frontiers | Research Topics

Methods in computational genomics

Edited by
Nathan Olson and Lei Chen

Published in
Frontiers in Genetics

ISSN 1664-8714
ISBN 978-2-8325-4423-5
DOI 10.3389/978-2-8325-4423-5

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

Methods in computational genomics

Topic editors

Nathan Olson — National Institute of Standards and Technology (NIST), United States

Lei Chen — Shanghai Maritime University, China

Citation

Olson, N., Chen, L., eds. (2024). *Methods in computational genomics*. Lausanne: Frontiers Media SA. doi: 10.3389/978-2-8325-4423-5

Table of contents

frontiers | Frontiers in Genetics

TYPE Editorial
PUBLISHED 25 January 2024
DOI 10.3389/fgene.2024.1367531

OPEN ACCESS

EDITED AND REVIEWED BY
Richard D. Emes,
Nottingham Trent University, United Kingdom

*CORRESPONDENCE
Nathan D. Olson,
nathanael.olson@nist.gov

RECEIVED 08 January 2024
ACCEPTED 16 January 2024
PUBLISHED 25 January 2024

CITATION
Chen L and Olson ND (2024), Editorial: Methods in computational genomics.
Front. Genet. 15:1367531.
doi: 10.3389/fgene.2024.1367531

Editorial: Methods in computational genomics

Lei Chen[1] and Nathan D. Olson[2]*

[1]College of Information Engineering, Shanghai Maritime University, Shanghai, China, [2]National Institute of Standards and Technology (NIST), Gaithersburg, MD, United States

KEYWORDS

genomics, artificial intelligence, machine learning, computational tool, software, bioinformatics

Editorial on the Research Topic
Methods in computational genomics

In the rapidly evolving field of Methods in Computational Genomics, this editorial series illuminates the forefront of experimental techniques and methodologies. From dissecting large multidimensional numeric datasets to predicting the functions of novel genomic entities, these approaches have revolutionized our understanding of genomic data. This editorial underscores two pivotal themes: the development of innovative tools and software, and the integration of artificial intelligence (AI) and machine learning in genomic research. These themes exemplify the significant impact computational methods have had on genomics, providing novel insights into complex biological questions.

Theme 1—Tools and Software. Tools and software are fundamental to computational genomics. This section underscores the broader impact of these innovations in advancing genomic research, particularly in metagenomics and gene expression analysis.

Melzer et al.- *CLARITY App*. This application, developed for high-resolution genetic mapping, exemplifies the integration of computational tools with traditional genomics. Its capacity to interconnect physical and genetic maps and visualize recombination hotspots illustrates how software can significantly enhance genomic research, making complex data more accessible and interpretable.

Munro et al.- *Real-time ONT Sequence Analysis Pipeline*. Their tool addresses the need for efficient pathogen monitoring. By optimizing sequencing time and costs, this pipeline demonstrates the practical benefits of computational tools in real-time sequence analysis, which is crucial for epidemiological surveillance.

Liang et al.–ARGem*: Antimicrobial Resistant Pipeline.* This user-friendly pipeline for profiling antibiotic resistance genes reflects the growing importance of metagenomics in environmental monitoring. Its high performance in analyzing aquatic metagenomes highlights the tool's flexibility and utility in diverse research contexts.

Alves et al.- *EasySSR*. This web tool for microsatellite analysis streamlines genomic comparisons, catering to the need for simple yet effective tools in genomic research. EasySSR's functionality, providing outputs like PTT files and interactive charts, demonstrates how computational tools can facilitate complex genomic analyses.

Each manuscript in this theme showcases how computational tools and software are not just auxiliary but integral to genomic research, offering innovative solutions to traditional challenges and opening new avenues for exploration and discovery in the field.

Theme 2–AI/Machine Learning. This theme highlights the transformative impact of AI and machine learning in computational genomics, showcasing a range of algorithms from statistical learning to advanced deep learning.

Sub-theme 1–Genomic Analysis. Genomic analysis is critical in computational genomics for its ability to unravel complex biological mechanisms.

Ju et al.—*DNA N4-methylcytosine (4mC) Analysis*. The use of deep learning models for predicting 4mC sites, as demonstrated in the brief research report, exemplifies the potential of these advanced techniques in enhancing our understanding of gene regulation and genome stability.

Jia et al.—*Enhancer Prediction*. Jia et al.'s iEnhancer-DCSV method, which employs densely connected convolutional networks, showcases how AI can be leveraged to predict enhancers, thus contributing to the understanding of gene transcription and expression.

Zhang et al.—*CTCF Binding Sites Analysis*. Zhang et al.'s machine learning model for predicting chromatin loop anchors from CTCF binding sites underscores the role of AI in dissecting complex genomic structures.

Deng et al.—*Osteoporosis Genomic Research*. Deng et al.'s investigation into osteoporosis demonstrates the power of bioinformatics in identifying immune-related genetic markers, highlighting the intersection of AI and medical genomics.

Sub-theme 2–Protein/Peptide Analysis. Protein/peptide analysis is essential for understanding complex biological functions and disease mechanisms.

Su et al.—*Outer Membrane Protein Prediction*. Su et al.'s computational model for predicting outer membrane proteins illustrates the application of AI in protein analysis, enhancing our understanding of cellular structures.

Liu et al.—*Neuropeptide Prediction*. Liu et al.'s ensemble tool, integrating multiple convolution neural network models, demonstrates the effectiveness of AI in predicting biologically significant peptides.

Wang et al.—*Antimicrobial Peptide Prediction*. Wang et al.'s deep learning strategy for predicting antimicrobial peptides represents a significant advance in therapeutic research, offering potential applications in treating conditions like diabetic foot.

Livesey et al.- *Cancer Genomic Analysis*. Livesey et al.'s approach to kidney renal clear cell carcinoma employs AI for gene analysis, showcasing how these techniques can lead to meaningful insights in cancer research.

Throughout this theme, the diverse range of AI/ML methods and their specific applications in different genomic contexts highlight the vast potential of these technologies in pushing the frontiers of genomic research.

