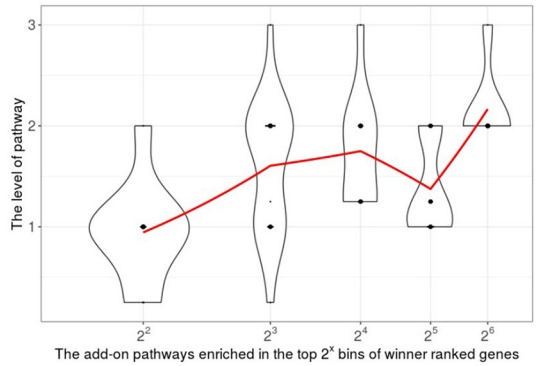


FIGURE 11
The correlation between the add-on pathways enriched in the top 2^x bins and the bin size. The violin plot shows the pathway level distribution. The red points connected by solid red lines are the means of pathway levels.

markers, especially the "cell-killer ones" associated with the disease outcomes, and there is less emphasis rather than the regulating growing mechanism. WINNER results are insightful in all of these cases, whereas whether other techniques have insightful results is yet to be examined in multiple studies.

In conclusion, WINNER gene prioritization is generally more accurate and robust than other network-based prioritization techniques, such as PageRank and node-degree ranking, and can be effective for identifying genes that may be missing from established gene networks, for determining the relative position (i.e., upstream or downstream) of genes within a pathway, and for ranking a list of differentially expressed genes. The superior performance is linked to better retrieval precision when expanding the network among the seed genes. The important case studies presented in this work are in a scenario where new disease-specific gene-expression data were generated, and novel genes associated with the disease and phenotype are expected. Then, network expansion is required. In this expansion, WINNER emphasizes precision, where only a small expanded but highly relevant candidates are explored, over recall, where more comprehensive candidate genes were explored but may involve many irrelevant ones. Other methods tend to emphasize recall; therefore, they may computationally retrieve more candidates; however, at the same time, make it much more difficult for the user to choose the rightly relevant ones. Also, having too many irrelevant genes in the network significantly affects the ranks of the well-known disease-specific genes. This scenario explains the advantage of WINNER over other methods. Future investigations are warranted to determine what additional biological insights can be obtained by using WINNER to rank genes that participate in other cellular processes, in metabolic regulatory pathways (Berkhout et al., 2013), and in co-expression networks (Radulescu et al., 2018).

Data availability statement

The gene expression data used in this work are publicly available at the Gene Expression Omnibus database, accession number GSE144883, https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE144883.

Author contributions

TN developed the algorithm, performed case studies, and wrote the manuscript. ZY and RS performed case studies and performed the literature validation of the results. ZY built the website. RW and JZ provided data and participated in the case studies. JC conceptualized the ideas, helped design the analytical experiments, and revised the final manuscript. All authors read, edited, and approved the manuscript.

Funding

The work was in part supported by the internal University of Alabama at Birmingham research grants to JC, the National Institutes of Health grant awards U54TR001005 in which JC serves as a co-investigator, and R01 awards R01HL150078 in which RW serves as principle investigator and JC serves as co-investigator.

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

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fdata.2022.1016606/full#supplementary-material

SUPPLEMENTARY FIGURE 1
Schematic diagrams of WINNER gene prioritization and network expansion. **(a)** Seeded genes (green) and candidate expansion genes (yellow) are assembled into a network as indicated by their pairwise interactions. **(b)** The expansion p-value (p_e) are calculated among the expansion-candidate genes, then genes with $p_e < 0.05$ will be further evaluate and added into and expand the network, one gene at a time. Then **(c)** the expansion score **(e)** are calculated for the candidate expansion genes; then, the highest-scored gene is added to the network; this process is repeated until all candidates are added or being halted (not adding all candidates). And **(d)**, after completing the expansion, the statistical significance of the rankings are recalculated for the expanded network.

SUPPLEMENTARY FIGURE 2
WINNER filtering of candidate genes for network expansion. Red nodes represent seeded genes, open nodes represent candidate expansion genes, black lines represent interactions between two seeded genes, and gray lines represent interactions between one seeded gene and one expansion gene or between two expansion genes. Candidate genes for network expansion were filtered *via* two tests: (1) the likelihood of the candidate expansion gene (E.Gene) having a seeded interaction relative to its total number of interactions (bottom left table), and (2) the likelihood of the candidate expansion gene having seeded interactions relative to the seeded interactions of its most similar seeded gene (S.Gene), with similarity determined by node degree (bottom right table).

SUPPLEMENTARY FIGURE 3
WINNER ranking of the network of Alzheimer's disease pathways in KEGG release 50. The network graph was constructed with Cytoscape (Shannon et al., 2003) version 3.6.0 and the force-directed layout; the size of the node represents the WINNER score.

SUPPLEMENTARY TABLE 1
WINNER ranking for genes in cardiac regeneration dataset. The table includes gene symbol, the indication of whether a gene is a seeded (S) or expanded (E) gene, and WINNER score.

SUPPLEMENTARY TABLE 2
Gene-gene interaction network in the cardiac regeneration dataset.

SUPPLEMENTARY TABLE 3
WINNER ranking for genes in triple negative breast cancer (TNBC) dataset. The table includes gene symbol, the indication of whether a gene is a seeded (S) or expanded (E) gene, WINNER score, and p-value.

SUPPLEMENTARY TABLE 4
Gene-gene interaction network in triple negative breast cancer (TNBC) dataset.

SUPPLEMENTARY VIDEO 1
The .cys (cytoscape) file of the regulatory (r-type) pathway-to-pathway network in the triple negative breast cancer study.

References

Aerts, S., Lambrechts, D., Maity, S., Van Loo, P., Coessens, B., De Smet, F., et al. (2006). Gene prioritization through genomic data fusion. *Nat. Biotechnol.* 24, 537–544. doi: 10.1038/nbt1203

Alvarez-Ponce, D., Lopez, P., Bapteste, E., and McInerney, J. O. (2013). Gene similarity networks provide tools for understanding eukaryote origins and evolution. *Proc. Natl. Acad. Sci. U. S. A.* 110, E1594–1603. doi: 10.1073/pnas.1211371110

Anders, S., Pyl, P. T., and Huber, W. (2015). HTSeq–a Python framework to work with high-throughput sequencing data. *Bioinformatics* 31, 166–169. doi: 10.1093/bioinformatics/btu638

Antanaviciute, A., Daly, C., Crinnion, L. A., Markham, A. F., Watson, C. M., Bonthron, D. T., et al. (2015). GeneTIER: prioritization of candidate disease genes using tissue-specific gene expression profiles. *Bioinformatics* 31, 2728–2735. doi: 10.1093/bioinformatics/btv196

Beissbarth, T., and Speed, T. P. (2004). GOstat: find statistically overrepresented Gene Ontologies within a group of genes. *Bioinformatics* 20, 1464–1465. doi: 10.1093/bioinformatics/bth088

Berkhout, J., Teusink, B., and Bruggeman, F. J. (2013). Gene network requirements for regulation of metabolic gene expression to a desired state. *Sci. Rep*. 3, 1417. doi: 10.1038/srep01417

Bian, W., Chen, W., Nguyen, T., Zhou, Y., and Zhang, J. (2021). miR-199a overexpression enhances the potency of human induced-pluripotent stem-cell-derived cardiomyocytes for myocardial repair. *Front. Pharmacol*. 12, 673621. doi: 10.3389/fphar.2021.673621

Bland, J. M., and Altman, D. G. (1998). Survival probabilities (the Kaplan-Meier method). *Br. Med. J*. 317, 1572. doi: 10.1136/bmj.317.7172.1572

Bland, T., Wang, J., Yin, L., Pu, T., Li, J., Gao, J., et al. (2021). WLS-Wnt signaling promotes neuroendocrine prostate cancer. *iScience* 24, 101970. doi: 10.1016/j.isci.2020.101970

Bowman, A. W., and Azzalini, A. (1997). *Applied Smoothing Techniques for Data Analysis: The Kernel Approach with S-Plus Illustrations, vol. 18*. Oxford: Oxford University Press.

Brase, J. C., Schmidt, M., Fischbach, T., Sultmann, H., Bojar, H., Koelbl, H., et al. (2010). ERBB2 and TOP2A in breast cancer: a comprehensive analysis of gene amplification, RNA levels, and protein expression and their influence on prognosis and prediction. *Clin. Cancer Res*. 16, 2391–2401. doi: 10.1158/1078-0432.CCR-09-2471

Bromberg, Y. (2013). Chapter 15: disease gene prioritization. *PLoS Comput. Biol*. 9, e1002902. doi: 10.1371/journal.pcbi.1002902

Broughton, S. E., Hercus, T. R., Hardy, M. P., McClure, B. J., Nero, T. L., Dottore, M., et al. (2014). Dual mechanism of interleukin-3 receptor blockade by an anti-cancer antibody. *Cell Rep*. 8, 410–419. doi: 10.1016/j.celrep.2014.06.038

Cantor, R. M., Lange, K., and Sinsheimer, J. S. (2010). Prioritizing GWAS results: a review of statistical methods and recommendations for their application. *Am. J. Hum. Genet*. 86, 6–22. doi: 10.1016/j.ajhg.2009.11.017

Chandrashekar, D. S., Karthikeyan, S. K., Korla, P. K., Patel, H., Shovon, A. R., Athar, M., et al. (2022). UALCAN: an update to the integrated cancer data analysis platform. *Neoplasia* 25, 18–27. doi: 10.1016/j.neo.2022.01.001

Chatr-Aryamontri, A., Breitkreutz, B. J., Heinicke, S., Boucher, L., Winter, A., Stark, C., et al. (2013). The BioGRID interaction database: 2013 update. *Nucleic Acids Res*. 41, D816–823. doi: 10.1093/nar/gks1158

Chen, J., Bardes, E. E., Aronow, B. J., and Jegga, A. G. (2009). ToppGene Suite for gene list enrichment analysis and candidate gene prioritization. *Nucleic Acids Res*. 37, W305–311. doi: 10.1093/nar/gkp427

Chen, J. Y., Mamidipalli, S., and Huan, T. (2009). HAPPI" an online database of comprehensive human annotated and predicted protein interactions. *BMC Genomics* 10, S16. doi: 10.1186/1471-2164-10-S1-S16

Chen, J. Y., Pandey, R., and Nguyen, T. M. (2017). HAPPI-2: a comprehensive and high-quality map of human annotated and predicted protein interactions. *BMC Genomics* 18, 182. doi: 10.1186/s12864-017-3512-1

Chen, J. Y., Pinkerton, S. L., Shen, C., and Wang, M. (2006b). "An integrated computational proteomics method to extract protein targets for fanconi anemia studies," in *21st Annual ACM Symposium on Applied Computing*. Dijon, 173–179. doi: 10.1145/1141277.1141316

Chen, J. Y., Piquette-Miller, M., and Smith, B. P. (2013). Network medicine: finding the links to personalized therapy. *Clin. Pharmacol. Therapeut*. 94, 613–616. doi: 10.1038/clpt.2013.195

Chen, J. Y., Shen, C., and Sivachenko, A. Y. (2006a). Mining Alzheimer disease relevant proteins from integrated protein interactome data. *Pacific Symposium on Biocomputing Pacific Symposium on Biocomputing* 2006, 367–378. doi: 10.1142/9789812701626_0034

Chou, C. P., Huang, N. C., Jhuang, S. J., Pan, H. B., Peng, N. J., Cheng, J. T., et al. (2014). Ubiquitin-conjugating enzyme UBE2C is highly expressed in breast microcalcification lesions. *PLoS ONE* 9, e93934. doi: 10.1371/journal.pone.0093934

Cornish, A. J., David, A., and Sternberg, M. J. E. (2018). PhenoRank: reducing study bias in gene prioritization through simulation. *Bioinformatics* 34, 2087–2095. doi: 10.1093/bioinformatics/bty028

Costa, A., Rossi, E., Scicchitano, B. M., Coletti, D., Moresi, V., Adamo, S., et al. (2014). Neurohypophyseal hormones: novel actors of striated muscle development and homeostasis. *Eur. J. Transl. Myol*. 24, 3790. doi: 10.4081/bam.2014.3.217

Cowen, L., Ideker, T., Raphael, B. J., and Sharan, R. (2017). Network propagation: a universal amplifier of genetic associations. *Nat. Rev. Genet*. 18, 551–562. doi: 10.1038/nrg.2017.38

D'Amelio, M., Cavallucci, V., Middei, S., Marchetti, C., Pacioni, S., Ferri, A., et al. (2011). Caspase-3 triggers early synaptic dysfunction in a mouse model of Alzheimer's disease. *Nat. Neurosci*. 14, 69–76. doi: 10.1038/nn.2709

Desouza, M., Gunning, P. W., and Stehn, J. R. (2012). The actin cytoskeleton as a sensor and mediator of apoptosis. *Bioarchitecture* 2, 75–87. doi: 10.4161/bioa.20975

Di Leo, A., Desmedt, C., Bartlett, J. M., Piette, F., Ejlertsen, B., Pritchard, K. I., et al. (2011). HER2 and TOP2A as predictive markers for anthracycline-containing chemotherapy regimens as adjuvant treatment of breast cancer: a meta-analysis of individual patient data. *Lancet Oncol*. 12, 1134–1142. doi: 10.1016/S1470-2045(11)70231-5

do Valle, I. F., Menichetti, G., Simonetti, G., Bruno, S., Zironi, I., Durso, D. F., et al. (2018). Network integration of multi-tumour omics data suggests novel targeting strategies. *Nat. Commun*. 9, 4514. doi: 10.1038/s41467-018-06992-7

Dobin, A., Davis, C. A., Schlesinger, F., Drenkow, J., Zaleski, C., Jha, S., et al. (2013). STAR: ultrafast universal RNA-seq aligner. *Bioinformatics* 29, 15–21. doi: 10.1093/bioinformatics/bts635

Doll, S., Dressen, M., Geyer, P. E., Itzhak, D. N., Braun, C., Doppler, S. A., et al. (2017). Region and cell-type resolved quantitative proteomic map of the human heart. *Nat. Commun*. 8, 1469. doi: 10.1038/s41467-017-01747-2

ElShal, S., Tranchevent, L. C., Sifrim, A., Ardeshirdavani, A., Davis, J., and Moreau, Y. (2016). Beegle: from literature mining to disease-gene discovery. *Nucleic Acids Res*. 44, e18. doi: 10.1093/nar/gkv905

Eltohamy, M. I., Badawy, O. M., El kinaai, N., Loay, I., Nassar, H. R., Allam, R. M., et al. (2018). Topoisomerase II alpha gene alteration in triple negative breast cancer and its predictive role for anthracycline-based chemotherapy (Egyptian NCI Patients). *Asian Pac. J. Cancer Prev*. 19, 3581–3589. doi: 10.31557/APJCP.2018.19.12.3581

Erickson, S. L., O'Shea, K. S., Ghaboosi, N., Loverro, L., Frantz, G., Bauer, M., et al. (1997). ErbB3 is required for normal cerebellar and cardiac development: a comparison with ErbB2-and heregulin-deficient mice. *Development* 124, 4999–5011. doi: 10.1242/dev.124.24.4999

Erten, S., Bebek, G., Ewing, R. M., and Koyuturk, M. (2011). DADA: degree-aware algorithms for network-based disease gene prioritization. *BioData Min*. 4, 19. doi: 10.1186/1756-0381-4-19

Eschenhagen, T., Bolli, R., Braun, T., Field, L. J., Fleischmann, B. K., Frisen, J., et al. (2017). Cardiomyocyte regeneration: a consensus statement. *Circulation* 136, 680–686. doi: 10.1161/CIRCULATIONAHA.117.029343

Espinoza, M. (2012). *On Network Randomization Methods: A Negative Control Study*. Fairfield, CT: Fairfield University.

Gene Ontology, C., Blake, J. A., Dolan, M., Drabkin, H., Hill, D. P., Li, N., et al. (2013). Gene Ontology annotations and resources. *Nucleic Acids Res*. 41, D530–535. doi: 10.1093/nar/gks1050

Ghiassian, S. D., Menche, J., and Barabasi, A. L. A. (2015). DIseAse MOdule Detection (DIAMOnD) algorithm derived from a systematic analysis of connectivity patterns of disease proteins in the human interactome. *PLoS Comput. Biol*. 11, e1004120. doi: 10.1371/journal.pcbi.1004120

Gotoh, N. (2008). Regulation of growth factor signaling by FRS2 family docking/scaffold adaptor proteins. *Cancer Sci*. 99, 1319–1325. doi: 10.1111/j.1349-7006.2008.00840.x

Gottlieb, A., Magger, O., Berman, I., Ruppin, E., and Sharan, R. (2011). PRINCIPLE: a tool for associating genes with diseases *via* network propagation. *Bioinformatics* 27, 3325–3326. doi: 10.1093/bioinformatics/btr584

Grover, A., and Leskovec, J. (2016). "node2vec: scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Francisco, CA), 855–864. doi: 10.1145/2939672.2939754

Guala, D., Sjolund, E., and Sonnhammer, E. L. (2014). MaxLink: network-based prioritization of genes tightly linked to a disease seed set. *Bioinformatics* 30, 2689–2690. doi: 10.1093/bioinformatics/btu344

Guala, D., and Sonnhammer, E. L. L. (2017). A large-scale benchmark of gene prioritization methods. *Sci. Rep*. 7, 46598. doi: 10.1038/srep46598

Guney, E., and Oliva, B. (2012). Exploiting protein-protein interaction networks for genome-wide disease-gene prioritization. *PLoS ONE* 7, e43557. doi: 10.1371/journal.pone.0043557

Guo, X., Waddell, D. S., Wang, W., Wang, Z., Liberati, N. T., Yong, S., et al. (2008). Ligand-dependent ubiquitination of Smad3 is regulated by casein kinase 1 gamma 2, an inhibitor of TGF-beta signaling. *Oncogene* 27, 7235–7247. doi: 10.1038/onc.2008.337

Gyanchandani, R., Ortega Alves, M. V., Myers, J. N., and Kim, S. (2013). A proangiogenic signature is revealed in FGF-mediated bevacizumab-resistant head and neck squamous cell carcinoma. *Mol. Cancer Res*. 11, 1585–1596. doi: 10.1158/1541-7786.MCR-13-0358

Gyorffy, B., Lanczky, A., Eklund, A. C., Denkert, C., Budczies, J., Li, Q., et al. (2010). An online survival analysis tool to rapidly assess the effect of 22,277 genes on breast cancer prognosis using microarray data of 1,809 patients. *Breast Cancer Res. Treat*. 123, 725–731. doi: 10.1007/s10549-009-0674-9

Hale, P. J., Lopez-Yunez, A. M., and Chen, J. Y. (2012). Genome-wide meta-analysis of genetic susceptible genes for Type 2 Diabetes. *BMC Syst. Biol*. 6(Suppl.3), S16. doi: 10.1186/1752-0509-6-S3-S16

Hercus, T. R., Broughton, S. E., Ekert, P. G., Ramshaw, H. S., Perugini, M., Grimbaldeston, M., et al. (2012). The GM-CSF receptor family: mechanism of activation and implications for disease. *Growth Fact*. 30, 63–75. doi: 10.3109/08977194.2011.649919

Huang, de. W., Sherman, B. T., and Lempicki, R. A. (2009). Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucleic Acids Res*. 37, 1–13. doi: 10.1093/nar/gkn923

Huang, H., Li, J., and Chen, J. Y. (2009). "Disease gene-fishing in molecular interaction networks: a case study in colorectal cancer," in *Conference proceedings : Annual International Conference of the IEEE Engineering in Medicine and Biology Society IEEE Engineering in Medicine and Biology Society Conference* (Minneapolis, MN), 6416–6419.

Jiang, X., Lopez, A., Holyoake, T., Eaves, A., and Eaves, C. (1999). Autocrine production and action of IL-3 and granulocyte colony-stimulating factor in chronic myeloid leukemia. *Proc. Natl. Acad. Sci. U. S. A*. 96, 12804–12809. doi: 10.1073/pnas.96.22.12804

Johnston, S. R. (2006). Targeting downstream effectors of epidermal growth factor receptor/HER2 in breast cancer with either farnesyltransferase inhibitors or mTOR antagonists. *Int. J. Gynecol. Cancer* 16(Suppl.2), 543–548. doi: 10.1111/j.1525-1438.2006.00692.x

Jonsson, T., Atwal, J. K., Steinberg, S., Snaedal, J., Jonsson, P. V., Bjornsson, S., et al. (2012). A mutation in APP protects against Alzheimer's disease and age-related cognitive decline. *Nature* 488, 96–99. doi: 10.1038/nature11283

Kanehisa, M., Furumichi, M., Tanabe, M., Sato, Y., and Morishima, K. (2017). KEGG new perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Res*. 45, D353–D361. doi: 10.1093/nar/gkw1092

Kanehisa, M., Goto, S., Furumichi, M., Tanabe, M., and Hirakawa, M. (2010). KEGG for representation and analysis of molecular networks involving diseases and drugs. *Nucleic Acids Res*. 38, D355–360. doi: 10.1093/nar/gkp896

KEGG (2022). *Chronic Myeloid Leukemia - Homo Sapiens (Human)*. Kyoto: Human Genome Center, Institute of Medical Science; University of Tokyo, Bioinformatics Center; Institute for Chemical Research, Kyoto University.