The transformative impact of these methods is particularly evident in precision medicine. The perspective by Latapiat et al., focusing on individualized co-expression networks, is a testament to this evolution. Their approach to patient stratification in complex diseases underscores the real-world implications of computational genomics, enhancing diagnostics and treatment personalization. This aligns with the ongoing need for tool and software development aimed at optimizing data generation and information extraction, furthering our understanding of biological systems.

This series has showcased remarkable advancements in computational genomics, exhibiting the synergy between innovative software tools, AI, and machine learning techniques. These manuscripts demonstrate how both cutting-edge and established algorithms contribute to the field's robustness and innovation. From the efficient mapping of genetic landscapes with tools like CLARITY to the sophisticated prediction of neuropeptides using ensemble AI models, these studies exemplify the diverse range of applications in genomics.

As we conclude, the integration of novel computational methodologies with traditional approaches is not just enhancing genomic research but is pivotal in deciphering the complexities of life sciences. The future of genomics, rich with potential, is set to be driven by these innovative computational strategies.

Author contributions

LC: Writing–original draft, Writing–review and editing. NO: Writing–original draft, Writing–review and editing.

Funding

The author(s) declare that no financial support was received for the research, authorship, and/or publication of this article.

Acknowledgments

We are grateful for all the authors who contributed to the Research Topic and those who volunteered their time to review the manuscripts. We would like to thank Megan Cleveland and Sierra Miller for their feedback on the manuscript. These opinions, recommendations, findings, and conclusions do not necessarily reflect the views or policies of NIST or the United States Government. Certain equipment, instruments, software, or materials are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement of any product or service by NIST, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

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.

The author(s) declared that they were an editorial board member of Frontiers, at the time of submission. This had no impact on the peer review process and the final decision.

Publisher's note

frontiers | Frontiers in Genetics

TYPE Original Research
PUBLISHED 01 March 2023
DOI 10.3389/fgene.2023.1132018

OPEN ACCESS

EDITED BY
Lei Chen,
Shanghai Maritime University, China

REVIEWED BY
Guoxian Yu,
Shandong University, China
Qi Dai,
Zhejiang Sci-Tech University, China
Hongtao Lu,
Shanghai Jiao Tong University, China

*CORRESPONDENCE
Jianhua Jia,
jjh163yx@163.com
Rufeng Lei,
rufeng_lei@163.com

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

RECEIVED 26 December 2022
ACCEPTED 13 February 2023
PUBLISHED 01 March 2023

CITATION
Jia J, Lei R, Qin L, Wu G and Wei X (2023), iEnhancer-DCSV: Predicting enhancers and their strength based on DenseNet and improved convolutional block attention module.
Front. Genet. 14:1132018.
doi: 10.3389/fgene.2023.1132018

iEnhancer-DCSV: Predicting enhancers and their strength based on DenseNet and improved convolutional block attention module

Jianhua Jia[1]*, Rufeng Lei[1]*, Lulu Qin[1], Genqiang Wu[1] and Xin Wei[2]

[1]School of Information Engineering, Jingdezhen Ceramic University, Jingdezhen, China, [2]Business School, Jiangxi Institute of Fashion Technology, Nanchang, China

Enhancers play a crucial role in controlling gene transcription and expression. Therefore, bioinformatics puts many emphases on predicting enhancers and their strength. It is vital to create quick and accurate calculating techniques because conventional biomedical tests take too long time and are too expensive. This paper proposed a new predictor called iEnhancer-DCSV built on a modified densely connected convolutional network (DenseNet) and an improved convolutional block attention module (CBAM). Coding was performed using one-hot and nucleotide chemical property (NCP). DenseNet was used to extract advanced features from raw coding. The channel attention and spatial attention modules were used to evaluate the significance of the advanced features and then input into a fully connected neural network to yield the prediction probabilities. Finally, ensemble learning was employed on the final categorization findings *via* voting. According to the experimental results on the test set, the first layer of enhancer recognition achieved an accuracy of 78.95%, and the Matthews correlation coefficient value was 0.5809. The second layer of enhancer strength prediction achieved an accuracy of 80.70%, and the Matthews correlation coefficient value was 0.6609. The iEnhancer-DCSV method can be found at https://github.com/leirufeng/iEnhancer-DCSV. It is easy to obtain the desired results without using the complex mathematical formulas involved.

KEYWORDS

enhancer, DenseNet, channel attention, spatial attention, ensemble learning

1 Introduction

Genes are functional areas of an organism's DNA (Dai et al., 2018; Kong et al., 2020) that hold genetic information. The gene is transferred to the protein through a sequence of transcription (Maston et al., 2006) and translation (Xiao et al., 2016), and proteins control the organism's exterior phenotypic shape (Buccitelli and Selbach, 2020). Transcription is one of the most crucial aspects of gene expression. The enhancer and promoter (Cvetesic and Lenhard, 2017) are the most significant sequence regions for transcriptional activity. An enhancer is a brief non-coding DNA fragment on DNA (Kim et al., 2010) and controls rapid and slow gene expression (Shrinivas et al., 2019). According to previous studies, several illnesses (Yang et al., 2022) are produced as a result of enhancer mutations and deletions (Emison et al., 2005; Liu G. et al., 2018; Boyd et al., 2018; Wu et al., 2019). In terms of the

activities they express, the enhancers may be categorized into groups, such as strong and weak enhancers, closed (balanced) enhancers, and latent enhancers (Shlyueva et al., 2014). Therefore, understanding and recognizing these specific gene sequence segments is an urgent problem (Pennacchio et al., 2013).