Khanin, R., and Wit, E. (2006). How scale-free are biological networks. *J. Comput. Biol*. 13, 810–818. doi: 10.1089/cmb.2006.13.810

Kim, J., and Bang, H. (2016). Three common misuses of *P*-values. *Dent. Hypotheses* 7, 73–80. doi: 10.4103/2155-8213.190481

Kim, Y. J., Lee, G., Han, J., Song, K., Choi, J. S., Choi, Y. L., et al. (2019). UBE2C overexpression aggravates patient outcome by promoting estrogen-dependent/independent cell proliferation in early hormone receptor-positive and HER2-negative breast cancer. *Front. Oncol*. 9, 1574. doi: 10.3389/fonc.2019.01574

Kobayashi, C. I., Takubo, K., Kobayashi, H., Nakamura-Ishizu, A., Honda, H., Kataoka, K., et al. (2014). The IL-2/CD25 axis maintains distinct subsets of chronic myeloid leukemia-initiating cells. *Blood* 123, 2540–2549. doi: 10.1182/blood-2013-07-517847

Koschmann, J., Bhar, A., Stegmaier, P., Kel, A. E., and Wingender, E. (2015). "Upstream analysis": an integrated promoter-pathway analysis approach to causal interpretation of microarray data. *Microarrays* 4, 270–286. doi: 10.3390/microarrays4020270

Krallinger, M., Valencia, A., and Hirschman, L. (2008). Linking genes to literature: text mining, information extraction, and retrieval applications for biology. *Genome Biol*. 9(Suppl.2), S8. doi: 10.1186/gb-2008-9-s2-s8

Kramer, A., Green, J., and Pollard, J. Jr., and Tugendreich, S. (2014). Causal analysis approaches in Ingenuity Pathway Analysis. *Bioinformatics* 30, 523–530. doi: 10.1093/bioinformatics/btt703

Krueger, F. (2015). *Trim galore. A wrapper tool around Cutadapt and FastQC to Consistently Apply Quality and Adapter Trimming to FastQ Files*, 516.

Kubin, T., Poling, J., Kostin, S., Gajawada, P., Hein, S., Rees, W., et al. (2011). Oncostatin M is a major mediator of cardiomyocyte dedifferentiation and remodeling. *Cell Stem Cell*. 9, 420–432. doi: 10.1016/j.stem.2011.08.013

La Bella, V., Liguori, M., Cittadella, R., Settipani, N., Piccoli, T., Manna, I., et al. (2004). A novel mutation (Thr116Ile) in the presenilin 1 gene in a patient with early-onset Alzheimer's disease. *Eur. J. Neurol*. 11, 521–524. doi: 10.1111/j.1468-1331.2004.00828.x

Lam, N. T., and Sadek, H. A. (2018). Neonatal heart regeneration: comprehensive literature review. *Circulation* 138, 412–423. doi: 10.1161/CIRCULATIONAHA.118.033648

Lanczky, A., Nagy, A., Bottai, G., Munkacsy, G., Szabo, A., Santarpia, L., et al. (2016). miRpower: a web-tool to validate survival-associated miRNAs utilizing expression data from 2178 breast cancer patients. *Breast Cancer Res. Treat*. 160, 439–446. doi: 10.1007/s10549-016-4013-7

Li, J., Zhu, X., and Chen, J. Y. (2009). Building disease-specific drug-protein connectivity maps from molecular interaction networks and PubMed abstracts. *PLoS Comput. Biol*. 5, e1000450. doi: 10.1371/journal.pcbi.1000450

Li, R., and Campos, J. (2015). Iida J: a gene regulatory program in human breast cancer. *Genetics* 201, 1341–1348. doi: 10.1534/genetics.115.180125

Liberzon, A., Birger, C., Thorvaldsdottir, H., Ghandi, M., Mesirov, J. P., Tamayo, P., et al. (2015). The Molecular Signatures Database (MSigDB) hallmark gene set collection. *Cell Syst*. 1, 417–425. doi: 10.1016/j.cels.2015.12.004

Liu, Y., Liang, Y., and Wishart, D. (2015). PolySearch2: a significantly improved text-mining system for discovering associations between human diseases, genes, drugs, metabolites, toxins and more. *Nucleic Acids Res*. 43, W535–542. doi: 10.1093/nar/gkv383

Liu, Y., Zhu, Y. H., Mao, C. Q., Dou, S., Shen, S., Tan, Z. B., et al. (2014). Triple negative breast cancer therapy with CDK1 siRNA delivered by cationic lipid assisted PEG-PLA nanoparticles. *J. Control Release*. 192, 114–121. doi: 10.1016/j.jconrel.2014.07.001

Love, M. I., Huber, W., and Anders, S. (2014). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol*. 15, 550. doi: 10.1186/s13059-014-0550-8

Lu, Y., Yang, G., Xiao, Y., Zhang, T., Su, F., Chang, R., et al. (2020). Upregulated cyclins may be novel genes for triple-negative breast cancer based on bioinformatic analysis. *Breast Cancer*. 27, 903–911. doi: 10.1007/s12282-020-01086-z

Martin-Orozco, E., Sanchez-Fernandez, A., Ortiz-Parra, I., and Ayala-San Nicolas, M. (2019). WNT: signaling in tumors: the way to evade drugs and immunity. *Front. Immunol*. 10, 2854. doi: 10.3389/fimmu.2019.02854

Moreau, Y., and Tranchevent, L. C. (2012). Computational tools for prioritizing candidate genes: boosting disease gene discovery. *Nat. Rev. Genet*. 13, 523–536. doi: 10.1038/nrg3253

Morry, J., Ngamcherdtrakul, W., Gu, S., Reda, M., Castro, D. J., Sangvanich, T., et al. (2017). Targeted treatment of metastatic breast cancer by PLK1 siRNA delivered by an antioxidant nanoparticle platform. *Mol. Cancer Ther*. 16, 763–772. doi: 10.1158/1535-7163.MCT-16-0644

Muhammad, S. A., Raza, W., Nguyen, T., Bai, B., Wu, X., Chen, J., et al. (2017). Cellular signaling pathways in insulin resistance-systems biology analyses of microarray dataset reveals new drug target gene signatures of type 2 diabetes mellitus. *Front. Physiol*. 8, 13. doi: 10.3389/fphys.2017.00013

Mustonen, E., Ruskoaho, H., and Rysa, J. (2013). Thrombospondins, potential drug targets for cardiovascular diseases. *Basic Clin. Pharmacol. Toxicol*. 112, 4–12. doi: 10.1111/bcpt.12026

Nakada, Y., Zhou, Y., Gong, W., Zhang, E., Skie, E., Nguyen, T., et al. (2021). Single nucleus transcriptomics: apical resection in newborn pigs extends the time-window of cardiomyocyte proliferation and myocardial regeneration. *Circulation* 121, 56995. doi: 10.1161/CIRCULATIONAHA.121.056995

Nedeljkovic, M., and Damjanovic, A. (2019). Mechanisms of chemotherapy resistance in triple-negative breast cancer-how we can rise to the challenge. *Cells* 8, 90957. doi: 10.3390/cells8090957

Neupane, M., and Kiser, J. N. (2018). Bovine respiratory disease complex coordinated agricultural project research T, Neibergs HL: gene set enrichment analysis of SNP data in dairy and beef cattle with bovine respiratory disease. *Anim. Genet*. 49, 527–538. doi: 10.1111/age.12718

Newman, M. E. (2005). A measure of betweenness centrality based on random walks. *Social Netw*. 27, 39–54. doi: 10.1016/j.socnet.2004.11.009

Newman, M. E. (2006). Modularity and community structure in networks. *Proc. Natl. Acad. Sci. U. S. A*. 103, 8577–8582. doi: 10.1073/pnas.0601602103

Newman, M. E. J. (2008). "Mathematics of networks," in *The New Palgrave Encyclopedia of Economics, 2 Edn*, eds L. E. Blume, S. N. Durlauf. London: Palgrave Macmillan UK.

Nguyen, T., Wei, Y., Nakada, Y., Zhou, Y., and Zhang, J. (2022). Cardiomyocyte cell-cycle regulation in neonatal large mammals: single nucleus RNA-sequencing

data analysis *via* an artificial-intelligence-based pipeline. *Front. Bioeng. Biotechnol.* 10, 914450. doi: 10.3389/fbioe.2022.914450

Nitsch, D., Tranchevent, L. C., Goncalves, J. P., Vogt, J. K., Madeira, S. C., Moreau, Y., et al. (2011). PINTA: a web server for network-based gene prioritization from expression data. *Nucleic Acids Res.* 39, W334–338. doi: 10.1093/nar/gkr289

Ogata, H., Goto, S., Sato, K., Fujibuchi, W., Bono, H., Kanehisa, M., et al. (1999). KEGG: kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.* 27, 29–34. doi: 10.1093/nar/27.1.29

Olsen, C., Fleming, K., Prendergast, N., Rubio, R., Emmert-Streib, F., Bontempi, G., et al. (2014). Inference and validation of predictive gene networks from biomedical literature and gene expression data. *Genomics* 103, 329–336. doi: 10.1016/j.ygeno.2014.03.004

Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). *The PageRank Citation Ranking: Bringing Order to the Web*. Stanford, CA: Stanford InfoLab.

Patel, J. R., Thangavelu, P., Terrell, R. M., Israel, B., Sarkar, A. B., Davidson, A. M., et al. (2022). Novel allosteric inhibitor targets PLK1 in triple negative breast cancer cells. *Biomolecules* 12, 40531. doi: 10.3390/biom12040531

Peng, J., Guan, J., and Shang, X. (2019). Predicting Parkinson's disease genes based on node2vec and autoencoder. *Front. Genet.* 10, 226. doi: 10.3389/fgene.2019.00226

Perez-Iratxeta, C., Bork, P., and Andrade, M. A. (2002). Association of genes to genetically inherited diseases using data mining. *Nat. Genet.* 31, 316–319. doi: 10.1038/ng895

Peters, L. A., Perrigoue, J., Mortha, A., Iuga, A., Song, W. M., Neiman, E. M., et al. (2017). A functional genomics predictive network model identifies regulators of inflammatory bowel disease. *Nat. Genet.* 49, 1437–1449. doi: 10.1038/ng.3947

Porrello, E. R., Mahmoud, A. I., Simpson, E., Hill, J. A., Richardson, J. A., Olson, E. N., et al. (2011). Transient regenerative potential of the neonatal mouse heart. *Science* 331, 1078–1080. doi: 10.1126/science.1200708

Porrello, E. R., and Olson, E. N. (2014). A neonatal blueprint for cardiac regeneration. *Stem Cell Res.* 13, 556–570. doi: 10.1016/j.scr.2014.06.003

Puente, B. N., Kimura, W., Muralidhar, S. A., Moon, J., Amatruda, J. F., Phelps, K. L., et al. (2014). The oxygen-rich postnatal environment induces cardiomyocyte cell-cycle arrest through DNA damage response. *Cell* 157, 565–579. doi: 10.1016/j.cell.2014.03.032

Radulescu, E., Jaffe, A. E., Straub, R. E., Chen, Q., Shin, J. H., Hyde, T. M., et al. (2018). Identification and prioritization of gene sets associated with schizophrenia risk by co-expression network analysis in human brain. *Mol. Psychiatry* 2018, 286559. doi: 10.1101/286559

Rajab, A., Straub, V., McCann, L. J., Seelow, D., Varon, R., Barresi, R., et al. (2010). Fatal cardiac arrhythmia and long-QT syndrome in a new form of congenital generalized lipodystrophy with muscle rippling (CGL4) due to PTRF-CAVIN mutations. *PLoS Genet.* 6, e1000874. doi: 10.1371/journal.pgen.1000874

Rao, A. R., Jana, R., and Bandyopadhyay, S. A. (1996). Markov chain Monte Carlo method for generating random (0, 1)-matrices with given marginals. *Sankhyā* 1996, 225–242.

Rolland, T., Tasan, M., Charloteaux, B., Pevzner, S. J., Zhong, Q., Sahni, N., et al. (2014). A proteome-scale map of the human interactome network. *Cell* 159, 1212–1226. doi: 10.1016/j.cell.2014.10.050

Saha, S., Harrison, S. H., and Chen, J. Y. (2008). Dissecting the human plasma proteome and inflammatory response biomarkers. *Proteomics* 2008, 507. doi: 10.1002/pmic.200800507

Sayers, E. (2008). *E-utilities Quick Start.* Bethesda, MD: Entrez Programming Utilities Help.

Schlotterer, C., Tobler, R., Kofler, R., and Nolte, V. (2014). Sequencing pools of individuals - mining genome-wide polymorphism data without big funding. *Nat. Rev. Genet.* 15, 749–763. doi: 10.1038/nrg3803

Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., et al. (2003). Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res.* 13, 2498–2504. doi: 10.1101/gr.1239303

Singh, A. R., Sivadas, A., Sabharwal, A., Vellarikal, S. K., Jayarajan, R., Verma, A., et al. (2016). Chamber specific gene expression landscape of the zebrafish heart. *PLoS ONE* 11, e0147823. doi: 10.1371/journal.pone.0147823

Singh-Blom, U. M., Natarajan, N., Tewari, A., Woods, J. O., Dhillon, I. S., Marcotte, E. M., et al. (2013). Prediction and validation of gene-disease associations using methods inspired by social network analyses. *PLoS ONE* 8, e58977. doi: 10.1371/journal.pone.0058977

Smedley, D., Kohler, S., Czeschik, J. C., Amberger, J., Bocchini, C., Hamosh, A., et al. (2014). Walking the interactome for candidate prioritization in exome sequencing studies of Mendelian diseases. *Bioinformatics* 30, 3215–3222. doi: 10.1093/bioinformatics/btu508

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

Sun, J., and Zhao, Z. (2010). A comparative study of cancer proteins in the human protein-protein interaction network. *BMC Genomics* 11(Suppl.3), S5. doi: 10.1186/1471-2164-11-S3-S5

Szklarczyk, D., Franceschini, A., Wyder, S., Forslund, K., Heller, D., Huerta-Cepas, J., et al. (2015). STRING v10: protein-protein interaction networks, integrated over the tree of life. *Nucleic Acids Res.* 43, D447–452. doi: 10.1093/nar/gku1003

Szklarczyk, D., Morris, J. H., Cook, H., Kuhn, M., Wyder, S., Simonovic, M., et al. (2017). The STRING database in 2017: quality-controlled protein-protein association networks, made broadly accessible. *Nucleic Acids Res.* 45, D362–D368. doi: 10.1093/nar/gkw937

Talesa, V. N. (2001). Acetylcholinesterase in Alzheimer's disease. *Mech. Ageing Dev.* 122, 1961–1969. doi: 10.1016/S0047-6374(01)00309-8

Tessier, L., Cote, O., Clark, M. E., Viel, L., Diaz-Mendez, A., Anders, S., et al. (2018). Gene set enrichment analysis of the bronchial epithelium implicates contribution of cell cycle and tissue repair processes in equine asthma. *Sci. Rep.* 8, 16408. doi: 10.1038/s41598-018-34636-9

Timar, G., Dorogovtsev, S. N., and Mendes, J. F. (2016). Scale-free networks with exponent one. *Phys. Rev. E.* 94, 022302. doi: 10.1103/PhysRevE.94.022302

Tiong, K. L., and Yeang, C. H. (2019). MGSEA - a multivariate Gene set enrichment analysis. *BMC Bioinformatics* 20, 145. doi: 10.1186/s12859-019-2716-6

Tyner, C., Barber, G. P., Casper, J., Clawson, H., Diekhans, M., Eisenhart, C., et al. (2017). The UCSC Genome Browser database: 2017 update. *Nucleic Acids Res.* 45, D626–D634. doi: 10.1093/nar/gkw1134

Ueda, A., Oikawa, K., Fujita, K., Ishikawa, A., Sato, E., Ishikawa, T., et al. (2019). Therapeutic potential of PLK1 inhibition in triple-negative breast cancer. *Lab. Invest.* 99, 1275–1286. doi: 10.1038/s41374-019-0247-4

UniProt Consortium, T. (2018). UniProt: the universal protein knowledgebase. *Nucleic Acids Res.* 46, 2699. doi: 10.1093/nar/gky092

Valent, P. (2014). Targeting the JAK2-STAT5 pathway in CML. *Blood* 124, 1386–1388. doi: 10.1182/blood-2014-07-585943

van Dam, S., Cordeiro, R., Craig, T., van Dam, J., Wood, S. H., de Magalhaes, J. P., et al. (2012). GeneFriends: an online co-expression analysis tool to identify novel gene targets for aging and complex diseases. *BMC Genomics* 13, 535. doi: 10.1186/1471-2164-13-535

Van Vooren, S., Thienpont, B., Menten, B., Speleman, F., De Moor, B., Vermeesch, J., et al. (2007). Mapping biomedical concepts onto the human genome by mining literature on chromosomal aberrations. *Nucleic Acids Res.* 35, 2533–2543. doi: 10.1093/nar/gkm054

Waksman, A. (1968). A permutation network. *J. ACM* 15, 159–163. doi: 10.1145/321439.321449

Wang, C., Li, Y., Li, H., Zhang, Y., Ying, Z., Wang, X., et al. (2020). Disruption of FGF signaling ameliorates inflammatory response in hepatic stellate cells. *Front. Cell Dev. Biol.* 8, 601. doi: 10.3389/fcell.2020.00601

Wang, J., Karra, R., Dickson, A. L., and Poss, K. D. (2013). Fibronectin is deposited by injury-activated epicardial cells and is necessary for zebrafish heart regeneration. *Dev. Biol.* 382, 427–435. doi: 10.1016/j.ydbio.2013.08.012

Wang, S. L., Li, X. L., and Fang, J. (2012). Finding minimum gene subsets with heuristic breadth-first search algorithm for robust tumor classification. *BMC Bioinformatics* 13, 178. doi: 10.1186/1471-2105-13-178

Wang, Z., Duenas-Osorio, L., and Padgett, J. E. (2015). A new mutually reinforcing network node and link ranking algorithm. *Sci. Rep.* 5, 15141. doi: 10.1038/srep15141

Wei, W., Norton, D. D., Wang, X., and Kusiak, J. W. (2002). Abeta 17-42 in Alzheimer's disease activates JNK and caspase-8 leading to neuronal apoptosis. *Brain* 125, 2036–2043. doi: 10.1093/brain/awf205

Winter, C., Kristiansen, G., Kersting, S., Roy, J., Aust, D., Knosel, T., et al. (2012). Google goes cancer: improving outcome prediction for cancer patients by network-based ranking of marker genes. *PLoS Comput. Biol.* 8, e1002511. doi: 10.1371/journal.pcbi.1002511

Wu, X., Chen, J. Y., Alterovitz, G., Benson, R., and Ramoni, M. (2009). Molecular interaction networks: topological and functional characterizations. *Automat. Proteom. Genom.* 145, 6. doi: 10.1002/9780470741191.ch6

Xia, Q., Cai, Y., Peng, R., Wu, G., Shi, Y., Jiang, W., et al. (2014). The CDK1 inhibitor RO3306 improves the response of BRCA-proficient breast cancer cells to PARP inhibition. *Int. J. Oncol.* 44, 735–744. doi: 10.3892/ijo.2013.2240

Xie, B., Agam, G., Balasubramanian, S., Xu, J., Gilliam, T. C., Maltsev, N., et al. (2015). Disease gene prioritization using network and feature. *J. Comput. Biol.* 22, 313–323. doi: 10.1089/cmb.2015.0001

Yang, C., Jin, C., Li, X., Wang, F., McKeehan, W. L., Luo, Y., et al. (2012). Differential specificity of endocrine FGF19 and FGF21 to FGFR1 and FGFR4 in complex with KLB. *PLoS ONE* 7, e33870. doi: 10.1371/journal.pone.0033870

Ye, L., D'Agostino, G., Loo, S. J., Wang, C. X., Su, L. P., Tan, S. H., et al. (2018). Early regenerative capacity in the porcine heart. *Circulation* 138, 2798–2808. doi: 10.1161/CIRCULATIONAHA.117.031542

Yin, T., Chen, S., Wu, X., and Tian, W. (2017). GenePANDA-a novel network-based gene prioritizing tool for complex diseases. *Sci. Rep.* 7, 43258. doi: 10.1038/srep43258

Yu, L., Fernandez, S., and Brock, G. (2017). Power analysis for RNA-Seq differential expression studies. *BMC Bioinformatics* 18, 234. doi: 10.1186/s12859-017-1648-2

Yu, W., Wulf, A., Liu, T., Khoury, M. J., and Gwinn, M. (2008). Gene Prospector: an evidence gateway for evaluating potential susceptibility genes and interacting risk factors for human diseases. *BMC Bioinformatics* 9, 528. doi: 10.1186/1471-2105-9-528

Yue, Z., Arora, I., Zhang, E. Y., Laufer, V., Bridges, S. L., Chen, J. Y., et al. (2017). Repositioning drugs by targeting network modules: a Parkinson's disease case study. *BMC Bioinformatics* 18, 532. doi: 10.1186/s12859-017-1889-0

Yue, Z., Kshirsagar, M. M., Nguyen, T., Suphavilai, C., Neylon, M. T., Zhu, L., et al. (2015). PAGER: constructing PAGs and new PAG-PAG relationships for network biology. *Bioinformatics* 31, i250–257. doi: 10.1093/bioinformatics/btv265

Yue, Z., Zheng, Q., Neylon, M. T., Yoo, M., Shin, J., Zhao, Z., et al. (2018). 2.0: an update to the pathway, annotated-list and gene-signature electronic repository for Human Network Biology. *Nucleic Acids Res.* 46, D668–D676. doi: 10.1093/nar/gkx1040

Zhang, E., Nguyen, T., Zhao, M., Dang, S. D. H., Chen, J. Y., Bian, W., et al. (2020). Identifying the key regulators that promote cell-cycle activity in the hearts of early neonatal pigs after myocardial injury. *PLoS ONE* 15, e0232963. doi: 10.1371/journal.pone.0232963

Zhang, F., and Chen, J. Y. (2010). Discovery of pathway biomarkers from coupled proteomics and systems biology methods. *BMC Genomics* 11(Suppl.2), S12. doi: 10.1186/1471-2164-11-S2-S12

Zhang, F., and Chen, J. Y. (2013). Breast cancer subtyping from plasma proteins. *BMC Medical Genom.* 6(Suppl.1), S6. doi: 10.1186/1755-8794-6-S1-S6

Zhang, H., Ferguson, A., Robertson, G., Jiang, M., Zhang, T., Sudlow, C., et al. (2021). Benchmarking network-based gene prioritization methods for cerebral small vessel disease. *Brief Bioinform.* 22, bbab006. doi: 10.1093/bib/bbab006

Zhao, M., Zhang, E., Wei, Y., Zhou, Y., Walcott, G. P., Zhang, J., et al. (2020). Apical resection prolongs the cell cycle activity and promotes myocardial regeneration after left ventricular injury in neonatal pig. *Circulation* 142, 913–916. doi: 10.1161/CIRCULATIONAHA.119.044619

Zhao, S., Geng, Y., Cao, L., Yang, Q., Pan, T., Zhou, D., et al. (2021). Deciphering the performance of polo-like kinase 1 in triple-negative breast cancer progression according to the centromere protein U-phosphorylation pathway. *Am. J. Cancer Res.* 11, 2142–2158.