Traditional medical experimental methods (Yang et al., 2020) in bioinformatics are costly and time-consuming. Therefore, it is crucial to develop computational techniques and derive some excellent predictors (Firpi et al., 2010; Fernández and Miranda-Saavedra, 2012; Erwin et al., 2014; Ghandi et al., 2014; Kleftogiannis et al., 2015; Lu et al., 2015; Bu et al., 2017; Yang et al., 2017). However, these techniques have limitations in the prediction of strong and weak enhancers. Liu et al. (2015) developed a predictor called iEnhancer-2L based on the support vector machine (SVM) algorithm and used the sequence pseudo-K-tuple nucleotide composition (PseKNC) approach to encode features. Afterward, machine learning-based methods were applied to the prediction of enhancers, such as SVM (Jia and He, 2016; He and Jia, 2017), RF (Singh et al., 2013; Wang et al., 2021), and XGBoost (Cai et al., 2021), and many excellent predictors have been created. However, a single machine learning classifier has obvious performance drawbacks. A predictor based on an ensemble learning model (Liu B. et al., 2018) was developed to address this problem, which generally has a significantly better performance. The ensemble learning model has diversity and complexity in feature processing. For instance, Wang C. et al. (2022) developed a predictor called Enhancer-FRL, which used 10 feature methods for feature coding. The manual creation of feature coding is a relatively difficult problem, and the presence of many complex feature coding types can lead to dimensional disasters. Furthermore, the effectiveness of conventional machine learning models depends on the extracted complex features. Consequently, the development of a predictor that requires only simple features is crucial.

Nowadays, deep learning is becoming increasingly popular. Nguyen et al. (2019) proposed the iEnhancer-ECNN model based on convolutional neural networks (CNNs). Niu et al. (2021) proposed a model called the iEnhancer-EBLSTM based on bi-directional long short-term memory (Bi-LSTM). They used one-hot and K-mers coding techniques to encode the enhancer sequences and then fed these features into the deep learning network to get relatively good prediction results. For example, in the iEnhancer-ECNN model, the ACC and MCC of enhancer recognition results were 0.769 and 0.537, and the ACC and MCC of enhancer strength prediction results were 0.678 and 0.368, respectively. However, there is a wide gap in prediction precision using a better deep learning model.

In deep learning networks, CNNs with more convolutional layers extract more advanced local features but lead to the problem of gradient disappearance and network degradation. To solve this problem, the residual neural network (ResNet) (Li et al., 2022) uses a short-circuit connection structure, which allows the convolutional layers to be connected several layers apart and can solve the problem of network degradation to some extent. However, the densely connected convolutional network (DenseNet) (Huang et al., 2010) has been enhanced based on ResNet. DenseNet extracts richer feature information by reusing the features of each previous layer, and it is more effective than ResNet. The attention model is also increasingly used, and the essence of the attention model is to focus on more useful feature information and suppress useless feature information. Convolutional block attention module (CBAM) (Zhang et al., 2022) can focus on more useful feature information from channel and spatial dimensions. The current computational method has the disadvantages of poor performance and complex features. For this purpose, we developed a new predictor called iEnhancer-DCSV. The predictor is conducted using a modified DenseNet and an improved CBAM attention module. The DenseNet framework makes it easier to extract more advanced features. Experimental results show that our model outperforms the existing models. The iEnhancer-DCSV model is currently the optimal choice for predicting enhancers and their strengths.

TABLE 1 Specifics of the benchmark dataset.

Layer	Original dataset	Enhancer	Non-enhancer
First layer	Training dataset	1,484	1,484
	Testing dataset	200	200
	Original dataset	Strong enhancers	Weak enhancers
Second layer	Training dataset	742	742
	Testing dataset	100	100

2 Materials and methods

2.1 Benchmark dataset

The benchmark dataset was created by Liu et al. (2015). They took the enhancer fragments from nine cell lines, removed 80% of the redundant sequences with the CD-HIT (Huang et al., 2010) and then calculated the ideal fragment length of 200 bp for each enhancer sequence to create the final dataset. The dataset is split into two sections: a training dataset for the model's training and an independent test dataset for model testing. The independent test dataset is made up of 200 enhancer samples (with 100 strongly and 100 weakly enhancer samples) and 200 non-enhancer samples, whereas the training dataset is made up of 1,484 enhancer samples (with 742 strongly and 742 weakly enhancer samples) and 1,484 non-enhancer samples. All enhancer samples in the independent test dataset were different from the training dataset to guarantee that the samples are independent. The benchmark dataset is described in Table 1 and may be downloaded conveniently from the website: https://github.com/leirufeng/iEnhancer-DCSV.

2.2 Feature coding schemes

Two simple and effective coding techniques are used in this study: one-hot and NCP. Notably, these two coding techniques produce columns with a dimension of 200, so they can be feature-combined. For instance, an enhancer sequence with a length of 200 bp can obtain a 4 × 200 feature matrix and a 3 × 200 feature

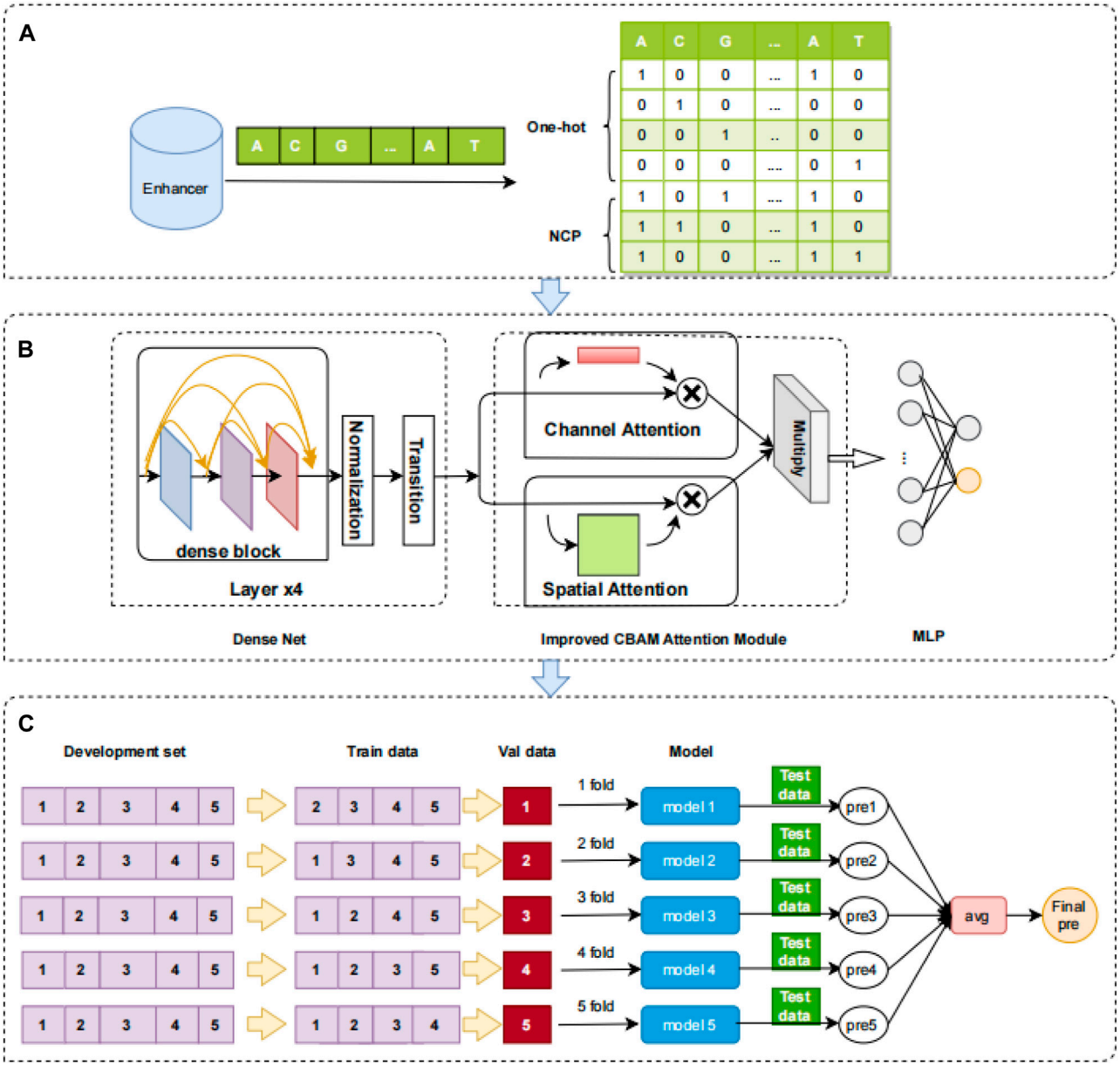


FIGURE 1
Overview of the iEnhancer-DCSV model. **(A)** Feature coding. One-hot and NCP are used to encode the enhancer sequence, and a 7 × 200 matrix is produced. **(B)** Framework of the iEnhancer-DCSV model. The original features are input directly to the modified DenseNet structure (which includes four dense blocks, normalized layers, and transition layers), and the improved structure is used to extract advanced features. Modules for spatial and channel attention are introduced to assess the extracted advanced features' importance. The two evaluated advanced feature maps are multiplied together at the corresponding positions. The fully connected neural network is used to output the prediction probabilities. **(C)** Ensemble model. The model uses fivefold cross-validation, where each fold is tested using an independent test set, each test enhancer sequence generates five prediction probabilities, and the final classification is voted using ensemble learning.

matrix after one-hot and NCP coding, respectively. Finally, combining these two matrices through feature fusion can yield a 7 × 200 feature matrix. In this study, the enhancer sequence is considered a gray image by the feature coding matrix. The 7 × 200 matrix is directly used as the original feature input.

2.2.1 One-hot coding

In the field of bioinformatics, one-hot coding is one of the most used coding techniques. The advantages of this coding technique are its feasibility, efficiency, and ability to assure that each nucleotide letter is coded independently. The method is effective in avoiding the expression of interdependencies. This coding technique is particularly popular in bioinformatics. The double helix structure (Sinden et al., 1998) of DNA is widely known, and it is made up of four nucleotides: A (adenine deoxyribonucleotide), C (cytosine deoxyribonucleotide), G (guanine deoxyribonucleotide), and T (thymine deoxyribonucleotide) (Chou, 1984). The enhancer sequences are DNA sequences designated "0,1,2,3" in the order

TABLE 2 Nucleotide chemical property.

Chemical property	Category	Nucleotide
Ring structure	Purine	A, G
	Pyrimidine	C, T
Functional group	Amino	A, C
	Keto	G, T
Hydrogen bonding	Strong	C, G
	Weak	A, T

"ACGT." The nucleotides in the sequences are then coded, and the coding length is four nucleotides. The coding elements are 0 and 1. The position corresponding to the nucleotide letter marker is coded as 1, and the other positions are coded as 0. For instance, "A" is coded as (1,0,0,0), "C" is coded as (0,1,0,0), "G" is coded as (0,0,1,0), and "T" is coded as (0,0,0,1) (Zhang et al., 2022). The one-hot coding is shown in Figure 1A.

2.2.2 NCP coding

The four DNA nucleotides are structurally different from each other and have different chemical molecular structures (Zhang et al., 2022). For instance, C and T contain one loop each, whereas A and G have two loops between the four nucleotides. G and T may be classified as ketone groups from the standpoint of chemical composition, whereas A and C can be classified as amino groups. A and T have two hydrogen bonds, but C and G have three hydrogen bonds. The strength between C and G is more powerful than that between A and T. The specific chemical properties between nucleotides are shown in Table 2.

Then, coding is performed based on the chemical characteristics. The nucleotide N_i is located in position i in the sequence. Three chemical characteristics of nucleotide N_i are "ring structure," "functional group," and "hydrogen bond strength" (Xiao et al., 2019). The vector representation of $N_i = (x_i, y_i, z_i)$, x_i, y_i, z_i is expressed as

$$\begin{cases} x_i = \begin{cases} 1, & if\ N_i \in \{A, G\}, \\ 0, & if\ N_i \in \{C, T\}, \end{cases} \\ y_i = \begin{cases} 1, & if\ N_i \in \{A, C\}, \\ 0, & if\ N_i \in \{G, T\}, \end{cases} \\ z_i = \begin{cases} 1, & if\ N_i \in \{A, T\}, \\ 0, & if\ N_i \in \{C, G\}. \end{cases} \end{cases} \tag{1}$$

A, C, G, and T may be encoded using this approach as (1,1,1), (0,1,0), (1,0,0), and (0,0,1). NCP coding is shown in Figure 1A.