Zhao, Z. Q., Han, G. S., Yu, Z. G., and Li, J. (2015). Laplacian: normalization and random walk on heterogeneous networks for disease-gene prioritization. *Comput. Biol. Chem.* 57, 21–28. doi: 10.1016/j.compbiolchem.2015.02.008

Zhu, W., Zhang, E., Zhao, M., Chong, Z., Fan, C., Tang, Y., et al. (2018). Regenerative potential of neonatal porcine hearts. *Circulation* 138, 2809–2816. doi: 10.1161/CIRCULATIONAHA.118.034886

frontiers | Frontiers in Artificial Intelligence

TYPE Review
PUBLISHED 08 November 2022
DOI 10.3389/frai.2022.1034631

OPEN ACCESS

EDITED BY
Ramin Homayouni,
Oakland University William Beaumont School of Medicine, United States

REVIEWED BY
Zhining Wen,
Sichuan University, China
Tadahaya Mizuno,
The University of Tokyo, Japan

*CORRESPONDENCE
Weida Tong
weida.tong@fda.hhs.gov

†These authors have contributed equally to this work

SPECIALTY SECTION
This article was submitted to Medicine and Public Health, a section of the journal Frontiers in Artificial Intelligence

RECEIVED 01 September 2022
ACCEPTED 17 October 2022
PUBLISHED 08 November 2022

CITATION
Connor S, Li T, Roberts R, Thakkar S, Liu Z and Tong W (2022) Adaptability of AI for safety evaluation in regulatory science: A case study of drug-induced liver injury.
Front. Artif. Intell. 5:1034631.
doi: 10.3389/frai.2022.1034631

Adaptability of AI for safety evaluation in regulatory science: A case study of drug-induced liver injury

Skylar Connor[1†], Ting Li[1†], Ruth Roberts[2,3], Shraddha Thakkar[4], Zhichao Liu[1] and Weida Tong[1*]

[1]National Center for Toxicological Research, US Food and Drug Administration, Jefferson, AR, United States, [2]ApconiX Ltd., Macclesfield, United Kingdom, [3]Department of Biosciences, University of Birmingham, Birmingham, United Kingdom, [4]Center for Drug Evaluation and Research, US Food and Drug Administration, Silver Spring, MD, United States

Artificial intelligence (AI) has played a crucial role in advancing biomedical sciences but has yet to have the impact it merits in regulatory science. As the field advances, *in silico* and *in vitro* approaches have been evaluated as alternatives to animal studies, in a drive to identify and mitigate safety concerns earlier in the drug development process. Although many AI tools are available, their acceptance in regulatory decision-making for drug efficacy and safety evaluation is still a challenge. It is a common perception that an AI model improves with more data, but does reality reflect this perception in drug safety assessments? Importantly, a model aiming at regulatory application needs to take a broad range of model characteristics into consideration. Among them is adaptability, defined as the adaptive behavior of a model as it is retrained on unseen data. This is an important model characteristic which should be considered in regulatory applications. In this study, we set up a comprehensive study to assess adaptability in AI by mimicking the real-world scenario of the annual addition of new drugs to the market, using a model we previously developed known as DeepDILI for predicting drug-induced liver injury (DILI) with a novel Deep Learning method. We found that the target test set plays a major role in assessing the adaptive behavior of our model. Our findings also indicated that adding more drugs to the training set does not significantly affect the predictive performance of our adaptive model. We concluded that the proposed adaptability assessment framework has utility in the evaluation of the performance of a model over time.

KEYWORDS

adaptability, AI, deep learning, drug-induced liver injury (DILI), drug safety, risk assessment, regulatory science

AI in regulatory sciences

The term Artificial Intelligence (AI) refers to the ability of a computer system to learn from past data to predict future outcomes. Machine Learning (ML), a subset of AI, refers to the study and use of computer algorithms that automatically improve in making predictions or decisions based on their experiences and interactions with the training data (Gupta et al., 2021). Deep Learning (DL), a subset of ML, mimics the cognitive behaviors associated with the approach the human brain would take in learning and problem-solving of data-intensive problems (Gupta et al., 2021). Although AI has gained momentum in recent advancements within the biomedical field, especially in areas like drug safety evaluation and assessment, from a regulatory science perspective AI has yet to have the impact it merits.

Regulatory science is the science of developing new tools, standards, and approaches to assess the safety, efficacy, quality, and performance of FDA-regulated products (FDA, 2021). The main role of regulatory science is to certify the safety, proper labeling, and efficacy of food, drug and cosmetic items, like mandating food standards for packaging and quality, and regulating cosmetic products and medical devices (Patel and Miller, 2012). Despite being a critical component in the continued evolution of our approaches to certifying the safety and quality of food and medical products, regulatory science research has yet to have the impact it merits (Hamburg, 2011).

As the field of regulatory science advances, *in silico* and *in vitro* approaches have been extensively evaluated as alternatives to some animal studies, in a drive to identify and mitigate safety concerns earlier in the drug development process (Hamburg, 2011). AI and DL tools have begun to play a crucial role in the advancement of computer-aided drug discovery, design, and development (Gupta et al., 2021), specifically for the study of drug safety and efficacy. DL is arguably the most advanced ML approach that frequently outperforms conventional ML approaches (Slikker et al., 2012; Gupta et al., 2021; Anklam et al., 2022). DL usually consists of multiple layers of neural networks which can be constructed and connected in diverse ways, giving rise to a broad range of methodologies. As a result, DL has become the first-choice algorithm in regulatory science research due to its diversity and superior performance.

Regulatory frameworks and the initiatives benefiting from AI

As interest in the use of AI within scientific and clinical research has grown, the global government agencies such as the European Medicines Agency, the European Food Safety Agency, the Unites States National Institute of Standards and Technology (NIST), the US Food and Drug Administration (FDA), and the United States Congress have worked to strengthen the guidance on how to safely implement the use of AI as software tools and medical devices. In 2021, the US House of Representatives introduced the FDA Modernization Act, H.R. 2565 (Text-H.R.2565-117th Congress (2021–2022), 2021) and S.2952 (Text-S.2952-117th Congress (2021–2022), 2021), intended to reform the drug approval process and drive the use of non-animal testing methods. In June 2022, the FDA Modernization Act as was passed as an additional provision, Section 701 (Text-H.R.7667-117th Congress (2021–2022), 2022), in a larger legislative package of FDA-related reforms known as the Food and Drug Amendments of 2022, H.R. 7667 (Text-H.R.7667-117th Congress (2021–2022), 2022). NIST has released several whitepapers providing guidance on how to properly implement AI in regulatory sciences like the 116th Congress AI in Government Act of 2020 (Text-H.R.2575-116th Congress (2019–2020), 2020) and the 117th Congress GOOD AI Act of 2021 (Text-S.3035-117th Congress (2021–2022), 2022).

The FDA has made major strides in guiding developmental and more recently computational opportunities within regulatory science through programs like the Drug Development Tool Qualification Programs (U. S. Food and Drug Administration, 2021a) and the FDA's Predictive Toxicology Roadmap (U. S. Food and Drug Administration, 2017), as well as many initiatives at the Center for Drug Evaluation and Research (CEDR) and for the first time an AI/ML specific Action Plan named "Artificial Intelligence/Machine Learning (AI/ML)-Based Software as a Medical Device (SaMD) Action Plan" has been instituted by the Center for Devices and Radiological Health (CDRH) (U. S. Food and Drug Administration, 2019b, 2021b).

In 2016 the FDA passed the Cures Act which defined a three-stage qualification process that allowed the use of qualified Drug Development Tools (DDTs) across drug development programs (U. S. Food and Drug Administration, 2021a). DDTs are methods, materials, or measures that have the potential to facilitate drug development. There is a total of four DDT Qualification Programs (U. S. Food and Drug Administration, 2021a). A qualified DDT has been determined to have a trusted specific interpretation and application within drug development and regulatory review for the qualified context of use. Once qualified, DDTs are made publicly available and can generally be included in Investigational New Drug (IND), New Drug Application (NDA), or Biologics License Application (BLA) submissions without requiring the FDA to reconsider or reconfirm its suitability (U. S. Food and Drug Administration, 2017, 2021a,c). The four programs, Animal Model, Biomarker, Clinical Outcome Assessment (COA), and the newest addition the Innovative Science and Technology Approaches for New Drugs (ISTAND) Pilot Program, rely on a context of use

statement. The Context of use statement is one of the most important parts of the qualification process. The context of use should describe all elements that characterize the manner and purpose of use for the DDT being submitted (U. S. Food and Drug Administration, 2021a). Once qualified the context of use will define the boundaries that justify to others where they can use the qualified DDT. The ISTAND Pilot Program (U. S. Food and Drug Administration, 2021c) was developed to expand the current types of DDTs by encouraging the development and acceptance of DDTs that are outside of the scope of existing programs but are still novel approaches to drug development and acceptable for regulatory use. Once a new model is considered qualified by the FDA for a specific context of use, industry and other stakeholders may use it for the qualified purpose during product development without the need for FDA reviewers to re-review the underlying supporting data (U. S. Food and Drug Administration, 2017, 2021a,c).

In December of 2017 the FDA's Toxicology Working Group published the FDA's Predictive Toxicology Roadmap (U. S. Food and Drug Administration, 2017), a six-part framework outlining Agency priorities and engagement in predictive toxicology, and identifying current toxicology issues related to FDA-regulated products. The roadmap describes the FDA's current thoughts on practical ways to incorporate the development and evaluation of emerging toxicological methods and innovative technologies into the FDA regulatory review process. The six-part framework moves to enhance FDA engagement in the science of toxicology through the organization of a senior-level Toxicology Working Group that will help identify areas where research is needed, assist with efforts to reduce duplication and increase collaboration inside and outside the FDA through the encouragement of frequent communication and fostering collaborations across sectors and disciplines both nationally and internationally (U. S. Food and Drug Administration, 2021a).

Adaptability of AI in regulatory science

Although there are several interpretations of adaptability and adaptive AI in the field, within this article we define adaptability as the study of the adaptive behavior of a model as it is retrained on unseen data. An adaptive model is a model that has the ability to continuously learn and change as it is used, meaning as time goes on the same question will not yield the same results as the model learns to better address the problem. A locked model is trained, developed, and tested to produce the best version of the model and once the model is launched for public or private use it should produce the same results every time the same input is used.

The AI/ML specific action plan was a response to a discussion paper published by the FDA in April of 2019 with a request for stakeholder feedback on the potential approach to the premarket review of AI and ML driven software modifications for Software used as a Medical Device (SaMD) (U. S. Food and Drug Administration, 2019b, 2021b). SaMD (Health et al., 2018) is "software intended to be used for one or more medical purposes that perform these purposes without being part of a hardware medical device" as defined by the International Medical Device Regulators Forum (IMDRF) (U. S. Food and Drug Administration, 2019a). As stated in the proposed plan, the FDA has cleared or approved several AI/ML-based SaMDs, but to date, SaMDs have typically only included algorithms that are "locked" prior to the systems or software's launch to market. Any proposed algorithm changes to a "locked" algorithm will likely require an FDA premarket review, especially if those changes are beyond the original approved authorization (U. S. Food and Drug Administration, 2019b). However, some algorithms have the capability and need to adapt over time through continuous learning from real-world experience after distribution.

The advantage and drawback, depending on the circumstance, of a "locked" algorithm is the fact it will not continually adapt or learn from its post market use, this feature is important in some instances but occasionally an adaptive algorithm is needed. The newly released AI/ML-Based SaMD Action Plan outlines five actions that the FDA intends to take to advance the use of AI/ML based software within regulatory science. The first of which is tailored toward the further development of adaptive AI and ML algorithms within the regulatory framework through the "issuance of Draft Guidance on the Predetermined Change Control Plan" which includes SaMD Pre-Specifications (SPS), where manufacturers describe "what" aspects they intend or anticipate modifying through continuously learning, and Algorithm Change Protocol (ACP) which explains "how" the algorithm will learn and change while remaining safe and effective (U. S. Food and Drug Administration, 2021b). The four other actions include encouraging the development of good ML practices, fostering a patient-centered approach through incorporating transparency to users, supporting regulatory science efforts to evaluate and improve ML algorithms; and working with stakeholders who are piloting the Real-World Performance (RWP) process for AI/ML-based SaMD.

Programs like ISTAND and the AI/ML-based SaMD Action Plan help lay the foundation for methodologies and tools to advance the use of computation within regulatory science. To test the assumption that drug safety models improve as more data is added to the training set, we set up a comprehensive study to mimic the real-world scenario of annually adding novel drugs to the market, using a model we previously developed for assessing drug-induced liver injury (DILI), known as DeepDILI (Li et al., 2021). In using this approach, we addressed two important questions: First, did the model's performance improve

or decline as more data was added? Second, did the context of use change as the model adapted? Our evaluation followed the real-world scenario where a model was developed based on the drugs approved in the early years (before 1997) and assessed with the drugs approved thereafter (after 1997).

DeepDILI: A deep learning model to evaluate drug-induced liver injury in humans

Evaluating DILI has been a persistent challenge for the past 60 years and continues to be the leading cause of toxicity failures in pharmaceutical development (PoPPer et al., 1965; Zimmerman, 1999; Van Norman, 2019). In our previous study, we developed an AI drug safety model, known as DeepDILI (Li et al., 2021), a deep learning-powered prediction model designed to identify drugs with DILI potential in humans solely based on chemical structure information. DeepDILI was created by combining model-level representation generated from five conventional ML algorithms [k-nearest neighbor (kNN), logistic regression (LR), support vector machine (SVM), random forest (RF), and extreme gradient boosting (XGBoost)] with a deep learning framework using Mold2 (Hong et al., 2008) chemical descriptors. With DeepDILI, we aimed to evaluate whether the DILI potential of newly approved drugs could be predicted by accumulating knowledge from previously approved drugs. For that reason, the DeepDILI model was trained with 753 drugs released to the market prior to 1997 and evaluated on the 249 drugs approved in 1997 and thereafter. Upon evaluation the model yielded an accuracy of 68.7%. In addition, DeepDILI was compared with a published DL DILI prediction model using three external validation sets, resulting in the DeepDILI model achieving better results with two data sets and comparable result with one.

Adaptability of DeepDILI: An assessment based on a real-world scenario

To explore the adaptability of an AI solution for drug risk, we implemented a time-split based adaptability framework using our DeepDILI prediction model (Li et al., 2021). We utilized our DILI Severity and Toxicity (DILIst) dataset, which is currently the largest binary human DILI classification data set (Thakkar et al., 2020). The 1,002 drugs from DILIst were first split based on the drugs' approval year; 753 drugs with an approval year before 1997 were used for model development and 249 drugs with an approval year after 1997 were used for testing. To implement a time-split adaptability framework analysis, the 249 drugs (with an approval year of 1997–2019) were split into five chronological groups or buckets of relatively the same size (Figure 1). Drugs approved from 1997 to 1998 were put into bucket 1, 1999 to 2001 in bucket 2, 2002 to 2004 in bucket 3, 2005 to 2007 in bucket 4, and 2008 to 2019 into bucket 5, with 53 (36+/17–), 44(29+/15–), 46(24+/22–), 45 (23+/22–), and 61 (38+/23–) drugs, respectively in each bucket (Figure 1). DILI positive and negative are labeled as "+"and "–", respectively.

The adaptability of DeepDILI was assessed by adding drugs from each of the previously mentioned buckets by year into the training set to develop adaptive DeepDILI models (Figure 2A). The new training set was used to develop a new and evolved DeepDILI model. More in depth details about the model development can be found in our previous DeepDILI work (Li et al., 2021). To mimic the real-world scenario of annually adding novel drugs to the market, we increased the number of new drugs by stepwise and chronologically adding each bucket of drugs. Through this method, there was at most four buckets of drugs added to the initial locked training set (i.e., the 753 drugs approved before 1997) and one bucket used for evaluating the performance of the adaptive models. For example, if bucket 5 containing drugs approved from 2008 to 2019 was used as the test set, the adaptative models were developed as follows (Figure 2B). The first adaptative model was developed with the locked training set (602 drugs approved before 1997) in addition to the new drugs from bucket 1 (53 drugs approved in 1997 and 1998) and evaluated with bucket 5 (61 drugs approved in 2008 to 2019). The second adaptive model was developed with the locked training set in addition to the new drugs from bucket 1 and bucket 2 (44 drugs approved in 1999 to 2001) and evaluated with bucket 5. The third adaptive model was developed with the locked training set in addition to the new drugs from buckets 1 through 3 (46 drugs approved in 2002 to 2004) and evaluated with bucket 5. The fourth adaptive model was developed with the locked training set in addition to the new drugs from buckets 1 through 4 (45 drugs approved in 2005 to 2007) and evaluated with bucket 5. Additionally, the performance of the four adaptive models were compared with that of the initial DeepDILI model with the test bucket, which in this case is bucket 5. This process was reiterated five times. Each time a different bucket served as the new test set and all remaining buckets were chronologically added to the training set as described above. The data and code are available through https://github.com/TingLi2016/Adaptability.