2.3 Model construction

In this study, we constructed a network framework to automatically learn advanced features called iEnhancer-DCSV. The framework of iEnhancer-DCSV is divided into three parts: (A) feature coding, (B) framework of iEnhancer-DCSV model, and (C) ensemble model. The details are shown in Figure 1.

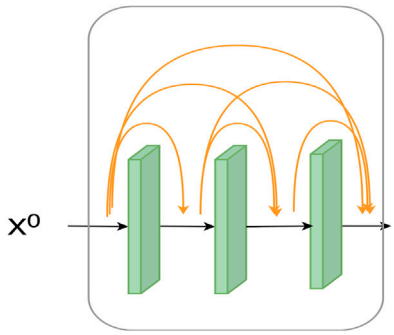


FIGURE 2
Structure of a dense block.

2.3.1 DenseNet

In this study, we modified the initial DenseNet structure. The original DenseNet consists of a convolutional layer, a dense block layer, and a transition layer. First, convolution is applied to the original features. Then, the convolution features are processed by the dense block and transition layers. The dense block layer is a dense connection of all the preceding layers to the following layers. In particular, each layer accepts all its preceding layers as its additional input, enabling feature reuse. The transition layer, which mainly connects two adjacent dense blocks, reduces the feature map size. Instead, we deleted the first convolutional layer and added a batch normalization layer between the dense block layer and the transition layer. This processing method can extract better-quality feature information and reduce the risk of overfitting.

2.3.1.1 Dense block

The traditional CNN network does not perform very well in extracting feature information. A convolutional structure called dense convolutional block extracts richer feature information by reusing previous features. Experimentally, the dense convolutional network feature extraction is proven better than traditional CNN. The structure diagram is shown in Figure 2.

In the dense block, the input of layer i is related to not only the output of layer $i-1$, but the output of all the previous layers. The X_l level is represented as follows:

$$X_l = H_l\left([X_0, X_1, X_2, \ldots, X_{l-1}]\right), \tag{2}$$

where is denoted as layers X_0 to X_{l-1} stitched together by the dimension of the channel. H is a non-linear combinatorial function. It is a combination of batch normalization, ReLU activation function, and convolution (3×3).

In this study, we used four dense blocks, each containing three layers of convolution. The final extraction of features was $X^{(seq)}$.

2.3.1.2 Transition layer

The l-1 layers in front of the dense block are combined by channel dimension. As the number of channels in the l-layer becomes larger, it leads to an explosion of parameters, along with a slow training speed. We can improve the efficiency by connecting a transition layer with the dense block layer. The transition layer

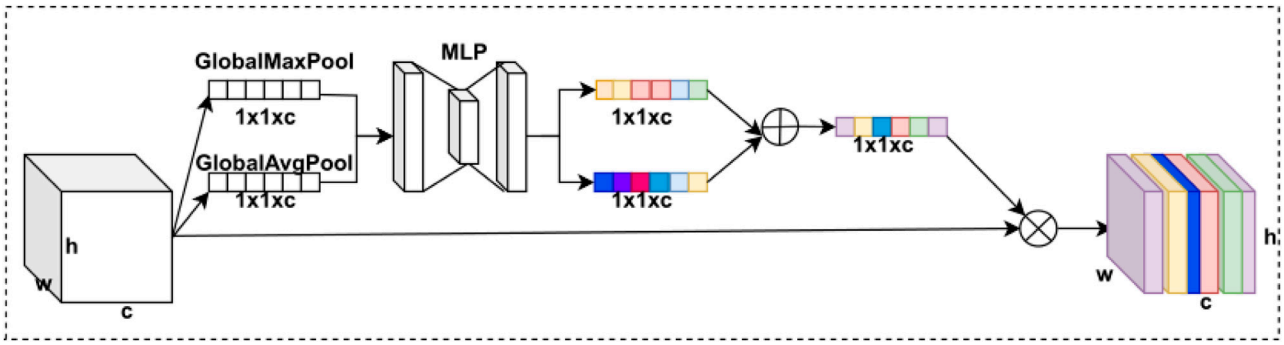


FIGURE 3
CBAM's channel attention module structure.

consists of a 1 × 1 convolution and a 2 × 2 average pooling. It is a function of reducing the number of channels and parameters in the dense block layer by downsampling to compress the model.

2.3.2 Batch normalization

Gradient explosion and gradient disappearance are serious problems in deep learning training, and this phenomenon tends to occur more likely in the deeper network structure. If the shallow parameters are changed, their fluctuations during backpropagation may be significant, resulting in significant variable shifts in the deeper network. Batch normalization (Min et al., 2016) has been shown to improve the generalization ability of the model. The batch normalization is expressed as follows:

$$\widetilde{x_i} = \frac{x_i - \mu}{\sqrt{\sigma^2 - \epsilon}}, \tag{3}$$

$$y_i = \gamma \widetilde{x_i} + \beta, \tag{4}$$

where A is the set of the feature dataset $[x_1, x_2, \ldots, x_i]$, μ is the mean of dataset A, and σ^2 is the variance of dataset A. γ and β are trainable parameters.

2.3.3 Improved CBAM attention module

The CBAM attention module comprises channel attention and spatial attention modules (Chen et al., 2017). First, we use the channel attention module to evaluate the original features. Second, we take the feature map output from the channel attention module and feed it back into the spatial attention module. Finally, we output the final feature maps from the spatial attention module. This serial connection of CBAM attention modules has the disadvantage that the attention modules are all computed in a specific way, and the computation of weights destroys the feature shape of the input. This leads to inaccurate weight calculation of the spatial attention modules and loss of channel weighting information in the final feature map. We change the original serial approach in the CBAM attention module to a parallel method. The principle is to input the original features into the channel attention module and the spatial attention module and let the output features be multiplied by their corresponding positions. By this method, the effect of each attention model after evaluation can be maximally preserved and the expressiveness of the features can be improved.