To assess the adaptative nature of DeepDILI, seven performance metrics were compared between the locked and adaptive DeepDILI models. We calculated seven performance metrics to evaluate the performance of the model: the area under the receiver operating characteristic curve (AUC), accuracy, sensitivity, specificity, F1, Matthew's correlation coefficient

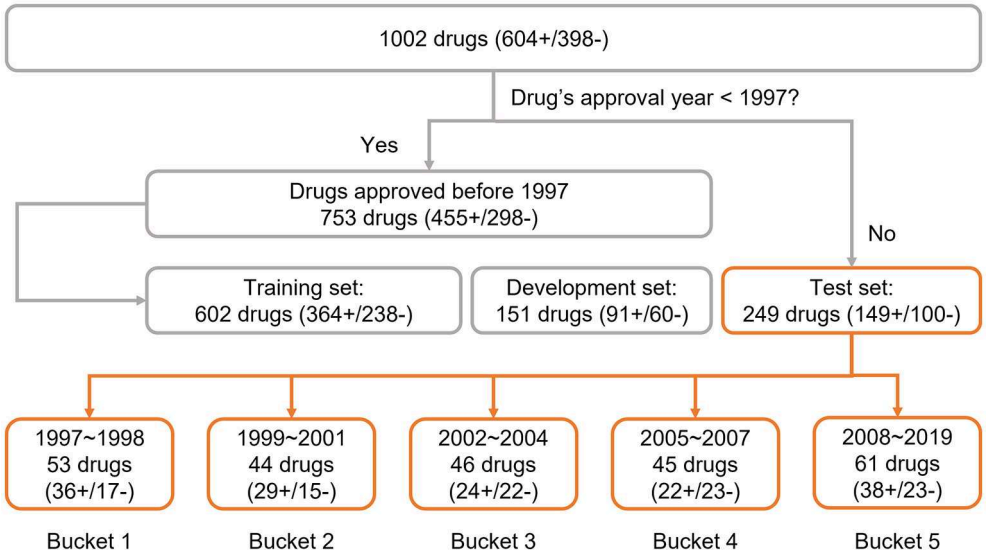


FIGURE 1
Data preparation: the data set was adopted from the previous DeepDILI study. The DeepDILI test set was split into five buckets based on the information of drugs' approval year. DILI positive and negative was labeled as "+"and "−".

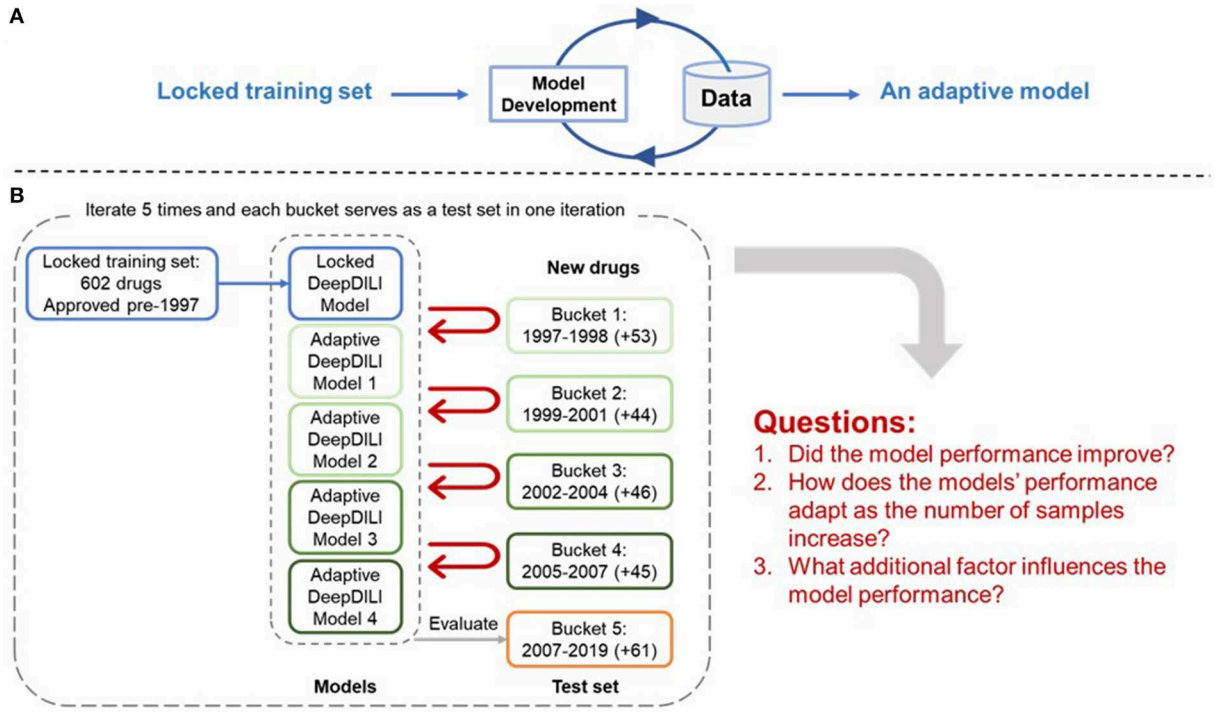


FIGURE 2
Adaptability Assessment Framework. **(A)** General framework of the adaptive model development, where the DeepDILI model adapts to new data by incorporating more data in the initial training set; **(B)** One iteration of the adaptability assessment process. In this iteration, bucket 5 was used as the test set, and the other four buckets served as the new drugs, that were chronologically and incrementally added to the initial training set. The process iterates five times as each bucket served as a test set.

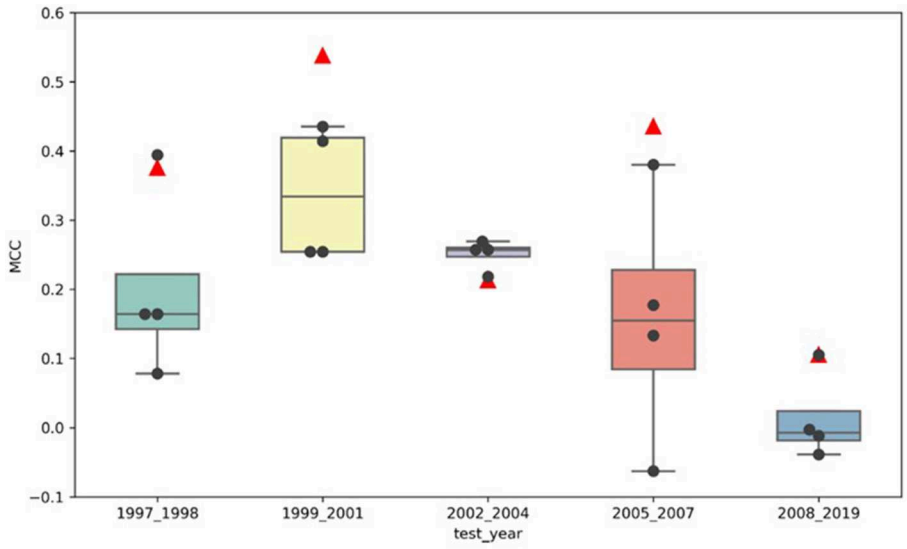


FIGURE 3
MCC distribution of the locked DeepDILI and adaptive DeepDILI models: the red triangle is the MCC of locked DeepDILI and the black dots represent the MCCs of the adaptive DeepDILI models for every test bucket. For example, 1997_1998 means that the tested drugs were approved in 1997 and 1998.

(MCC), and balanced accuracy (BA), were calculated using the following formulas:

True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN)

$$accuracy = \frac{TP + TN}{TP + TN + FN + FP} \quad (1)$$

$$sensitivity = \frac{TP}{TP + FN} \quad (2)$$

$$specificity = \frac{TN}{TN + FP} \quad (3)$$

$$F1 = \frac{2TP}{2TP + FP + FN} \quad (4)$$

$$MCC = \frac{TP^{*}TN - FP^{*}FN}{\sqrt{(TP + FP)^{*}(TP + FN)^{*}(TN + FP)^{*}(TN + FN)}}$$

$$BA = \frac{sensitivity + specificity}{2} \quad (5)$$

MCC ranges from −1 to 1, with extreme values −1 and 1 representing perfect misclassification and perfect classification, respectively. All the other six metrics range from 0 to 1; a score of 1 indicates the model makes correct decision on every test case. Thus, the higher value the better. Although we evaluated seven metrics for the locked and adaptative DeepDILI models, it was common to find that one model had better performance in some metrics but may be inferior to other metrics during the model comparison. Therefore, we selected MCC as the main metric, which has proven to have advantages in the binary classifications for an unbalanced data set (Chicco and Jurman, 2020; Chicco et al., 2021).

Key questions in adaptability assessment for the DeepDILI model

Has the model performance improved?

Figure 3 illustrates the comparison of the MCCs for the adaptive models (marked by the black dots) to the MCCs of the locked model (marked by the red triangles) for all five test sets, buckets 1–5. The locked DeepDILI model achieved the highest MCC of 0.538 and 0.436 in comparison to the adaptive models in the same test sets for bucket 2 (1999 to 2001) and bucket 4 (2005 to 2007), a comparable MCC of 0.376 and 0.106 in comparison to the adaptive models in the same test sets for bucket 1 (1997 to 1998) and bucket 5 (2008 to 2019), and the lowest MCC of 0.213 in comparison to the adaptive models in the same test sets for bucket 3 (2002 to 2004). Thus, we found that bucket 3 (2002 to 2004) was the only bucket in which the adaptive models MCC improved, as more drugs were added, in comparison to the locked DeepDILI model. The same trend was observed for the accuracy and F1, but a slight variance was found in the AUC, BA, sensitivity and specificity. Detailed information for these seven performance metrics can be found in Supplementary Table 1.

How does the performance of the model adapt as the number of drugs increases?

To investigate whether the model performance was positively associated with the increasing number of drugs in the training set, we assessed the MCCs of the locked DeepDILI

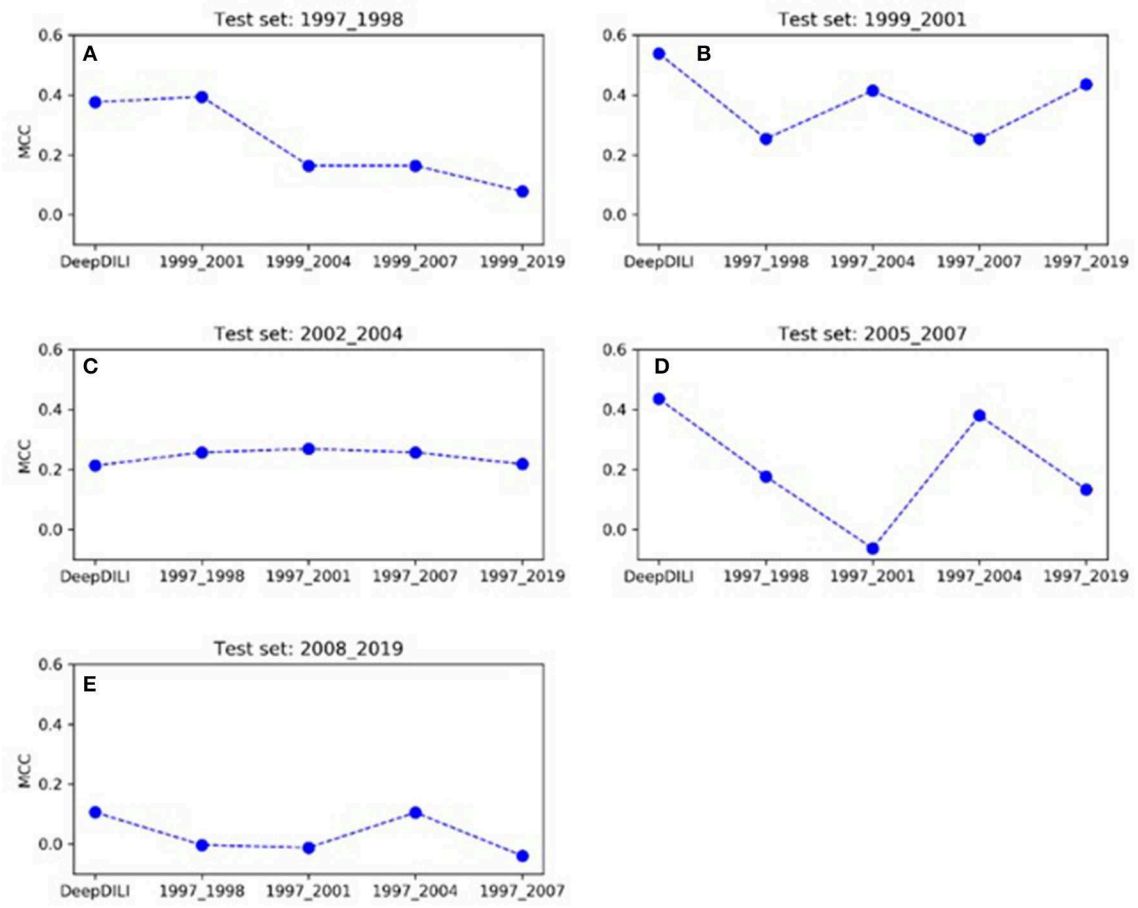


FIGURE 4
The trend of MCC among the locked DeepDILI and adaptive DeepDILI models within each buckets test set: for example, **(A)** showed the MCC trend of the locked DeepDILI model (labeled DeepDILI) and four adaptive DeepDILI models (labeled by the added drugs' approval year) on the test set with the drugs approved in 1997- and 1998. The following 4 sub-figures **(B–E)** follow this exact trend with their corresponding years.

model (labeled as DeepDILI) and individual adaptive DeepDILI model for each test set (Figure 4). The locked DeepDILI model, which has the smallest number of drugs in the training set as compared to the adaptative DeepDILI models, was used as a baseline. In Figure 4A, the MCCs of the adaptive DeepDILI models for the test set of bucket 1 (1997 to 1998) decreased as more drugs were added to the training set. In Figures 4B,D, the MCCs of the adaptive models for the test sets of buckets 2 (1999 to 2001) and 4 (2005 to 2007) presented as a wave shape as more drugs were added to the training set. In Figures 4C,E, the MCCs of the adaptive models for the test sets of buckets 3 (2002 to 2004) and 5 (2008 to 2019) exhibited a relatively flat trend as more drugs were added to the training set, indicating that as more drugs were used in the training, the performance of the adaptive models did not improve. Thus, there is no positive relationship between the model performance and the number of drugs in the training set. In addition, no general pattern was found in the adaptive models performance as we increased the number of drugs in the training set.

What additional factors influence the models' performance?

As the performance of the models adapted to the addition of new drugs, we observed the average MCC varied from one test set to another (Supplementary Table 1). The test set of bucket 2 (1999 to 2001) achieved the highest average MCC of 0.379, while bucket 5 (2008 to 2019) yielded the lowest average MCC of 0.031. The test sets of buckets 1 (1997 to 1998), 3 (2002 to 2004), and 4 (2005 to 2007) yielded similar average MCCs of 0.235, 0.243, and 0.213, respectively. This indicates that different test sets presented various levels of challenges for DILI prediction, showing that the properties of the test set data are a key factor in the model's performance.

Discussion

Although AI is promising, there is still work to do; a comprehensive assessment of the adaptive behavior and context-of-use of AI models for regulatory application is required. As two important aspects of regulatory significance, especially for the application of AI, the applicability domain and context of use play a significant role in enhancing AI solutions for risk assessments within the regulatory arena. On every occasion, the context of use should clearly convey to users where the model is best utilized as well as whether the model is intended to complement or replace current technologies (Anklam et al., 2022), while the applicability domain outlines how the model is used through defining best practices (Anklam et al., 2022).

When it comes to using adaptive models and assessing their adaptive behavior there are a number of strategies and approaches being used across the field of AI (Groce et al., 2002; Yang et al., 2005; Xiao et al., 2016; López and Tucker, 2018). Currently, a random split cross-validation model is considered the ML standard for model building and evaluation (Morita et al., 2022). Random split cross-validation is often found to be overoptimistic in comparison to real-world situations, while a time-split approach is considered suitable for real-world prediction (Morita et al., 2022). In this study, we proposed a time-split adaptability framework approach to exploring the adaptive behavior of an AI-based solution for drug toxicity and risk assessments within regulatory science. In using the time-split approach, we were able to discuss two important questions: (1) Did the models performance improve or decline as more data was added? And (2) Did the context of use change as the model adapted?

Through the real-world scenario of annually adding new drugs to the market to retrain our model, we found that the target test set plays a major role in the adaptive behavior of our model. Our findings suggest that regardless of the individual model performance, the average MCC was found to vary from one test set to another. This indicates that different test sets possess different levels of challenge for prediction, demonstrating that the target test set appears to be the most important factor in performance. The context of use for our DeepDILI model was the same for the locked and adaptive models. DeepDILI aims to flag the human DILI potential of DILI positive drugs using the chemical structure that have a molecular weight lower than 1,000 g/mol. Since these criteria were used to screen the drugs for the initial model that our adaptive framework was remodeled from our context of use did not change as the model adapted to "new" data. Although a time-split approach is seen to be better for real-world prediction, a major caveat of this approach are the limitations with respect to the amount of usable or available data for model training, development, and testing. In future studies, it would be beneficial to assess the application of our adaptive framework to other types of predictive models to determine their adaptive behavior. Since drug induced organ injury is a leading cause of drug withdrawals, it would be beneficial to see how our locked and adaptive model frameworks perform when used on other organ systems.

Our results indicated that adding more drugs to the training set did not substantially contribute to the performance of the adaptive DeepDILI model. Overall, based on these findings we conclude that the proposed adaptability assessment framework has utility in the evaluation of a model's adaptive performance over time, which would greatly support the advancement of AI-based models in regulatory science. Using comprehensive assessments to evaluate the adaptive behavior and context-of-use of AI based safety evaluation and risk assessment models, whether locked or adaptive, can have a positive impact on decision making within regulatory science. Currently, reviewers utilize animal pharmacology and toxicology data, manufacturing information, clinical protocols and any past knowledge of the compound to assess the safety of a new drug. The development and parallel use of alternative approaches to identify and signal different safety concerns earlier in the review process are essential to the future of regulatory science.

Author contributions

WT devised the study. SC and TL wrote the manuscript and performed data analysis. WT, ZL, RR, and ST revised the manuscript. All authors read and approved the final manuscript.

Conflict of interest

Author RR is co-founder and co-director of ApconiX, an integrated toxicology and ion channel company that provides expert advice on non-clinical aspects of drug discovery and drug development to academia, industry, and not-for-profit organizations.

The remaining 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.

Author disclaimer

This manuscript reflects the views of the authors and does not necessarily reflect those of the Food and Drug Administration. Any mention of commercial products is for clarification only and is not intended as approval, endorsement, or recommendation.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frai.2022.1034631/full#supplementary-material

References

Anklam, E., Bahl, M. I., Ball, R., Beger, R. D., Cohen, J., Fitzpatrick, S., et al. (2022). Emerging technologies and their impact on regulatory science. *Exp. Biol. Med.* 247, 1–75. doi: 10.1177/15353702211052280

Chicco, D., and Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC genomics.* 21, 1–13. doi: 10.1186/s12864-019-6413-7

Chicco, D., Totsch, N., and Jurman, G. (2021). The Matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation. *BioData Min.* 14, 13. doi: 10.1186/s13040-021-00244-z

FDA (2021). *Advancing Regulatory Science at FDA: Focus Areas of Regulatory Science (FARS)*. MD, USA: FDA Silver Spring.

Groce, A., Peled, D., and Yannakakis, M. (2002). "Tools and Algorithms for the Construction and Analysis of Systems." In: *Lecture Notes in Computer Science*. eds. JP. Katoen, and P. Stevens (Berlin; Heidelberg: Springer) 2280. doi: 10.1007/3-540-46002-0_25

Gupta, R., Srivastava, D., Sahu, M., Tiwari, S., Ambasta, R. K., Kumar, P., et al. (2021). Artificial intelligence to deep learning: machine intelligence approach for drug discovery. *Mol. Divers.* 25, 1315–1360. doi: 10.1007/s11030-021-10217-3

Hamburg, M. A. (2011). Advancing regulatory science. *Science.* 331, 987. doi: 10.1126/science.1204432

Health, U. D.,o., Services, H., Food, & Administration, D., et al. (2018). *Software as a Medical Device (SaMD)*. 2018; Available online at: https://www.fda.gov/medical-devices/digital-health-center-excellence/software-medical-device-samd,

Hong, H., Xie, Q., Ge, W., Qian, F., Fang, H., Shi, L., et al. (2008). Mold2, molecular descriptors from 2D structures for chemoinformatics and toxicoinformatics. *J. Chem. Inf. Model.* 48, 1337–1344. doi: 10.1021/ci800038f

Li, T., Tong, W., Roberts, R., Liu, Z., and Thakkar, S. (2021). DeepDILI: deep learning-powered drug-induced liver injury prediction using model-level representation. *Chem. Res. Toxicol.* 34, 550–565. doi: 10.1021/acs.chemrestox.0c00374

López, C., and Tucker, C. (2018). Toward personalized adaptive gamification: a machine learning model for predicting performance. *IEEE Trans. Games.* 12, 155–168. doi: 10.1109/TG.2018.2883661

Morita, K., Mizuno, T., and Kusuhara, H. (2022). Investigation of a data split strategy involving the time axis in adverse event prediction using machine learning. *arXiv preprint arXiv:*2204.08682. doi: 10.1021/acs.jcim.2c00765

Patel, M., and Miller, M. A. (2012). Impact of regulatory science on global public health. *Kaohsiung J. Med. Sci.* 28, S5–9. doi: 10.1016/j.kjms.2012.05.003

PoPPer, H., Rubin, E., Gardiol, D., Schaffner, F., and Paronetto, F., et al. (1965). Drug-induced liver disease: a penalty for progress. *Arch. Intern. Med.* 115, 128–136. doi: 10.1001/archinte.1965.03860140008003

Slikker, W. Jr., Miller, M. A., Lou Valdez, M., and Hamburg, M. A. (2012). Advancing global health through regulatory science research: summary of the global summit on regulatory science research and innovation. *Regul. Toxicol. Pharmacol.* 62, 471–473. doi: 10.1016/j.yrtph.2012.02.001

Text-H.R.2565-117th Congress (2021–2022). (2021). FDA Modernization Act of 2021.

Text-H.R.2575-116th Congress (2019–2020). (2020). AI in Government Act of 2020.

Text-H.R.7667-117th Congress (2021–2022). (2022). Food and Drug Amendments of 2022.

Text-S.2952-117th Congress (2021–2022). (2021). FDA Modernization Act of 2021.

Text-S.3035-117th Congress (2021–2022). (2022). GOOD AI Act of 2021, in, S.3035.

Thakkar, S., Li, T., Liu, Z., Wu, L., Roberts, R., and Tong, W. (2020). Drug-induced liver injury severity and toxicity (DILIst): Binary classification of 1279 drugs by human hepatotoxicity. *Drug Discov.* 25, 201–208. doi: 10.1016/j.drudis.2019.09.022

U. S. Food and Drug Administration (2017). *FDA's Predictive Toxicology Roadmap*. MD: Silver Spring.

U. S. Food and Drug Administration (2019a). *International Medical Device Regulators Forum.*

U. S. Food and Drug Administration (2019b). *Proposed Regulatory Framework for Modifications to Artificial Intelligence/Machine Learning (AI/ML)-Based Software as a Medical Device* (SaMD).

U. S. Food and Drug Administration (2021a). *Drug Development Tool (DDT) Qualification Programs*. Available online at: https://www.fda.gov/drugs/development-approval-process-drugs/drug-development-tool-ddt-qualification-programs

U. S. Food and Drug Administration (2021b). *Artificial intelligence/machine learning (AI/ML)–based software as a medical device (SaMD) action plan*. MD: FDA Silver Spring,.

U. S. Food and Drug Administration (2021c). *Innovative Science and Technology Approaches for New Drugs (ISTAND) Pilot Program*. Available online at: https://www.fda.gov/drugs/drug-development-tool-ddt-qualification-programs/innovative-science-and-technology-approaches-new-drugs-istand-pilot-program

Van Norman, G. A. (2019). Limitations of animal studies for predicting toxicity in clinical trials: is it time to rethink our current approach? *JACC: Basic Transl. Sci.* 4, 845–854. doi: 10.1016/j.jacbts.2019.10.008

Xiao, B., Xiong, J., and Shi, Y. (2016). "Novel applications of deep learning hidden features for adaptive testing." In: *2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC)*, 743–748. doi: 10.1109/ASPDAC.2016.7428100

Yang, J., Rivard, H., and Zmeureanu, R. (2005). On-line building energy prediction using adaptive artificial neural networks. *Energy Build.* 37, 1250–1259. doi: 10.1016/j.enbuild.2005.02.005

Zimmerman, H. J. (1999). *Hepatotoxicity: the adverse effects of drugs and other chemicals on the* liver.

frontiers | Frontiers in Big Data

TYPE Brief Research Report
PUBLISHED 16 November 2022
DOI 10.3389/fdata.2022.1018356

OPEN ACCESS

EDITED BY
Prashanti Manda,
University of North Carolina at Greensboro, United States

REVIEWED BY
Mai Oudah,
New York University, United States
C. Titus Brown,
University of California, Davis, United States

*CORRESPONDENCE
Vinhthuy Phan
vphan@memphis.edu

SPECIALTY SECTION
This article was submitted to Medicine and Public Health, a section of the journal Frontiers in Big Data

RECEIVED 13 August 2022
ACCEPTED 31 October 2022
PUBLISHED 16 November 2022

CITATION
Pham D-T and Phan V (2022) Representing bacteria with unique genomic signatures.
Front. Big Data 5:1018356.
doi: 10.3389/fdata.2022.1018356

Representing bacteria with unique genomic signatures

Diem-Trang Pham and Vinhthuy Phan*

Department of Computer Science, University of Memphis, Memphis, TN, United States

Classifying or identifying bacteria in metagenomic samples is an important problem in the analysis of metagenomic data. This task can be computationally expensive since microbial communities usually consist of hundreds to thousands of environmental microbial species. We proposed a new method for representing bacteria in a microbial community using genomic signatures of those bacteria. With respect to the microbial community, the genomic signatures of each bacterium are unique to that bacterium; they do not exist in other bacteria in the community. Further, since the genomic signatures of a bacterium are much smaller than its genome size, the approach allows for a compressed representation of the microbial community. This approach uses a modified Bloom filter to store short k-mers with hash values that are unique to each bacterium. We show that most bacteria in many microbiomes can be represented uniquely using the proposed genomic signatures. This approach paves the way toward new methods for classifying bacteria in metagenomic samples.

KEYWORDS

metagenomics, Bloom filter, bacteria detection, NGS analysis, k-mers

1. Introduction

Metagenomics is the study of analyzing genomes contained in environmental samples. Recent metagenomic studies revealed that the knowledge of the microbial composition in the human gut shows certain complex mechanisms of disorders of human health (Handelsman et al., 2007), such as diverse as diabetes, depression and rheumatoid arthritis. And although the dysbiosis has been proved to link to the gastrointestinal tract (Eloe-Fadrosh and Rasko, 2013), it can be on any exposed surface or mucus membrane, such as the skin or the respiratory system. This variation can impact the human health (Martín et al., 2014). A challenge in metagenomics that is caused by large and complex metagenomic data is the identification and classification of bacteria in microbial communities that consist of thousands or more environmental microbial species (Teeling and Fo, 2012; Sharpton, 2014). A number of approaches have been developed, including alignment reads to reference genomes, analyzing taxonomically informative gene markers, clustering sequences, assembling sequences into genomes and using k-mer based approach. In any approach, it requires a set of reference genomes as a database or an index. In alignment approach, the metagenome sequences (or reads) from the environment are aligned to the reference genome database. In k-mer based approach, an index is created from k-mers of the reference genomes, and this index is used in identification or profiling. While alignment approach has been shown to be

accurate, they require large amounts of time and resources. There are many approaches that utilize gene markers or k-mer have been introduced to reduce the running time while still achieving the high accuracy (Lindgreen et al., 2016).

A Bloom filter is a probabilistic data structure that provides very fast membership queries. This useful data structure has been used in several applications in bioinformatics and metagenomics. FACS (Stranneheim et al., 2010) creates a Bloom filter for each reference genome and inserts all k-mers in the filter. Later in query, if a match was found for a k-mer, a match score is computed and it has to surpass a threshold to be classified to a reference genome. BFCounter (Melsted and Pritchard, 2011) introduces an application of Bloom filter to count the k-mers efficiently. BioBloom tool (Chu et al., 2014) applied Bloom filter to create a filter-based sequence-screening tool which was claimed to be faster than BWA, Bowtie 2 and FACS. And another research in building Bloom filters (Pellow et al., 2017) with one-sided k-mers, two-sided k-mers and sparse k-mers data structures improves the performance of the Bloom filter, which will be useful in genome assembly, sequence comparison and sequence search applications. Sequence Bloom Tree (Solomon and Kingsford, 2016), another application of Bloom Filter, is a method for querying thousands of short-read sequencing in RNA-seq experiments for expressed isoforms. This method was able to search large collections of RNA-seq experiments for a given transcript order of magnitude faster than existing approaches.

Most of the existing work use one Bloom filter for each genome, this may not efficiently represent a microbiome or community. In this work, we introduce a method that uses a modified Bloom filter to store unique signatures of bacteria. As such, it can be used to provide unique representation of bacteria in microbiomes. We also show that this method can be used to retrieve species in two microbiomes.

2. Methods

Similar to other existing profiling methods, our method consists of two procedures. The first procedure builds an *index* based on the genomes of all the bacteria that might exist in metagenomic samples. The index stores unique genomic signatures of each genome in the microbiome. Once an index is built, it can be used to identify, classify or profile metagenomic samples. Given reads in a metagenomic sample, the second procedure, known as *the querying phase*, makes a query for each read to identify which bacterial genome the read may come from.

2.1. Set membership determination with bloom filters

A Bloom filter is a space-efficient probabilistic data structure used for set membership queries. Technically, a Bloom filter is an M-bit array B, which is initially all zeros, together with a set of n hash functions. To prepare a Bloom filter for identifying elements in a universe of elements, each element x_i is hashed to obtain n hashed values $h_1(x_i), \cdots, h_n(x_i)$. Each entry $B(h_j(x_i))$ is set to 1.

To check whether an item y exists in B, n hash values $h_1(y), \cdots, h_n(y)$ are computed. If all values are 1, the query answer is True. If not, it is False.

In membership querying, a Bloom filter does not make a false negative. A query to an element in the universe, which is stored in the filter, always correctly returns True. A false positive, however, can happen. Due to the nature of simply setting all hashed entries to 1 in the filter building phase, it is possible that the query of an element z that is not stored in the filter actually returns True. It is known that to minimize the probability of getting false positives, the optimal number of hash functions should be $\frac{b \ln 2}{m}$, where b is the size (number of bits) of the filter, and m is the number of elements stored in the filter (Bloom, 1970).

2.2. Finding k-mers with genome-unique hash values

Given a set of referenced bacterial genomes that might exist in the metagenomic environment of interest, an index, F, which is a modified Bloom filter, is built to store unique genomic signatures of each genome.

The index, F, is an array with m entries. During the processing of referenced genomes, k-mers from these genomes are hashed into F using n randomly generated hash functions. A k-mer x is hashed into n entries $h_1(x), \cdots, h_n(x)$ of F. After all referenced genomes are processed, an entry of F with a positive value g corresponds to a k-mer, whose hash values are unique to genome g. This allows F to be used in ways similar to those of a Bloom filter to detect genomes that are present in the metagenomic sample. The construction of F consists of two main phases. In each phase, all genomes are sequentially processed by Algorithm 1. In both phases, Algorithm 1 shares a common goal: it attempts to identify k-mers with hash values that are unique to the genome. It does this by going through each k-mer of the genome and marking all n locations (determined by n hash values) with dirty or with the genome id. A location is dirty (set to -1) if two k-mers on two different genomes get hashed to it. If a location is not dirty, it stores the id of some genome. If a k-mer x of genome g_1 is hashed to an entry that holds the id of another genome, say g_2, then that x is not unique

```
positions = []
for k-mer k at position pos in genome gid do
  unique = True
  idx = []
  for each hash function f do
    v = f(k)
    idx.append(v)
    if F[v] ≠ 0 and F[v] ≠ gid then
      unique = False
  if unique then
    if phase == 2 then
      positions.append(pos)
    for each value v in idx do
      F[v] = gid
  else
    for each value v in idx do
      F[v] = −1
if phase == 2 then
  Reduce(F, gid, positions)
```

Algorithm 1. ProcessGenome(*F*, *gid*, *phase*).

```
selected = [positions[0]]
for i = 1; i < len(positions); i = i + 1 do
  if selected[len(selected) − 1] + ω < positions[i] then
    selected.append(positions[i])
  else
    Let x be the k-mer at positions[i] in genome gid
    for each hash function f do
      F[f(x)] = −1
for each position p in selected do
  Let x be the k-mer at p in genome gid
  for each hash function f do
    F[f(x)] = gid
```

Algorithm 2. Reduce(*F*, *gid*, *positions*).

and all entries $h_1(x), \cdots, h_n(x)$ of F are set to dirty. If x is deemed unique, the genome id is stored in all of these entries. Suppose that after Phase 1, genomes $g_1, \cdots, g_l$ are processed sequentially in this order. Entries in F with values g_1 may not correspond to k-mers with unique hash values. To see this, suppose k-mer x appears in both g_1, k-mer y appears in g_2, and some of the hash values of x and y are the same. Because g_2 is processed after g_1, all the entries corresponding to the hash values of y are set to dirty, but not all the entries corresponding to the hash values of x are set to dirty.

It is, however, important to understand that after Phase 1, entries in F with values g_l will in fact correspond to k-mers in genome g_l with hash values unique to this genome. Since g_l is processed last, if an entry in F has value g_l, it means some k-mer in g_l with hash values that do not collide with any k-mer in all the other genomes that are already processed. Thus, this k-mer has hash values that are unique; no other k-mer in any other genome shares one of these hash values. Therefore, when a genome is processed by Algorithm 1 after all of the other genomes have already been processed, all of k-mers with unique hash values in that genome are correctly marked in F. This means that after Phase 2, when all genomes are processed again by Algorithm 1, all k-mers with unique hash values in all genomes will be correctly marked in F.

2.3. Query phase: Reads processing

Given reads from a metagenomic sample, the main task is to identify which bacteria exist in the sample. This boils down to processing reads and determining which bacterial genomes they most likely belong to. While all existing methods we are aware of process all reads in the metagenomic samples, the proposed method processes just enough reads to cover a fraction of bacterial genomes. This typically results in choosing a small random samples of reads for processing.

If a processed read belongs to a genome g and also contains a k-mer x with unique hash values stored in F, there is a good chance that the read will be correctly identified to belong to g. The read is not recognized if the k-mer x has a sequencing error or a genetic variant. A genetic variant can occur because the genome of the bacterium in the sample is likely not the same as the referenced genome of the same bacteria used to create F.

A processed read that does not belong to genome g might also be mistakenly identified to belong to g if it has a sequencing error or a genetic variant that results in a k-mer with hash value(s) collide with one of the k-mers of g stored in F.

Given a read to be processed, all k-mers are passed into k-mer processing to classify its g_i. Let V be the set of classified g_i of all k-mers of the read. If V consists of only 0 and/or –1, then the read is discarded. If, however, V consists of positive values, i.e., genome ids, then one of three different strategies can be used to determine which genome the read belongs to.

2.3.1. Majority

If there is a positive number, g, in V with frequency greater than 50%, then g is predicted to be the genome that contains the read. If there is no such number, then the read is discarded. This strategy is effective in the presence of significant amounts of sequencing errors and/or genetic variants. In such cases, a k-mer of the read can be misidentified to be a unique k-mer of a different bacteria. But if there are not too many of such mistakes, a majority of positive identification can identify the correct genome.

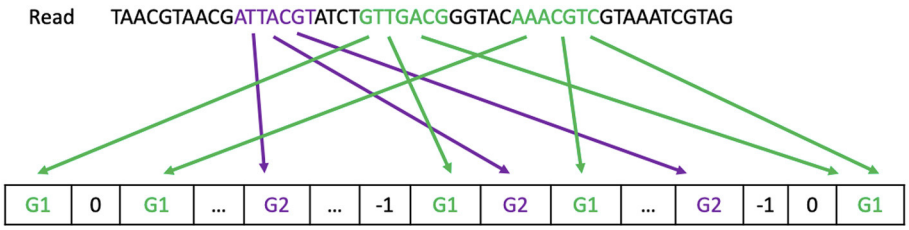


FIGURE 1
Read processing. The majority strategy predicts the read comes from G1. The First-hit strategy predicts the read comes from G2. The One-or-nothing strategy discards the read.

2.3.2. First-hit

K-mers are processed sequentially. When the first k-mer that has a positive hash value, g, is encountered, no additional k-mers are processed. g is predicted to be the genome that contains the read. This strategy is effective when k-mers stored in F are highly unique so that the first hit is most likely correct.

2.3.3. One-or-nothing

If V has only one positive value, g, then g is predicted to be the genome that contains the read. If this is not the case, the read is discarded. This strategy is highly conservative. If there is a disagreement, i.e., two genomes identified by different k-mers of the reads, the read is discarded from consideration.

Figure 1 gives an example on how each strategy classifies a read to a reference genome.

In order to optimize the running time of query phase, reads are distributed to different cores for processing.

3. Results

3.1. Experimental setup

To assess performance of our method, we used two microbial communities, with included 457 and 2,850 reference genomes, respectively. The first community consists of 457 reference genomes, named S1, combined from three metagenomes used by Mende et al. (2012) in a study of metagenomic assembly. To create a set of reference genomes, we extracted accession numbers from reads in these three metagenomes. This information allowed us to retrieve from NCBI reference genomes for the bacteria, from which the reads were created. The second community, named S2, includes genomes used in CAMI challenge (Sczyrba et al., 2017).

First, we show some statistics of the indexes of each reference genome set. Second, we compare results on different querying strategies. And finally, we also show the difference of indexes when using different number of hash functions.

3.2. Representing bacteria using unique signatures

We now report how the two microbial communities can be represented by unique genomic signatures. For the first set of bacterial genomes S1, we used 2 hash functions, k-mer of length 31 and the size of the index is 8GB. The index was built in two phases. All 457 genomes have unique signatures. Total number of signatures is 248,758,006. Minimum number of unique signatures is 152 and maximum number of unique signatures is 1,720,014.

As the more hash functions are used in building index, the more hash values are computed for each k-mer and the more unique it is. But that will also reduce the number of k-mers with unique hash values for each genome. Although larger genomes have a sufficient number of k-mers with unique hash values, smaller genomes have only a few of such unique k-mers. For this bacterial genome set S2, we build two indexes with the same k-mer size and index size, and only vary the number of hash functions to compare the effect on querying performance when different number of hash functions were used to build the index. Both indexes are built in 1 phase. All the genomes have unique signatures. Table 1 shows the total, the minimum, and the maximum number of signatures of each index. We found that the 3-hash-function index had fewer signatures than the 2-hash-function index. This is likely because as more hash values were computed, there was a higher chance of having collisions of those hash values. Figure 2 shows the distribution of number of unique signatures for each genome in genome set S2 in the change of number of hash functions when building index for set S2.

3.3. Querying

In order to evaluate the retrieval capability of our two indexes, we downloaded two simulated samples for querying.We

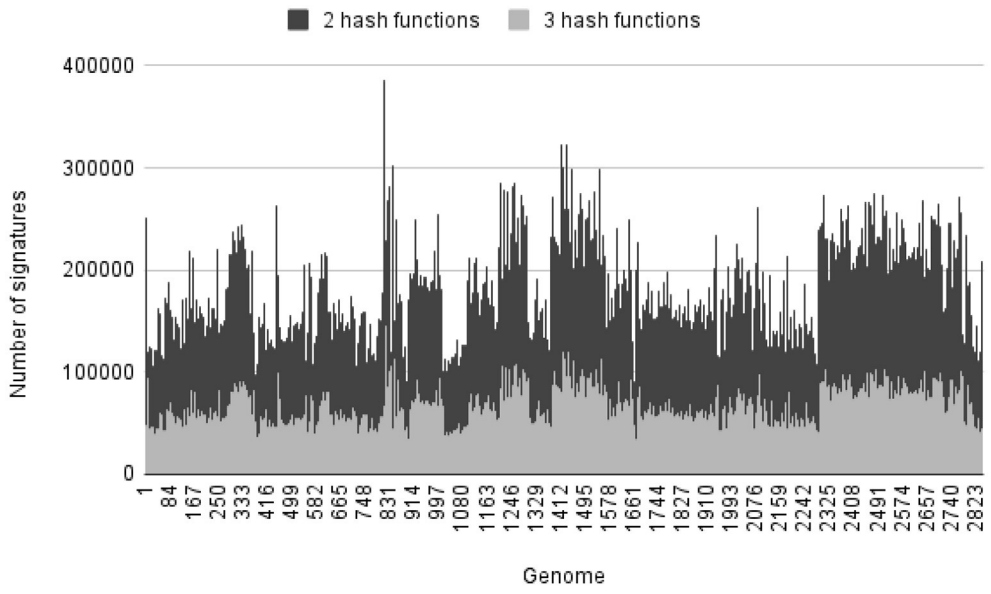


FIGURE 2
Number of unique signatures for each genome in genome set S2 in the change of number of hash functions.

TABLE 1 Comparison on number of signatures in the change of number of hash functions when building index for set S2.

Number of hash functions	Min	Max	Total
2	544	386,709	400,769,054
3	211	145,005	150,366,923

TABLE 2 Effect of different number of querying strategies.

Query strategy	Precision	Recall	F1-score
Majority	0.026	1.000	0.051
First-hit	0.026	0.990	0.051
One-or-nothing	0.028	0.987	0.053

TABLE 3 Effect of different number of hash functions.

Number of hash functions	Precision	Recall	F1-score
2	0.316	0.601	0.414
3	0.296	0.601	0.396

use the 10 species dataset by Mende et al. (2012) which consists of genomic reads from 10 genomes in S1, and the RH_S001 dataset from Sczyrba et al. (2017) consists of 302 genomes from S2. We will refer to these datasets as 10 species and RH_S001 in subsequent discussions. Reads from 10 species and RH_S001 are paired-end and were simulated with charateristics of Illumina sequencing technology with length of 75 and 150 bp, respectively. The 10 species dataset is used to query bacteria in S1, and the RH_S001 is queried in S2.