2.3.3.1 Channel attention module

In deep learning, the degree of importance varies between different feature map channels, so we use the channel attention module to calculate different weights for each channel. By weighting each channel of the feature map, the model automatically pays attention to the more useful channel information to achieve the fixation of channel dimension and compression of spatial dimension. The channel attention module comprises the max pooling layer, the average pooling layer, the MLP module, and the sigmoid activation function. The CBAM's channel attention module structure is shown in Figure 3.

The channel attention module starts with the feature map passing through two parallel max pooling and average pooling layers, which are input into the fully connected neural network (MLP) module separately. Second, the two results of the MLP output are summed element by element, and the channel attention module weights are obtained using the sigmoid activation function. Finally, these weights are multiplied by the feature map to obtain the feature map of the channel attention model weighting. The CBAM's channel attention model is expressed as follows:

$$M_c(F) = \sigma\left(MLP\left(AvgPool(F)\right) + MLP\left(MaxPool(F)\right)\right), \tag{5}$$

$$F = F_{scale}(F, Mc(F)) = Mc(F) \cdot F, \tag{6}$$

where pooling here is the global max pooling and the global average pooling. $F_{scale}(F, Mc(F))$ denotes each channel-specific value of F multiplied by the weight $Mc(F)$.

2.3.3.2 Spatial attention module

In deep learning, different receptive fields have different degrees of value to the feature map, so we use a spatial attention model to calculate the weights between receptive fields. By weighting the receptive fields, we allow the model to focus on the more useful target location information to achieve a constant spatial dimension and a compressed channel dimension. The spatial attention model is implemented through a max pooling layer, an average pooling layer, a CNN module, and a sigmoid activation function. The CBAM's spatial attention module structure is shown in Figure 4.

The spatial attention model first passes the feature maps through two parallel max pooling and average pooling layers and performs a

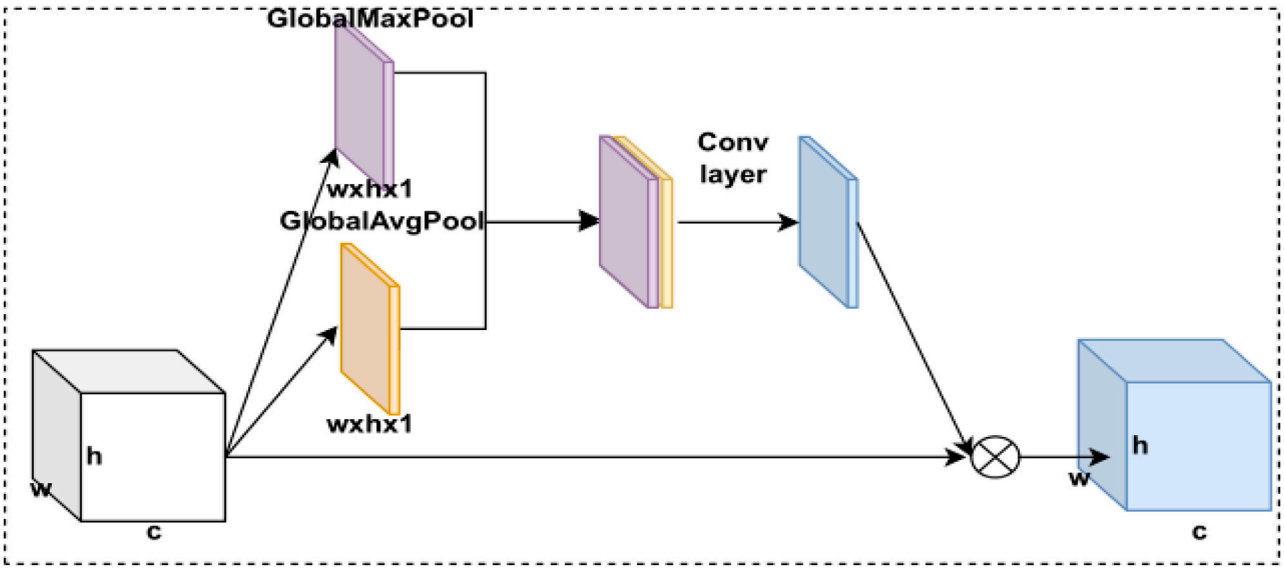


FIGURE 4
CBAM's spatial attention module structure.

stitching operation on the two pooling feature maps. Then, the newly obtained features are input into the CNN module to be transformed into a feature map with channel number 1, and the spatial attention module weights are obtained by the sigmoid activation function. Finally, this weight is multiplied by the feature map to obtain the weighted feature map of the spatial attention model. The CBAM's spatial attention model is expressed as follows:

$$M_s(F) = \sigma\left(f^{7\times7}\left(\left[AvgPool(F); MaxPool(F)\right]\right)\right), \quad (7)$$

$$F = F_{scale}(F, Ms(F)) = Ms(F) \cdot F. \quad (8)$$

Pooling here is the global max pooling and global average pooling. The size of the convolutional kernel used in the CNN module is 7 × 7. Finally, $F_{scale}(F, Mc(F))$ denotes each receptive field of F multiplied by the weight $Mc(F)$.

2.3.4 Fully connected neural network

We used a fully connected neural network (Wang. et al., 2022b) to predict the enhancers and their strength. After we extracted the advanced features, the size of the advanced features was reduced using a pooling layer. Then, these features are flattened into vectors, which are later input into the fully connected neural network. Finally, the softmax function is used to calculate the predicted probability of the enhancers. The softmax formula is expressed as

$$P\left(y = i|x\right) = \frac{e^{W_i^s * X}}{\sum_{j=1}^{C} e^{W_j^s * X}}, \quad (9)$$

where W_i^s and W_j^s denote the weights in the fully connected neural network, X denotes the sample, and C is the number of categories. $P(y = i|x)$ denotes the probability that x is predicted to be i. This is a dichotomous problem, $i = 0$ or $i = 1$.

2.3.5 Ensemble model

There is an ensemble method called bagging (Bauer and Kohavi, 1999). It is accomplished by training several different models, allowing independent test data to calculate the predicted results using different models and then averaging them. This ensemble learning approach is called model averaging. The advantage of model averaging is that different models do not usually produce the same error on the test data, and it is a very powerful method for reducing generalization errors.