Performance was measured in terms of precision, recall and F1-score as accuracy of the predicting process. Precision is computed as the number of correctly queried bacteria divided by the total number of predicted bacteria. Recall is the number of correctly queried bacteria divided by the total number of bacteria that actually exist in the sample. F1-score is the harmonic mean of precision and recall.

The 10 species sample is queried on index of set S1 using majority strategy. We were able to query all 10 species, results in recall of 100%. However, there are many incorrect querying, this leads to low precision of 2.6%. The F1-score is 5%. We also evaluated the performance of different querying strategies. As described earlier, the majority query strategy looks at all k-mers and picks the genome that shows up at least 50% among all k-mers. The one-or-nothing query strategy picks a genome only if it is the only genome predicted by all k-mers of the read. The first-hit strategy picks the first genome that is predicted by some k-mer of the read. Each of these strategies has its own pros and cons. And the most appropriate strategy depends on the dataset. Table 2 shows the performance resulted from each of the three query strategies.

We found that the performance resulted from the three query strategies was very similar. Both majority and first-hit strategies had lower precision, but higher recall than one-or-nothing. One-or-nothing, by design, is more conservative, and therefore, should have fewer false positives, and higher precision than the other two strategies.

The RH_S001 sample is queried on the index of set S2 using the majority strategy. There are 162 out of 302 genomes correctly predicted. Only 5 genomes in the sample are missing as there may have sequencing errors in the reads that causes wrong prediction to other genomes. Another reason is that no read has the exact unique signatures in the index. This leads to a precision of 26%, recall of 97% and F1-score is 41%.

Table 3 shows the effect on querying performance when two or three hash functions were used to build the index. We found that using 2 hash functions to build an index resulted in a slightly better overall performance than using 3 hash functions. While recall rates were similar, precision rates were higher when 2 hash functions were used. In this experiment, we used the majority strategy, and having more signatures could be useful for this querying strategy to reduce the false positive, which improves the precision.

4. Discussion

We introduced a method for representing bacteria in a microbial community uniquely. We showed that our method could be used to query reads in metagenomic samples. A method for efficiently representing bacteria in a microbial community would be useful for post-processing in order to have an accurate identification of bacteria, which requires more analysis as well as data interpretation on the query outputs. And due to the close relationship between the microbiome and health, improving the accuracy of bacteria identification would help to make metagenomic analysis more meaningful in understanding the human microbiome in health and disease. There is room to find parameters that can improve the performance of the query phase. Also, additional improvements can be made in the future to determine these choices more appropriately under different criteria.

Similar to most of other k-mer based approaches, when the database consists of hundreds of thousands reference genomes, it is challenging for the proposed method to obtain unique signatures for some genome, especially very small genomes. This method, however, can be promising for microbiomes that are not too big, e.g., skin, oral, or gut microbiomes.

Data availability statement

Publicly available datasets were analyzed in this study. This data can be found at: illumina 10 species http://www.bork.embl.de/~mende/simulated_data/; High complexity sample 1 https://edwards.flinders.edu.au/cami-challenge-datasets/.

Author contributions

D-TP and VP designed the methods and experiments and wrote the paper. D-TP wrote code, downloaded data, and ran experiments. Both authors contributed to the article and approved the submitted version.

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

References

Bloom, B. H. (1970). Space/time trade-offs in hash coding with allowable errors. *Commun. ACM*. 13, 422–426. doi: 10.1145/362686.362692

Chu, J., Sadeghi, S., Raymond, A., Jackman, S. D., Nip, K. M., Mar, R., et al. (2014). Biobloom tools: fast, accurate and memory-efficient host species sequence screening using bloom filters. *Bioinformatics* 30, 3402–3404. doi: 10.1093/bioinformatics/btu558

Eloe-Fadrosh, E. A., and Rasko, D. A. (2013). The human microbiome: from symbiosis to pathogenesis. *Annu. Rev. Med.* 64, 145. doi: 10.1146/annurev-med-010312-133513

Handelsman, J., Tiedje, J., Alvarez-Cohen, L., Ashburner, M., Cann, I., Delong, E., et al. (2007). "The new science of metagenomics: Revealing the secrets of our microbial planet," in *National Research Council (US) Committee on Metagenomics: Challenges and Functional Applications* (Washington, DC: National Academies Press U.S.).

Lindgreen, S., Adair, K., and Gardner, P. (2016). An evaluation of the accuracy and speed of metagenome analysis tools. *Sci. Rep*. 6, 19233. doi: 10.1038/srep 19233

Martín, R., Miquel, S., Langella, P., and Bermúdez-Humarán, L. G. (2014). The role of metagenomics in understanding the human microbiome in health and disease. *Virulence* 5, 413–423. doi: 10.4161/viru. 27864

Melsted, P., and Pritchard, J. K. (2011). Efficient counting of k-mers in dna sequences using a bloom filter. *BMC Bioinform*. 12, 1–7. doi: 10.1186/1471-2105-12-333

Mende, D. R., Waller, A. S., Sunagawa, S., Järvelin, A. I., Chan, M. M., Arumugam, M., et al. (2012). Assessment of metagenomic assembly using simulated next generation sequencing data. *PLoS ONE* 7, e31386. doi: 10.1371/journal.pone.0031386

Pellow, D., Filippova, D., and Kingsford, C. (2017). Improving bloom filter performance on sequence data using k-mer bloom filters. *J. Computat. Biol.* 24, 547–557. doi: 10.1089/cmb.2016.0155

Sczyrba, A., Hofmann, P., Belmann, P., Koslicki, D., Janssen, S., Dröge, J., et al. (2017). Critical assessment of metagenome interpretation–a benchmark of metagenomics software. *Nat. Methods* 14, 1063–1071. doi: 10.1038/nmeth.4458

Sharpton, T. (2014). An introduction to the analysis of shotgun metagenomic data. *Front. Plant Sci.* 5, 209. doi: 10.3389/fpls.2014.00209

Solomon, B., and Kingsford, C. (2016). Fast search of thousands of short-read sequencing experiments. *Nat. Biotechnol.* 34, 300–302. doi: 10.1038/nbt.3442

Stranneheim, H., Käller, M., Allander, T., Andersson, B., Arvestad, L., and Lundeberg, J. (2010). Classification of dna sequences using bloom filters. *Bioinformatics* 26, 1595–1600. doi: 10.1093/bioinformatics/btq230

Teeling, H., and Fo, G. (2012). Current opportunities and challenges in microbial metagenome analysis-a bioinformatic perspective. *Brief. Bioinform.* 13, 728–742. doi: 10.1093/bib/bbs039

frontiers | Frontiers in Artificial Intelligence

TYPE Original Research
PUBLISHED 06 December 2022
DOI 10.3389/frai.2022.999289

OPEN ACCESS

EDITED BY
Ramin Homayouni,
Oakland University William Beaumont School of Medicine, United States

REVIEWED BY
Yan Cui,
University of Tennessee Health Science Center (UTHSC), United States
Jun Wu,
East China Normal University, China
Zhining Wen,
Sichuan University, China

*CORRESPONDENCE
Xiaowei Xu
xwxu@ualr.edu
Zhichao Liu
Zhichao.Liu@fda.hhs.gov

SPECIALTY SECTION
This article was submitted to Medicine and Public Health, a section of the journal Frontiers in Artificial Intelligence

RECEIVED 20 July 2022
ACCEPTED 16 November 2022
PUBLISHED 06 December 2022

CITATION
Wang X, Xu X, Tong W, Liu Q and Liu Z (2022) DeepCausality: A general AI-powered causal inference framework for free text: A case study of LiverTox.
Front. Artif. Intell. 5:999289.
doi: 10.3389/frai.2022.999289

DeepCausality: A general AI-powered causal inference framework for free text: A case study of LiverTox

Xingqiao Wang[1], Xiaowei Xu[1]*, Weida Tong[2], Qi Liu[3] and Zhichao Liu[2]*

[1]Department of Information Science, University of Arkansas at Little Rock, Little Rock, AR, United States, [2]Division of Bioinformatics and Biostatistics, National Center for Toxicological Research, US Food and Drug Administration, Jefferson, AR, United States, [3]Office of Clinical Pharmacology, Office of Translational Sciences, Center for Drug Evaluation and Research, US Food and Drug Administration, Silver Spring, MD, United States

Causality plays an essential role in multiple scientific disciplines, including the social, behavioral, and biological sciences and portions of statistics and artificial intelligence. Manual-based causality assessment from a large number of free text-based documents is very time-consuming, labor-intensive, and sometimes even impractical. Herein, we proposed a general causal inference framework named DeepCausality to empirically estimate the causal factors for suspected endpoints embedded in the free text. The proposed DeepCausality seamlessly incorporates AI-powered language models, named entity recognition and Judea Pearl's Do-calculus, into a general framework for causal inference to fulfill different domain-specific applications. We exemplified the utility of the proposed DeepCausality framework by employing the LiverTox database to estimate idiosyncratic drug-induced liver injury (DILI)-related causal terms and generate a knowledge-based causal tree for idiosyncratic DILI patient stratification. Consequently, the DeepCausality yielded a prediction performance with an accuracy of 0.92 and an F-score of 0.84 for the DILI prediction. Notably, 90% of causal terms enriched by the DeepCausality were consistent with the clinical causal terms defined by the American College of Gastroenterology (ACG) clinical guideline for evaluating suspected idiosyncratic DILI (iDILI). Furthermore, we observed a high concordance of 0.91 between the iDILI severity scores generated by DeepCausality and domain experts. Altogether, the proposed DeepCausality framework could be a promising solution for causality assessment from free text and is publicly available through https://github.com/XingqiaoWang/https-github.com-XingqiaoWang-DeepCausality-LiverTox.

KEYWORDS

AI, causal inference analysis, transformer, NLP, DILI

Introduction

Causality is the study of the relationship between causes and effects, which is the foundation of almost every scientific discipline to verify hypotheses and uncover underlying mechanisms (Pearl, 2009). Notably, causal inference plays an essential role in medical practices to test scientific theories and decipher the etiology for advancing pharmacovigilance, optimize clinical trial designs, and establish real-world evidence (Naidu, 2013; Mazhar et al., 2020; Zheng et al., 2020; Ho et al., 2021). The conventional way to conduct causal inference relies on randomized controlled trials (RCTs) (Zheng et al., 2020). In randomized clinical trials, the test subjects are randomly assigned to one of two groups: the treated group receiving the intervention (e.g., drug) tested and the control group receiving an alternative (e.g., placebo) treatment. Causality is established if the clinical outcome is statistically significant in the treated group over the control one. However, conducting a randomized clinical trial is time-consuming, labor-intensive, expensive, and sometimes even impractical.

Consequently, there has been growing interest in alternative approaches, such as target trials based on observational data, to improve the causality assessment in real-world applications (Frieden, 2017; Gajra et al., 2020; Hernán, 2021). For example, the U.S. Food and Drug Administration (FDA) released guidance on a real-world evidence (RWE) program to create a framework for evaluating the potential use of RWE to help support the approval of a new indication for a drug already approved under section 505(c) of the FD&C Act or to help support or satisfy drug post-approval study requirements (https://www.fda.gov/science-research/science-and-research-special-topics/real-world-evidence). Under the 21st Century Cures Act, the FDA is mandated to evaluate the potential use of real-world data (RWD) and RWE to support the approval of a new indication for a drug. Draft guidance has been issued to address the generation of RWE, including the utilization of claims and electronic health records (EHRs), two major RWD sources, in support of regulatory decision-making. In addition, the FDA has prioritized the creation of an RWE Data Enterprise (the Sentinel System). An essential part of the initiative is incorporating EHR data from about 10 million individuals into the data infrastructure for FDA active drug safety surveillance (https://www.fda.gov/news-events/fda-voices/fda-budget-matters-cross-cutting-data-enterprise-real-world-evidence).

In the past decade, the generation of EHRs has increased substantially in the U.S., partly due to the Health Information Technology for Economic and Clinical Health (HITECH) Act of 2009, which provided $30 billion in incentives for hospitals and physician practices to adopt EHR systems. Whereas administrative claims data are highly structured, much of the potentially useful information contained within EHRs is unstructured, in the form of laboratory data, visit notes (e.g., narrative descriptions of a patient's signs and symptoms, family history, social history), radiology reports or images, and discharge summaries. EHRs contain rich clinical information and complex relations in the data that may not be fully harnessed using more traditional approaches. The ability of EHRs to generate quality RWE depends on whether we can address the challenge in curating and analyzing unstructured data. In response, FDA seeks to incorporate emerging data science innovations, such as natural language processing (NLP) and machine learning, to establish the organizational framework for ensuring high-fidelity, fit-for-purpose EHR data. To inform the causal inference framework for EHR-based signal detection (hypothesis generating), we will evaluate the emerging approaches that have been proposed or tested.

Accumulated observational data provide tremendous opportunities to promote target trials for causality establishment. Thus, there is an urgent need to develop novel statistical models to effectively estimate causal factors embedded in the extensive free text-based observational data. Artificial intelligence (AI) has made substantial progress in a variety of fields, such as computer vision (O'Mahony et al., 2019), NLP (Liu et al., 2021), speech recognition and generation (Hannun et al., 2014), and decision-making (Shrestha et al., 2019). Despite significant progress in AI, we still face a great challenge in understanding the mechanisms underlying intelligence, including reasoning, planning, and imagination (Schölkopf, 2019). Recent hype of AI-powered language models (LMs) and advanced statistical measures seem to pave a promising way to enhance the ability of AI in reasoning, such as causal inference (Veitch et al., 2020; Wang et al., 2021). In our previous work, we proposed a transformer-based causal inference framework called InferBERT by integrating the A Lite Bidirectional Encoder Representations from Transformers (ALBERT) (Lan et al., 2019) and Judea Pearl's Do-calculus (Wang et al., 2021). The proposed InferBERT has been successfully applied for causality assessment in pharmacovigilance and exemplified estimation of the causal factors related to opioid-related acute liver failure and tramadol-related mortalities in the FDA Adverse Event Reporting System (FAERS) database. However, there is still much space for improvement for InferBERT to facilitate real-world applications. First, the proposed InferBERT has only been used for structure-based data sets (e.g., FAERS), limiting its application in the free text-based corpus. Although we proposed a synthetic approach to transforming the different clinical entities into a sentence-based representation, the performance of the proposed InferBERT in free text needs to be further investigated. Second, domain-specific knowledge was not considered for causal inference, resulting in false positives or introduction of Irrelevant causal factors.

In this study, we proposed a general AI-powered framework called DeepCausality by fusing transformer, named entity recognition (NER), and Judea Pearl's Do-calculus for causal inference from free text-based documents. To demonstrate

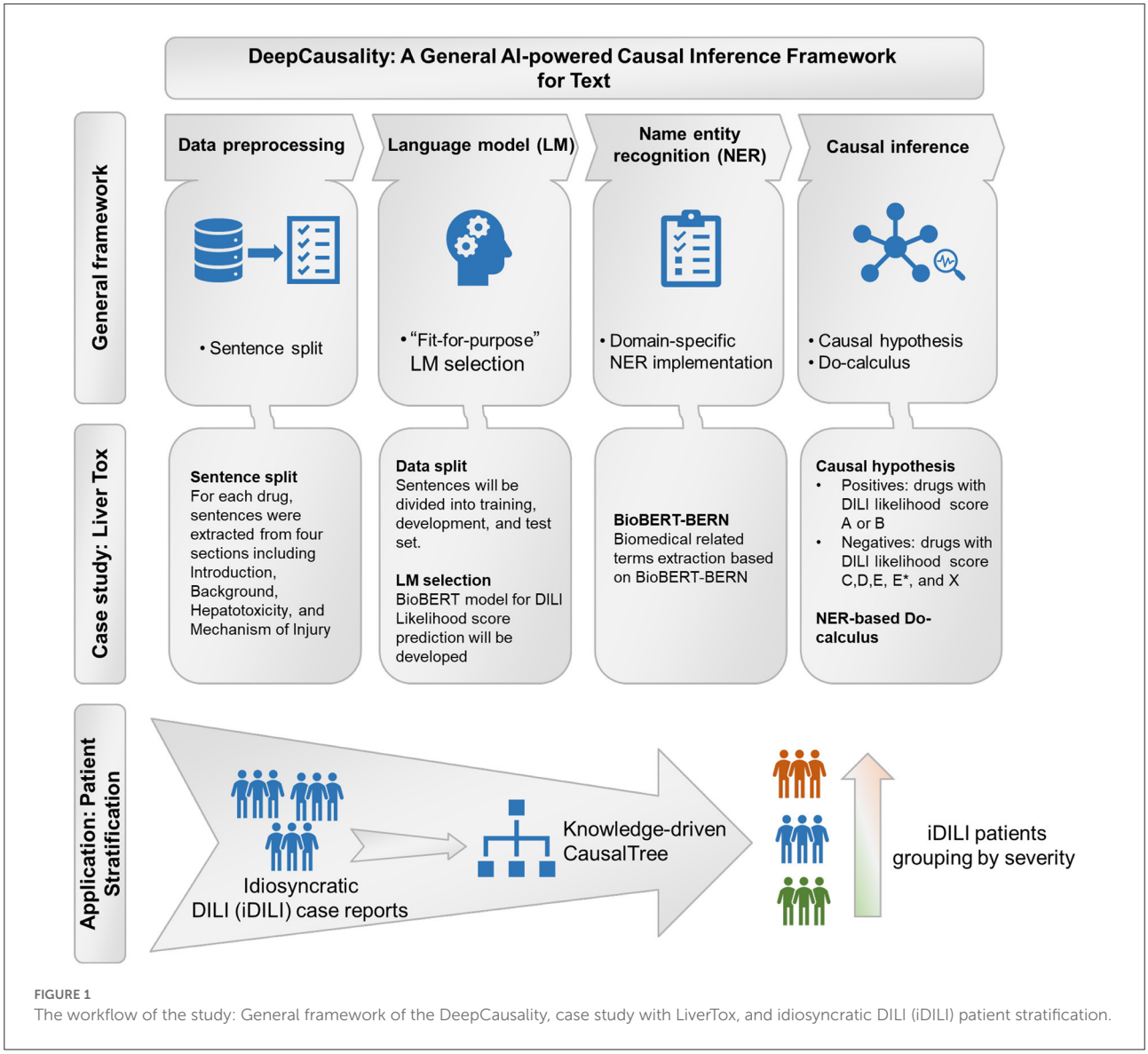


FIGURE 1
The workflow of the study: General framework of the DeepCausality, case study with LiverTox, and idiosyncratic DILI (iDILI) patient stratification.

the validity of the proposed DeepCausality, we employed the LiverTox database (https://www.ncbi.nlm.nih.gov/books/NBK547852/) to estimate the drug-induced liver injury (DILI)-related causal terms and further verified by using the American College of Gastroenterology (ACG) clinical guideline for idiosyncratic DILI (iDILI) (Chalasani et al., 2021). Furthermore, we developed a causal tree based on verified causal DILI terms and utilized it for iDILI patient stratification based on DILI case reports.

Materials and methods

DeepCausality overview

The proposed DeepCausality is a general transformer-based causal inference framework for free text, consisting of data preprocessing, LM development, NER, and Do-calculus based causal inference (Figure 1).

Data preprocessing

First, the corpus of free text-based documents was split into sentences. Then, an endpoint was assigned to each sentence based on the investigational causal question. For example, suppose you investigate causal factors of lung cancer etiology. The sentences describing the patient with lung cancer and related symptoms and clinical outcomes were labeled as positives, and vice versa. Consequently, we used $\boldsymbol{D}$ to denote the preprocessed corpus of free text-based documents, where $d_i = (x_i, y_i) \in \boldsymbol{D}$ indicates the i-th instance in the dataset $\boldsymbol{D}$, $i = 1,2,\ldots,N$, N (total number of instances), with x_i (i.e., sentence) and y_i (i.e., endpoint) being the text sequence. We employed

tf-idf [i.e., term frequency (tf)-inverse document frequency (idf)] values to investigate the distribution of terms in the corpus, which could be calculated based on the below formula,

$$tf - idf\left(t,d\right) = tf\left(t,d\right)^{*} idf\left(t\right) \quad (1)$$

$$tf\left(t,d\right) = \frac{count\ of\ t\ in\ d}{number\ of\ words\ in\ d} \quad (2)$$

$$idf\left(t\right) = log(N/(df+1)) \quad (3)$$

where t, d, N denote term, documents, and number of documents, respectively. The higher *tf-idf* value signified its importance in the document and corpus.

Language model development

Conditional probability distribution among words (i.e., tokens) in the text corpus is the basis for causal inference. LM uses various statistical and probabilistic techniques to determine joint probability among the words in the corpus. Specifically, a transformer-based LM could generate all joint probability among tokens as a gigantic probabilistic model using the Masked-Language Modeling (MLM) training strategy, allowing casual assessment among all the variables in the corpus. Two major types of transformer-based LM architectures, Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018) and its derives (Lan et al., 2019; Liu et al., 2019; Sanh et al., 2019; Clark et al., 2020), and Generative Pre-trained Transformer (GPT) models (Brown et al., 2020), currently dominate the field. Furthermore, efforts have also been made to develop transformers based on the domain-specific corpus [e.g., BioBERT (Lee et al., 2020), ClinicalBERT (Huang and Altosaar, 2019), SciBERT (Beltagy and Lo, 2019), LEGAL-BERT (Chalkidis et al., 2020)] for performance enhancement in specialized domains. Some reports have demonstrated that domain-specific pre-training is a solid foundation for a wide range of downstream domain-specialized NLP tasks (Gu et al., 2021).

With the pre-trained LM, the conditional probability distribution given free text is estimated by the LM-based downstream task. The pre-trained LM computes the attention between tokens. Then, the classification ([CLS]) special token representing the semantic information of the whole sequence is fed into the input layer of the downstream classification model. The softMax layer is adopted as the output layer to access the conditional probability distribution. We use the following cross entropy loss function for the classification of input text sequences:

$$LOSS(D) = -\sum_{1}^{N}\left(y_i * log\left(p\left(x_i\right)\right) + \left(1-y_i\right) * log(1-p(x_i))\right),\ i = 1,2,\ldots,\ N \quad (4)$$

where $p(x_i)$ is the output of the classification model for text sequence x_i, which is a calculated probability of the predicted class of x_i. y_i is the ground truth label of x_i.

By training the classifier with dataset $\boldsymbol{D}$, we can estimate the conditional probability distribution $\boldsymbol{P}(endpoint|\boldsymbol{X})$, where training dataset $\boldsymbol{X} = \{x_1, x_2, \ldots, x_N\}$. Then, we use the model to predict all the text sequences for each instance in the dataset $\boldsymbol{D}$. We denote the output of the classifier as $p(x_i)$, where $p(x_i)$ is the probability of the endpoint presented for instance d_i.

Name entity recognition

According to the task field, our framework adopts a domain-specific NER method, a text mining technique, to extract the name entities in the free text. The NER method can predict the span and category of name entities in the text according to the task with a domain-specific NER method.

For each instance, d_i in dataset $\boldsymbol{D}$, the NER method recognizes all the name entities in the text sequence x_i. Then, we get the set of name entities $\boldsymbol{ner}_i$ corresponding to x_i, where $\boldsymbol{ner}_i = \{ner_{i1}, ner_i2, \ldots, ner_{iM}\}$, with M being the total number of name entities in the text sequence x_i. Next, we combined and unified all the name entities in set $\boldsymbol{ner_i}$corresponding to the text in the extracted dataset $\boldsymbol{D}$. As a result, we obtained the unique name entity set $\boldsymbol{NER}$, where $NER = \bigcup ner_{\mathbf{i}},\ i = 1,2,\ldots,N$; It is the union of $\boldsymbol{ner_i}$. Then, the recognized name entities were fed into the Do-calculus component of the framework as causal factor candidates.

Do-calculus based causal inference

In our previous work, we performed causal inference on structured data by using the Do-calculus mechanism to check whether each feature in the structured data was the cause of the endpoint. In this study, to perform causal inference on the free text, we first extracted name entities in free text and then considered these name entities as causality candidates to infer potential causal factors.

In the proposed framework, the classifier model calculates the conditional probability distribution of the endpoint given the free text sequence. Then, the extracted name entities in each instance sequence act as the endpoint's candidate causal factors. To empirically estimate the candidate name entities in each instance causing the endpoint, we adopted Judea Pearl's Do-calculus framework (Tucci, 2013; Pearl and Mackenzie, 2018).

Do-calculus aims to investigate the interventional conditional probability distribution of $\boldsymbol{P}[endpoint = \text{true}|\mathbf{DO}(ner)]$ by counterfactually changing the appearance of the name entity *ner*. We use the conditional probability distribution expectation to represent the $\mathbf{DO}(ner)$ and $\mathbf{NOT\ DO}(ner)$. Suppose there exists a statistically significant difference when comparing the interventional conditional probability distributions of

P[endpoint = true|**DO**(ner)] and P[endpoint = true|**NOT DO**(ner)]. In that case, the causality relationship will be established.

Based on the Classification Prediction $p(x_i)$ from the developed classifier, the Do-calculus procedure was performed to estimate the cause of the endpoint. The pseudo-code of the name entity-based Do-calculus procedure is shown below:

Input: Classification Prediction result $p(x)$, dataset ***D***, ***NER*** results, statistic test threshold *thr*
Output: Do-calculus results ***C***

1. set ***C*** = {} // ***C*** is the set of established causes
2. **for** *ner* in ***NER*** **do** // for each name entity
3. set ***S1*** = {} // *S1* contains all results of **DO** (*ner*)
4. set ***S2*** = {} // *S2* contains all results of **NOT DO** (*ner*)
5. **for** d_i in ***D*** **do** // for each instance in the dataset
6. ***S1*** ← p(*endpoint*|**DO** (*ner*) // probability of **DO** (*ner*).
7. ***S2***← p(*endpoint*|**NOT DO** (*ner*) // probability of **NOT DO** (*ner*).
8. z-score = z_{test} (***S1***, ***S2***) // perform z-test based on ***S1*** and ***S2***
9. **if** z-score > *thr* **then**
10. ***C***←*ner* // ***C*** consists of all established causes
11. **return** ***C***;

Algorithm 1. Name entity-based Do-calculus algorithm.

For all the extracted name entities, we applied the name entity-based Do-calculus algorithm to check whether it was the cause of the endpoint. For a name entity *ner*, if *ner* ϵ x_i, we say instance d_i meets the condition of **DO** (*ner*), while if *ner* $\notin$ x_i, then it doesn't. For *ner,* we assigned the conditional probability p[*endpoint*|**DO** (*ner*)] or p[*endpoint*|**NOT DO** (*ner*)] to sets ***S1*** and ***S2*** respectively. ***S1*** is the set of conditional probability of **DO** (*ner*), while ***S2*** consists of conditional probabilities of those instances NOT DO (*ner*). We used the one tail z-test to evaluate whether the probabilities in ***S1*** were significantly different to ***S2***.

We perform one tail z-test between S1 and S2. If the p-value is less than a threshold like 0.05, we view the *ner* as a cause of the endpoint. To establish all the causal terms of the endpoint, we evaluated every candidate name entity. The generated term set ***C*** is the set of all the name entities that satisfy the statistical significance test.

Case study: Causal inference of idiosyncratic DILI based on LiverTox

Clinical knowledge of idiosyncratic DILI

iDILI is a rare adverse drug reaction, but common in gastroenterology and hepatology practices. The symptoms of iDILI have multiple presentations, characterized from asymptomatic elevations in liver biochemistries to hepatocellular or cholestatic jaundice, liver failure, or chronic hepatitis (Chalasani et al., 2021). Causal factors associated with iDILI recommended by ACG Clinical Guideline could be divided into three types: host, environmental, and drug-related factors (Chalasani et al., 2021). Specifically, host factors include age, gender, pregnancy, malnutrition, obesity, diabetes mellitus, co-morbidities (e.g., underlying liver disease), and indications for therapy. Environmental factors include smoking, alcohol consumption, infection, and inflammatory episodes. Drug-related factors consist of the daily dose, metabolic profiles, class effect and cross-sensitization, and drug interactions and polypharmacy. Furthermore, the ACG clinical guideline also suggested an algorithm to evaluate suspected iDILI by integrating DILI-related clinical measurements and iDILI-associated causal factors (Chalasani et al., 2021).

Data preprocessing of the LiverTox database

LiverTox®, launched by the National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK) and the National Library of Medicine (NLM), is a DILI atlas dedicated to providing up-to-date, easily accessed information and comprehensive clinical information on iDILI for both physicians and patients (Hoofnagle, 2013). There are 1,095 drug records in the LiverTox database, which are available at https://ftp.ncbi.nlm.nih.gov/pub/litarch/29/31/. For each drug record, the information was organized based on different sections, including Introduction, Background, Hepatotoxicity, Mechanism of Liver Injury, Outcome and Management, Case reports, Chemical and Product Information, and References.

To demonstrate the utility of the proposed DeepCausality framework, we employed drug records stored in the LiverTox® database. The purpose is to use our proposed DeepCausality to estimate the causal factors related to iDILI. For each drug record, we extracted the text from four sections, Introduction, Background, Hepatotoxicity, and Mechanism of Liver Injury, which are the major sections that describe the synthesized knowledge on hepatoxicity. The DILI Likelihood score is embedded in the hepatoxicity section. Each sentence except the one that included the DILI likelihood score in these four sections was considered as x_i, and all the extracted sentences were considered as ***D.***

Domain experts developed the DILI likelihood score to categorize drugs based on the likelihood of drugs associated with the known potential of DILI for causing liver injury. The

DILI likelihood score is largely opinion-based and derived from published medical literature to categorize the possibility of the drug causing idiosyncratic liver injury, including **Category A** – well known, **Category B** – known or highly likely, **Category C** – probable, **Category D** – possible, **Category E** – not believed or unlikely, and **Category X** – unknown. We labeled each sentence x_i according to the DILI likelihood score. Specifically, if the sentence x_i from the drug with a DILI likelihood score was either Category A or Category B, we assigned the sentence a label y_i as iDILI positives. Otherwise, the sentence was labeled as iDILI negatives.

Language model selection

Considering the LiverTox database provided the summarized knowledge on DILI mainly based on medical literature, we selected BioBERT as the domain-specific language model to develop DeepCausality. BioBERT was developed on top of the pre-trained BERT model by further fine-tuning with biomedical-specific corpora, including PubMed abstracts (PubMed) and PubMed Central full-text articles (PMC) using MLM (Lee et al., 2019). BioBERT has shown its superiority in various biomedical-related downstream tasks over the state-of-the-art NLP approaches. To make BioBERT more specific for the DILI application, we further fine-tuned the BioBERT model with the extracted sentences $\boldsymbol{D}$ from LiverTox. Consequently, the fine-tuned BioBERT could represent the joint conditional probability among words involved in the extracted sentence $\boldsymbol{D}$.

Biomedical entity recognition

Given that many words in the corpus were not biomedical specific, there was the potential risk of bringing false positives during the causal inference process. Therefore, we employed biomedical entity recognition to extract different biomedical-related terms and limit the causal inference within these domain-relevant terms. In this study, we used biomedical entity recognition and a multi-type normalization tool (BERN) to extract biomedical-related terms, including gene/protein, disease, drug/chemical, species information, and genetic variants (Kim et al., 2019). The BERN is a series of BioBERT-named entity recognition models with probability-based decision rules to recognize and discover different biomedical entities, accessible through https://bern.korea.ac.kr. Here, we only considered extracted name entities with more than a frequency of 50 across the corpus as causal factor candidates for further analysis.

NER-based Causal inference

The named entity-based Do-calculus strategy was developed to carry out the causal inference within biomedical entities extracted using the BERN. The potential causal terms of iDILI were enriched if the adjusted p value was less than 0.05 based on the one-tail z-test calculation. Furthermore, other statistical measures were also provided, including z-score, average DO probability, and average not DO probability.

We further developed a knowledge-based causal tree to organize the enriched causal factors by following the ACG clinical guideline for iDILI diagnosis (see *Clinical knowledge* section). Specifically, the enriched causal terms were classified into different causal factors of iDILI, including Concomitant diseases, History of other liver disorders, Physical findings, Laboratory results, Symptoms and Signs, Clinical outcome, Covering host, Environmental, and Drug-related. Furthermore, the liver enzymes test results were also incorporated into the proposed knowledge-based causal tree to facilitate the iDILI patient stratification.

Real-world application: Idiosyncratic DILI (iDILI) patient stratification

In the LiverTox database, some drug records contained one or more case reports related to DILI, which were curated from scientific literature or liver-specific clinical databases such as DILI Network (DILIN). The case report comprised the findings from a clinical laboratory, radiologic and histologic testing summarized in a formulaic table titled Key Points, and a short concluding discussion and comment on DILI severity. The key points included iDILI patterns and severity scores, which served as the ground truth for iDILI patient classification. The DILI patterns were divided into three categories (i.e., Hepatocellular - $R > 5$, mixed - $2<R<5$, and cholestatic - $R < 2$) by the ratio between serum alanine transaminase (ALT) and aspartate transaminase (AST). The severity score was based on five levels: 1+, Mild; 2+, Moderate; 3+, Moderate to Severe; 4+, Severe; and 5+, Fatal.

Because iDILI is a multifactorial endpoint caused by different underlying mechanisms, it was crucial to stratify iDILI patients into different DILI pattern subgroups to facilitate subsequent treatment regimen development. To demonstrate whether the developed knowledge-based causal tree could be utilized to categorize the iDILI patients, we extracted a total of 175 case reports from LiverTox for further analysis. First, we classified the patients by extracting the causal factors involved in the developed knowledge-based causal tree from each case report. Second, we verified the iDILI patient stratification results by comparing them to the ground truth classification results based on the DILI pattern and severity scores.

Robustness evaluation

The proposed DeepCausality framework employed transformer-based LMs to learn the joint probability among

variables for causal inference. However, this process can be less robust due to different random seeds, even though the same hyper-parameters were chosen. Toward real-world applications, the robustness of the proposed framework was investigated based on the strategy developed in our previous study (Wang et al., 2021). Specifically, we employed the proposed DeepCausality to run parallel experiments with the same hypermeters three times. Then, the enriched causal terms in the three repeated experiments were compared using a Venn diagram and the percentage of overlapped terms (POT) strategy (Wang et al., 2021). The POT could be calculated based on two steps: (1) rank the enriched terms based on z scores from high to low in each run, and (2) calculate the POT using the number of the overlapping terms among three repeated runs divided by *L*. *L* denotes the number of enriched terms of each subset of the ranked enriched term list. In this study, *L* was set from 1 to 30 at one interval.

Implementation of the DeepCausality

To facilitate the application of our model, we developed a standalone package for the readers' convenience. The proposed DeepCausality framework was exemplified based on a BioBERT (BioBERT, https://github.com/dmis-lab/biobert) and BERN under Python 3.6 TensorFlow version 1.15. We evaluated our proposed DeepCausality model on one NVIDIA Tesla V100 GPU. For the LiverTox dataset, the average runtime was approximately 8 h. We incorporated the Do-calculus causal function into the BioBERT source code, which easily migrated into other transformers. All the source code and the processed data sets used in this study are publicly available through https://github.com/XingqiaoWang/https-github.com-XingqiaoWang-DeepCausality-LiverTox.

Results

Data preprocessing of the LiverTox dataset

Figure 2 illustrates the sequence length of the extracted 14,361 sentences from four sections (i.e., Introduction, Background, Hepatotoxicity, Mechanism of Liver Injury) of LiverTox. The average and standard deviation of the sequence length of the extracted 14,361 sentences is 26.84 ± 15.58. Furthermore, the extracted 14,361 sentences contain 15,804 unique words (Supplementary Table S1). We observed the top ten terms based on the term frequency-inverse document frequency (Tf-idf) values, including *iu*, *hydroxycut*, *clobazam*, *dabrafenib*, *dapsone*, *germander*, *progesterone*, *asparaginase*, *barbiturate*, and *CDC*. These top ten terms were not directly associated with any current knowledge of iDILI, indicating the causal factors could not be enriched by the simple frequency-based strategy.

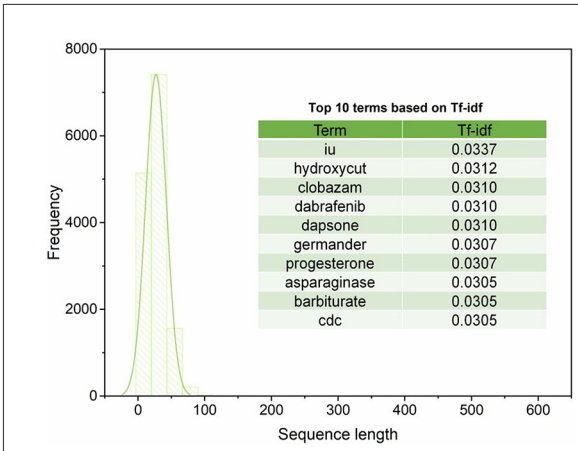


FIGURE 2
The distribution of sequence length and the top 10 terms based on Tf-idf values.

TABLE 1 Data information of preprocessed sentences in LiverTox.

Dataset	iDILI positive	iDILI negative	Positive ratio	Total
Training set	3,218	9,706	0.249	12,924
Test set	360	1,077	0.251	1,437
Total	3,578	10,783	0.249	14,361

Fine-tune BioBERT model with LiverTox data

Considering that LiverTox is summarized from literature and clinical reports, we employed BioBERT to establish the joint probability between variables. For that, we divided the extracted 14,361 sentences into two sets with a ratio of 9:1 in a stratified manner, with the ratio between positives (i.e., iDILI positives) and negatives (i.e., iDILI negatives) kept constant for both sets. It resulted in 12,924 (14,361 × 90% = 12,924) and 1,437 (14,361 × 10% = 1,437) sentences in training and test sets, respectively (Table 1). Then, we employed BioBERT-Base v1.1 (+ PubMed 1M), consisting of 12 transformer layers, 128 embeddings, 768 hidden, and 12 heads with 11M parameters. We further fine-tuned the $\text{BioBERT}_{\text{base}}$ model with the 12,924 sentences in the training set. We determined the optimized models based on the text classification result in the test set for iDILI sentence prediction. Specifically, we set the maximum sequence length to 128 and the mini-batch size to 128. A total of

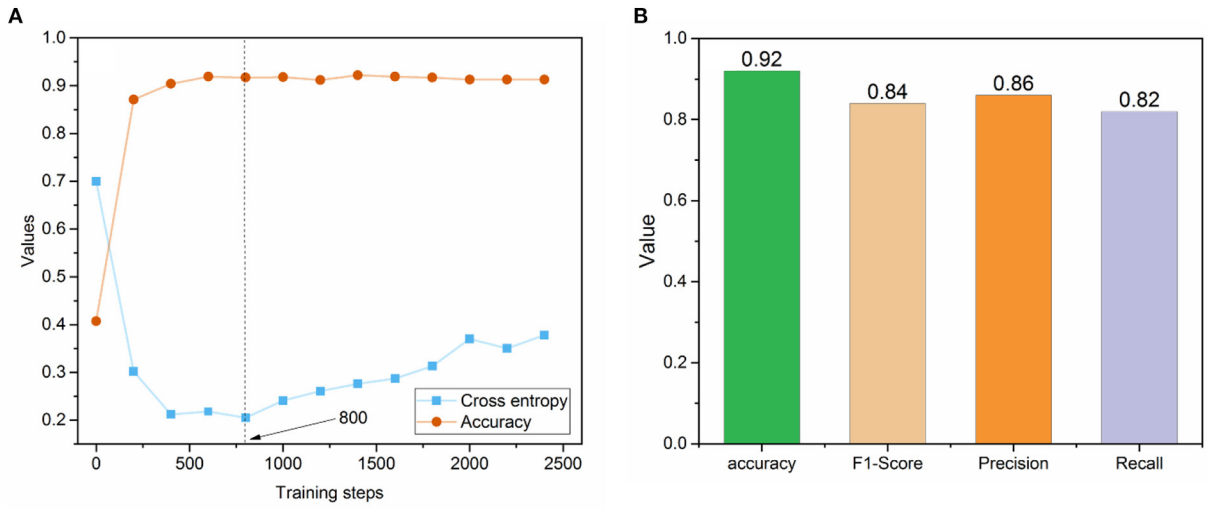


FIGURE 3
(A) The trend of cross-entropy loss and accuracy across the different training steps in the fine-tuned BioBERT model; **(B)** Prediction performance metrics of the optimized fine-tuned BioBERT model on the test set.

2,500 training steps were implemented with a 500-step warmup, and the checkpoint step was set to 200 for recording the prediction results.

Figure 3A depicted the trends of cross entropy loss and accuracy while increasing the number of training steps based on the text set. The cross-entropy loss decreased dramatically before 400 training steps and became stable between 400 and 800 training steps. Then, it increased after 1,000 steps, indicating the potential of overfitting phenomena. Meanwhile, the accuracies of the dataset tended to be stable after training step 400. Thus, we selected the optimized fine-tuned model based on the training step with the minimum loss (i.e., 800), where the accuracy value also showed no dramatic changes. The optimized fine-tuned model yielded a high accuracy of 0.92, an F1-score of 0.84, a precision of 0.86, and a recall of 0.82 in the test set, indicating the optimized fine-tuned model well captured the relationship between variables (Figure 3B).

Biomedical-based named entity recognition

To carry out the causal inference within the biomedical-based NER terms, we employed the BERN to extract the biomedical-related terms from the preprocessed sentences. We obtained a total of 87 biomedical-related terms that were divided into three categories using BERN, including 16 drugs, 11 genes, and 60 diseases (see Supplementary Table S2). Through the biomedical-based NER, we narrowed down the total terms (unique words) in the preprocessed sentences from a total of 15,804 to 87, with a 99.4% compression rate.