In this study, we used a fivefold cross-validation method (Shang et al., 2022). The training dataset was divided into five parts: four for training and one for validation. We used an independent test set put into each fold in cross-validation, by which five predictions are obtained. Finally, the final prediction results are obtained by the voting method. The ensemble method is shown in Figure 1C.

2.4 Performance evaluation

Scientific evaluation metrics are a measure of model performance. In this study, the evaluation of model performance contains four metrics: sensitivity (Sn), specificity (Sp), accuracy (Acc), and Mathew's correlation coefficient (MCC) (Sokolova and Lapalme, 2009). The specific calculation formula is shown as follows:

$$\begin{cases} Sp = \dfrac{TN}{TN + FP}, \\ Sn = \dfrac{TP}{TP + FN}, \\ Acc = \dfrac{TP + TN}{TP + TN + FP + FN}, \\ MCC = \dfrac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}}, \end{cases} \quad (10)$$

TABLE 3 Comparison results of different coding schemes.

Layer	Coding	SN (%)	SP (%)	Acc (%)	MCC	AUC
First layer	One-hot	81.70	75.50	78.60	0.5737	0.8275
	NCP	83.25	70.50	76.88	0.5428	0.8168
	One-hot + NCP	80.25	**77.65**	**78.95**	**0.5809**	**0.8527**
Second layer	One-hot	60.30	72.80	66.55	0.3418	0.7491
	NCP	90.50	53.40	71.95	0.4780	0.7666
	One-hot + NCP	**99.10**	62.30	**80.70**	**0.6609**	**0.8686**

TABLE 4 Comparison with different architecture methods at layer 1 (enhancer recognition).

Model framework	SN (%)	SP (%)	Acc (%)	MCC	AUC
ResNet	69.80	77.90	73.85	0.4927	0.8211
DenseNet	83.10	68.50	75.80	0.5219	0.8108
DenseNet + channel attention	78.20	78.35	78.27	0.5686	0.8316
DenseNet + spatial attention	78.75	78.20	78.48	0.5717	0.8304
DenseNet + CBAM attention	83.70	67.25	75.48	0.5183	0.8046
DenseNet + improved CBAM attention	80.25	77.65	**78.95**	**0.5809**	**0.8527**

where TP, TN, FP, and FN are the four metrics in the confusion matrix, representing true positive, true negative, false positive, and false negative, respectively (Niu et al., 2021). In addition, we added the ROC curve area AUC metric (Vacic et al., 2006) to evaluate the model, and higher values of these metrics indicate better model performance.

3 Results and discussion

3.1 Construction of the first layer (enhancer recognition) model

The recognition of enhancers in the first layer is very important to complete the prediction mission. For the first layer of enhancer recognition, we used the iEnhancer-DCSV network framework. The advanced feature extraction and weight assignment are performed automatically by the model's iEnhancer-DCSV network framework. First, the enhancer sequences are encoded using the one-hot and NCP methods, and then feature coding is fed into the DenseNet to extract advanced features. These advanced features are input into the channel attention module and the spatial attention module, respectively. The two evaluated advanced feature maps are multiplied at the corresponding positions, and then the pooling layer is used to compress the feature size. Finally, a fully connected neural network is used to derive the predicted probabilities. We validate the model by putting independent test sets into each fold of the fivefold cross-validation. The aforementioned five-time results are passed through a soft voting mechanism to arrive at the final prediction. The whole process was cycled 10 times to verify the stability of the model, and the obtained individual performance metrics were averaged. The experimental results for SN, SP, Acc, and MCC were 80.25%, 77.65%, 78.95%, and 0.5809, respectively.

3.2 Construction of the second layer (strong and weak enhancer prediction) model

On the basis of the correct identification of enhancers in the first layer, the second layer predicts the strengths and weaknesses of enhancers. As the second layer has less training data and the complex network structure can lead to overfitting, we removed the attention module from the iEnhancer-DCSV network framework and used the same training as the first layer, with experimental results of 99.10%, 62.30%, 80.70%, and 0.6609 for SN, SP, Acc, and MCC, respectively.

3.3 Comparison of different coding methods

Currently, feature engineering has been a very important part of the process because building a model and using a simple and efficient coding method is crucial. In this study, we compared the one-hot + NCP coding, one-hot coding, and NCP coding to determine the final coding method. We input the three encoding methods into the two network frameworks, layer 1 and layer 2, respectively, and the results of the experiment are shown in Table 3. In the first layer (enhancer recognition), the one-hot + NCP coding was slightly better than the one-hot coding and better than the NCP coding. In the second layer (strong and weak enhancer prediction), the one-hot + NCP coding was much better than these two coding types. Therefore, we adopted one-hot + NCP coding as the final coding method in this study.

TABLE 5 Performance of iEnhancer-DCSV in 10 trials.

Layer	Cycle index	Sn (%)	Sp (%)	Acc (%)	MCC
First layer	0	78.50	78.00	78.25	0.5650
	1	83.00	75.50	79.25	0.5866
	2	80.50	75.00	77.75	0.5558
	3	74.00	85.50	79.75	0.5989
	4	77.00	81.50	79.25	0.5855
	5	87.50	68.00	77.75	0.5658
	6	80.50	80.00	80.25	0.6050
	7	79.50	80.00	79.75	0.5950
	8	80.50	78.00	79.25	0.5851
	9	81.50	75.00	78.25	0.5661
	Mean ± STD	**80.25 ± 3.39**	**77.65 ± 4.47**	**78.95 ± 0.84**	**0.5809 ± 0.0158**
Second layer	0	99.99	61.99	81.00	0.6702
	1	95.99	62.99	79.50	0.6250
	2	95.99	64.99	80.50	0.6416
	3	99.99	60.99	80.50	0.6624
	4	99.99	56.99	78.50	0.6313
	5	99.99	61.99	81.00	0.6702
	6	99.99	60.99	80.50	0.6624
	7	99.99	64.99	82.50	0.6938
	8	99.99	58.99	79.50	0.6468
	9	98.99	67.99	83.50	0.7047
	Mean ± STD	**99.10 ± 1.58**	**62.30 ± 3.00**	**80.70 ± 1.38**	**0.6609 ± 0.0243**