Causal inference using NER-based Do-calculus

To further investigate whether the performance of the proposed DeepCausality could identify the causal terms of iDILI, we implemented NER-based Do-calculus to uncover the predictors from the fine-tuned BioBERT model (Table 2). Of 87 Biomedical-based name entities, 24 name entities were enriched with an adjusted p value < 0.05 based on a one-tail z-test using the NER-based Do-calculus. We excluded 4 drug entities, including iron, isoniazid, rifampin, and acetaminophen, since our objective is to identify the causal factors related to iDILI. For example, acetaminophen is a protype drug for dose-dependent drug-induced liver injury (DILI), which is not idiosyncratic in nature (Jaeschke, 2015). Furthermore, 18 of 20 enriched causal terms were highly consistent with current knowledge of iDILI, yielding an enrichment rate of 90% (Chalasani et al., 2021). These name entities were distributed into different categories, including Liver Enzymes, Concomitant diseases, History of other liver disorders, Physical findings, Laboratory results, Symptoms and Signs, and Clinical outcomes based on the ACG clinical guideline for iDILI diagnosis.

Table 2 lists enriched causal factors ranked based on the Z score. The causal factor (Z score) are as follows: for Liver Enzymes, alkaline phosphatase (3.772), ALT (2.561); for Concomitant diseases, tuberculosis (2.470), rheumatoid

TABLE 2 Causal inference results for idiosyncratic DILI.

Elements	Z score	Probability of DO value	Probability of not DO value	Probability difference
Liver Enzymes				
Alkaline phosphatase	3.772	0.398	0.244	0.154
ALT	2.561	0.307	0.244	0.064
Concomitant diseases				
Tuberculosis	2.470	0.382	0.244	0.138
Rheumatoid arthritis	1.759	0.334	0.244	0.089
History of other liver disorder				
Cholestasis	4.827	0.547	0.244	0.303
Cholestatic hepatitis	3.653	0.499	0.244	0.255
Physical findings				
Fever	6.508	0.383	0.241	0.141
Pain	2.377	0.395	0.244	0.150
Laboratory results				
Lactic acidosis	3.181	0.460	0.244	0.216
Symptoms and signs				
Hypersensitivity	3.966	0.383	0.243	0.139
Skin rash	2.066	0.333	0.244	0.088
Jaundice	1.773	0.274	0.244	0.030
Stevens Johnson syndrome	1.669	0.335	0.245	0.090
Clinical outcome				
Hepatic failure	4.119	0.437	0.244	0.193
Cirrhosis	2.944	0.391	0.244	0.147
Liver failure	2.905	0.366	0.244	0.122
Sinusoidal obstruction syndrome	2.490	0.403	0.244	0.159
Acute liver failure	1.669	0.326	0.244	0.082

arthritis (1.759); for History of other liver disorder, cholestasis (4.827), cholestatic hepatitis (3.653); for Physical findings, fever (6.508), pain (2.377); for Laboratory results, lactic acidosis (3.181); for Symptoms and Signs, hypersensitivity (3.966), skin rash (2.066), jaundice (1.773), and Stevens-Johnson syndrome (1.669); for Clinical outcome, hepatic failure (4.119), cirrhosis (2.944), liver failure (2.905), sinusoidal obstruction syndrome (2.490), and acute liver failure (1.669).

Figure 4 illustrates the developed knowledge-based causal tree with enriched causal factors based on the ACG clinical guideline for iDILI diagnosis. The proposed knowledge-based prediction tree could be divided into two major components: liver enzyme test and clinical observations. The liver enzyme test, including ALT and AST, divides iDILI patients into different DILI patterns, including hepatocellular, mixed, and cholestatic. Clinical observations could further classify the iDILI patients based on their severity and clinical symptoms.

iDILI patient stratification

To demonstrate the proposed knowledge-based causal tree could be utilized for iDILI patient stratification, we stratified 175 patients' case reports in the LiverTox dataset based on the developed causal tree and compared expert-based patient stratification results. There was a high correlation between the R (ALT/AST) values determined by DeepCausality and the experts, with a Pearson correlation coefficient of more than 0.9 (Figure 5). Furthermore, we observed that the clinical observations in the developed causal tree could be used to classify the patients into different severity groups, distinguished by the R scores estimated by DeepCausality (Figure 6).

Robustness of DeepCausality

To ensure the proposed DeepCausality could generate reproducible causal inference results, we investigated the robustness of causal inference results by running the DeepCausality three times (see Supplementary Table S3). Figure 7 depicted the POT enrichment after three different runs. We found highly reproducible results from three parallel runs of DeepCausality, with an average POT of 0.923. Furthermore, the Venn diagram indicates 87.5% commonality of enriched causal terms after three runs. Altogether, the proposed DeepCausality framework could generate highly repeatable results without interfering with factors such as initial seeds.

Discussion

Causality is one of the most critical notions in every branch of science. Causal inference based on observational data has gained more and more momentum as an alternative to the conventional random controlled trial-based causality assessment. Notably, More and more advocates promote using RWD and RWE to monitor post-market safety and adverse events and make regulatory decisions in drug development. An essential resource of RWD, observational data such as EHRs, clinical reports, and patient narratives are typically free text-based, posing a significant challenge to uncovering hidden causal factors. AI-powered LMs such as transformers

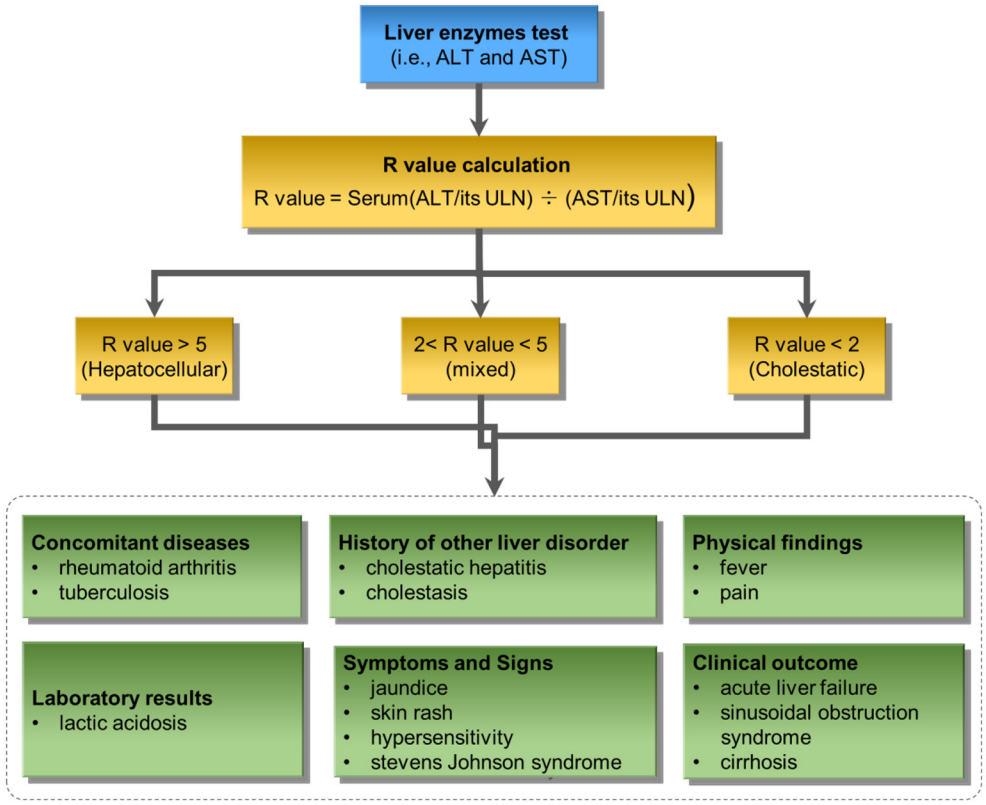


FIGURE 4
The proposed knowledge-based causal tree based on the ACG clinical guideline on iDILI patient diagnosis: ULN denotes upper limits of normal.

have shown great potential in various NLP tasks such as text classification, information retrieval, question & answering, and sentimental analysis. However, leveraging these AI-powered LMs to conduct causal inference as a human does is still at the infant stage. To bridge this gap, we proposed DeepCausality, a general AI-powered causal inference framework for free text. We exemplified the utility of the proposed DeepCausality for iDILI-related causal factor identification based on LiverTox and applied it to iDILI patient stratification. Consequently, DeepCausality identified 20 causal factors for iDILI, and 18 (90%) were aligned with the current clinical knowledge of iDILI. Furthermore, the developed knowledge-based causal tree was used to classify iDILI patients, which was highly consistent with stratification results based on domain experts.

AI-based language models such as transformers rely on a pre-trained model with a large corpus and then use the learned knowledge to solve the downstream tasks. In this study, without training on a large number of DILI-related literature and clinical reports, we hypothesized the accumulated knowledge from these large corpora of documents could be an alternative to accelerate the training process of transformer-based LMs. Furthermore, we introduced the domain-specific named entity recognition (NER) step into the general framework, aiming to eliminate the false positives and irrelevant enrichment in the causal inference process. If

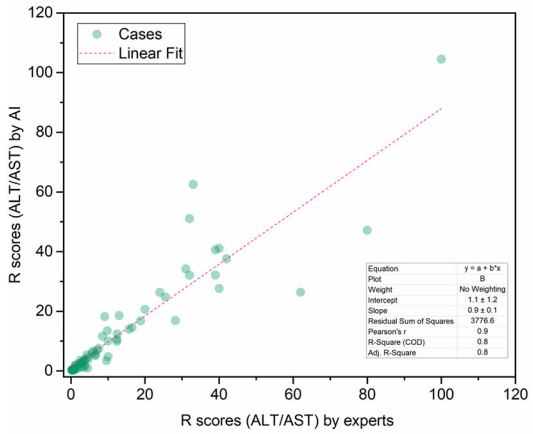


FIGURE 5
The correction between the R scores (ALT/AST) calculated by DeepCausality and expert: ALT and AST stand for Alanine transaminase and aspartate transaminase, respectively.

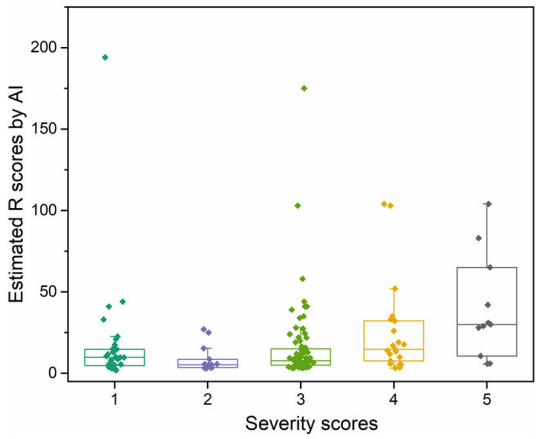


FIGURE 6
The distribution of iDILI patients stratified by DeepCausality across the different severity levels defined by domain experts.

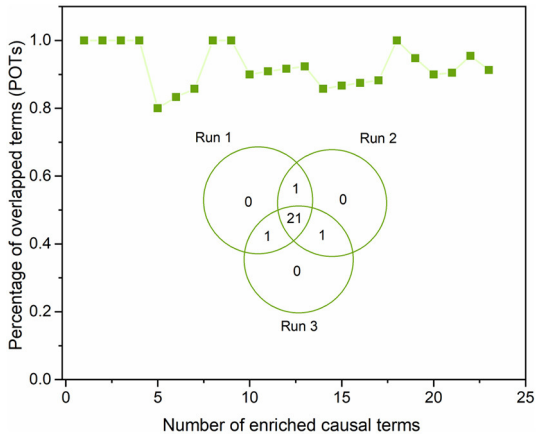


FIGURE 7
Robustness evaluation of the proposed DeepCausality: The Venn diagram illustrates the overlapping of the enriched causal terms by three parallel runs. The dotted-line curve illustrates the percentage of overlapping causal terms (POTs) among the three repeated runs across ranked order terms by z scores.

available, this step could also be substituted with domain-specific ontology and knowledge graphs.

One of the initial attempts conveyed in this study was to use the developed knowledge-based causal tree for iDILI patient stratification. The high consistency of iDILI patient stratification results from DeepCausality with determination by experts is encouraging. However, it is worth pointing out the causal tree was developed based on prior knowledge of iDILI diagnosis, indicating that expert knowledge is still an indispensable component to facilitating AI-based approaches in real-world applications.

It is also worth investigating a few aspects of the proposed DeepCausality for potential improvements. In this study, to showcase the proposed DeepCausality, we employed a biomedical-based free text in LiverTox. Additional validation of the utility in other domains is highly recommended. To facilitate the process, all developed codes, scripts, and processed datasets are open to the public through https://github.com/XingqiaoWang/https-github.com-XingqiaoWang-DeepCausality-LiverTox. Additionally, the BERT-based model was incorporated into the DeepCausality framework presented here. Some generative-based transformers, such as Generative Pre-trained Transformer 3 (GPT3), do not need intensive task-specific training (Brown et al., 2020), which may be a more efficient way to conduct causal inference. Lastly, although DeepCausality could identify the causal factors, it could not classify the identified causal factors further into cofounders or colliders. It may be solved by developing directional DO-calculus statistics in the Bayesian networks derived from the transformers.

In conclusion, DeepCausality provided an AI-powered solution for causal inference in free text by integrating transformers, NER, and Do-calculus into a unified framework. DeepCausality is proposed for real-world applications to promote RWE collection and utilization.

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

XX devised the DeepCausality model applied in this study. ZL and WT conceived and designed the study and the utilization of the LiverTox database. XW coded the DeepCausality model. XW and ZL performed data analysis. ZL, XW, and XX wrote the manuscript. WT and QL revised the manuscript. All authors read and approved the final manuscript.

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.

Author disclaimer

This manuscript reflects the views of the authors and does not necessarily reflect those of the Food and Drug Administration. Any mention of commercial products is for clarification only and is not intended as approval, endorsement, or recommendation.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frai.2022.999289/full#supplementary-material

References

Beltagy, I., and Lo, K. (2019). SciBERT: A pretrained language model for scientific text. *arXiv [Preprint]*. arXiv:1903.10676. doi: 10.18653/v1/D19-1371

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., et al. (2020). Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* 33, 1877–1901. doi: 10.48550/arXiv.2005.14165

Chalasani, N. P., Maddur, H., Russo, M. W., and Wong, R. J. (2021). ACG clinical guideline: diagnosis and management of idiosyncratic drug-induced liver injury. *ACG* 116, 878–898. doi: 10.14309/ajg.0000000000001259

Chalkidis, I., Fergadiotis, M., Malakasiotis, P., and Aletras, N. (2020). LEGAL-BERT: The muppets straight out of law school. *arXiv [Preprint]*. arXiv:2010.02559.

Clark, K., Luong, M. T., and Le, Q. V. (2020). Electra: Pre-training text encoders as discriminators rather than generators. *arXiv [Preprint]*. arXiv:2003.10555.

Devlin, J., Chang, M. W., and Lee, K. (2018). Bert: Pretraining of deep bidirectional transformers for language understanding. *arXiv [Preprint]*. arXiv:1810.04805.

Frieden, T. R. (2017). Evidence for health decision making — beyond randomized, controlled trials. *New Engl. J. Med.* 377, 465–475. doi: 10.1056/NEJMra1614394

Gajra, A., Zettler, M. E., and Feinberg, B. A. (2020). Randomization versus Real-World Evidence. *New England J. Med.* 383, e21. doi: 10.1056/NEJMc2020020

Gu, Y., Tinn, R., Cheng, H., Lucas, M., Usuyama, N., Liu, X., et al. (2021). Domain-specific language model pretraining for biomedical natural language processing. *ACM Trans. Comput. Healthcare (HEALTH)* 3, 1–23. doi: 10.1145/3458754

Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., et al. (2014). Deep speech: Scaling up end-to-end speech recognition. *arXiv [Preprint]*. arXiv:1412.5567.

Hernán, M. A. (2021). Methods of public health research — strengthening causal inference from observational data. *New Engl. J. Med.* 385, 1345–1348. doi: 10.1056/NEJMp2113319

Ho, M., van der Laan, M., Lee, H., Chen, J., Lee, K., Fang, Y., et al. (2021). The current landscape in biostatistics of real-world data and evidence: causal inference frameworks for study design and analysis. *Stat. Biopharmaceut. Res.* 52, 511–525. doi: 10.1080/19466315.2021.1883475

Hoofnagle, J. H. (2013). LiverTox: a website on drug-induced liver injury," in *Drug-Induced Liver Disease* (Elsevier) 725–732. doi: 10.1016/B978-0-12-387817-5.00040-6

Huang, K., and Altosaar, J. (2019). Clinicalbert: Modeling clinical notes and predicting hospital readmission. *arXiv [Preprint]*. arXiv:1904.05342.

Jaeschke, H. (2015). Acetaminophen: Dose-dependent drug hepatotoxicity and acute liver failure in patients. *Dig. Dis.* 33, 464–471. doi: 10.1159/000374090

Kim, D., Lee, J., So, C. H., Jeon, H., Jeong, M., Choi, Y., et al. (2019). A neural named entity recognition and multi-type normalization tool for biomedical text mining. *IEEE Access* 7, 73729–73740. doi: 10.1109/ACCESS.2019.2920708

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R., et al. (2019). Albert: A lite bert for self-supervised learning of language representations. *arXiv [Preprint]*. arXiv:1909.11942.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., et al. (2019). BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* 36, 1234–1240.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., et al. (2020). BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* 36, 1234–1240. doi: 10.1093/bioinformatics/btz682

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., et al. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv [Preprint]*. arXiv:1907.11692.

Liu, Z., Roberts, R. A., Lal-Nag, M., Chen, X., Huang, R., Tong, W., et al. (2021). AI-based language models powering drug discovery and development. *Drug Discov. Today* 26, 2593–2607. doi: 10.1016/j.drudis.2021.06.009

Mazhar, H., Foster, B. C., Necyk, C., Gardiner, P. M., Harris, C. S., Robaey, P., et al. (2020). Natural health product-drug interaction causality assessment in pediatric adverse event reports associated with attention-deficit/hyperactivity disorder medication. *J. Child Adolesc. Psychopharmacol.* 30, 38–47. doi: 10.1089/cap.2019.0102

Naidu, R. P. (2013). Causality assessment: A brief insight into practices in pharmaceutical industry. *Perspect. Clin. Res.* 4, 233–236. doi: 10.4103/2229-3485.120173

O'Mahony, N., Campbell, S., Carvalho, A., Harapanahalli, S., Hernandez, G. V., Krpalkova, L., et al. (2019). Deep learning vs. traditional computer vision," in *Science and Information Conference* (Springer) 128–144. doi: 10.1007/978-3-030-17795-9_10

Pearl, J. (2009). *Causality: Models, Reasoning and Inference*. Cambridge: Cambridge University Press. doi: 10.1017/CBO9780511803161

Pearl, J., and Mackenzie, D. (2018). *The Book of Why: The New Science of Cause and Effect*. Penguin: Basic Books.

Sanh, V., Debut, L., and Chaumond, J. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv [Preprint]*. arXiv:1910.01108.

Schölkopf, B. (2019). Causality for machine learning. *arXiv [Preprint]*. arXiv:1911.10500.

Shrestha, Y. R., Ben-Menahem, S. M., and Von Krogh, G. (2019). Organizational decision-making structures in the age of artificial intelligence. *California Manag. Rev.* 61, 66–83. doi: 10.1177/0008125619862257

Tucci, R. R. (2013). Introduction to Judea Pearl's Do-Calculus. *arXiv [Preprint]*. arXiv:1305.5506.

Veitch, V., Sridhar, D., and Blei, D. (2020). "Adapting text embeddings for causal inference," in *Conference on Uncertainty in Artificial Intelligence, PMLR*, 919–928.

Wang, X., Xu, X., Tong, W., Roberts, R., and Liu, Z. (2021). InferBERT: A transformer-based causal inference framework for enhancing pharmacovigilance. *Front. Artific. Intell.* 4, 659622. doi: 10.3389/frai.2021.659622

Zheng, C., Dai, R., Gale, R. P., and Zhang, M. J. (2020). Causal inference in randomized clinical trials. *Bone Marrow Transpl.* 55, 4–8. doi: 10.1038/s41409-018-0424-x

Frontiers in
Artificial Intelligence

Explores the disruptive technological revolution of AI

A nexus for research in core and applied AI areas, this journal focuses on the enormous expansion of AI into aspects of modern life such as finance, law, medicine, agriculture, and human learning.

Discover the latest Research Topics

See more →

Frontiers

Avenue du Tribunal-Fédéral 34
1005 Lausanne, Switzerland
frontiersin.org

Contact us

+41 (0)21 510 17 00
frontiersin.org/about/contact