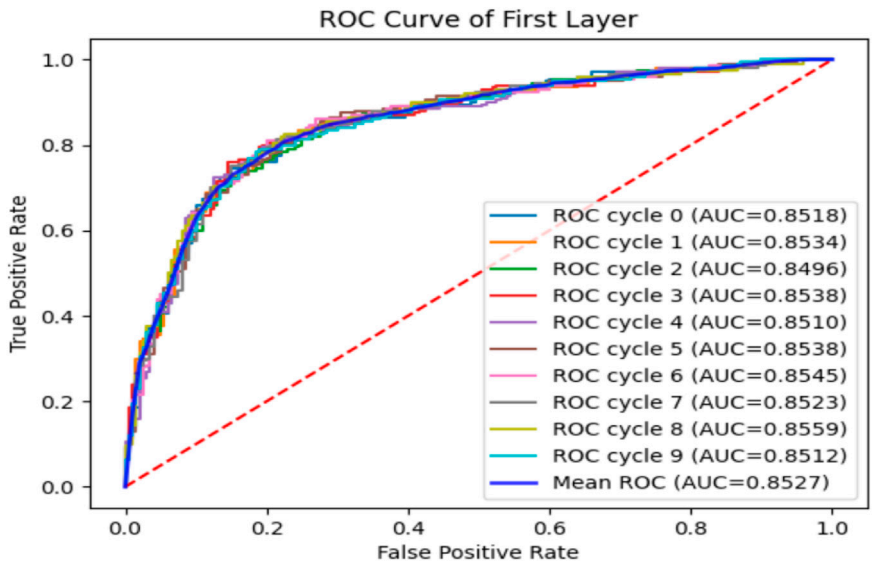


FIGURE 5
ROC curves for layer 1 (enhancer recognition).

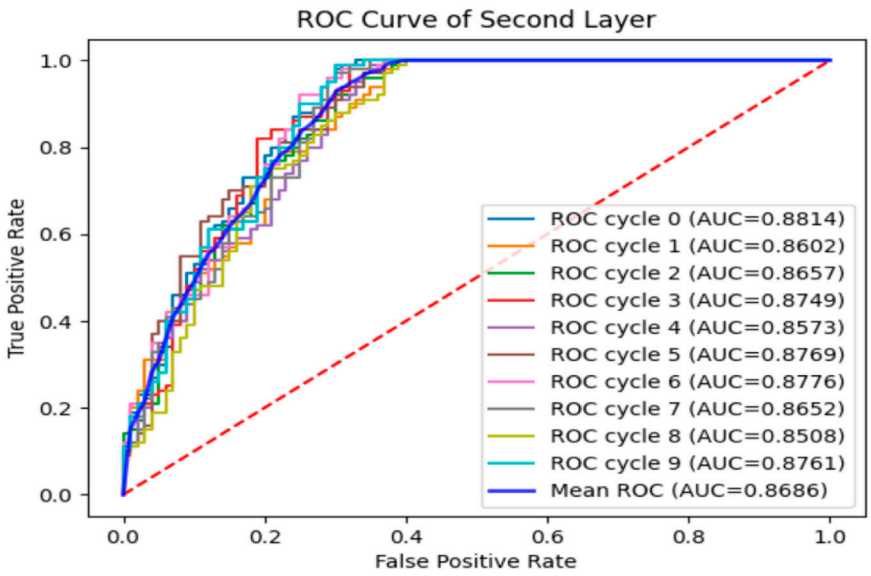


FIGURE 6
ROC curves for layer 2 (enhancer strength prediction).

TABLE 6 Comparison with other methods on the same independent datasets.

Layer	Predictor	SN	SP	Acc	MCC	AUC	Source
First layer	iEnhancer-2L	71.00	75.00	73.00	0.4604	0.8062	Liu et al. (2015)
	EnhancerPred	73.50	74.50	74.00	0.4800	0.8013	Jia and He (2016)
	iEnhancer-EL	71.00	78.50	74.75	0.4964	0.8173	Liu et al. (2018a)
	iEnhancer-ECNN	78.50	75.20	76.90	0.5370	0.8320	Nguyen et (2019)
	iEnhancer-XG	75.75	74.00	77.50	0.5150	—	Cai et al. (2021)
	iEnhancer-EBLSTM	75.50	79.50	77.20	0.5340	0.7720	Niu et al. (2021)
	Enhancer-FRL	80.50	75.50	78.00	0.5607	0.8573	Wang et al. (2022a)
	iEnhancer-DCSV	80.25	77.65	**78.95**	**0.5809**	0.8527	This study
Second layer	iEnhancer-2L	47.00	74.00	60.50	0.2181	0.6678	Liu et al. (2015)
	EnhancerPred	45.00	65.00	55.00	0.1021	0.5790	Jia and He (2016)
	iEnhancer-EL	54.00	68.00	61.00	0.2222	0.6801	Liu et al. (2018b)
	iEnhancer-ECNN	79.10	56.40	67.80	0.3680	0.7480	Nguyen et (2019)
	iEnhancer-XG	70.00	57.00	63.50	0.2720	—	Cai et al. (2021)
	iEnhancer-EBLSTM	81.20	53.60	65.80	0.3240	0.6580	Niu et al. (2021)
	Enhancer-FRL	98.00	49.00	73.50	0.5391	0.8723	Wang et al. (2022b)
	iEnhancer-DCSV	**99.10**	62.30	**80.70**	**0.6609**	0.8686	This study

3.4 Comparison of different model frameworks

In this study, we used six network frameworks: ResNet, DenseNet, DenseNet + channel attention model, DenseNet + spatial attention model, DenseNet + CBAM attention model, and DenseNet + improved CBAM attention model. We tested these five network frameworks in the first layer (enhancer recognition) task because the amount of data for the second layer (enhancer strength prediction) task was too small. The original features were extracted using each of these five network frameworks for the high-level features, and the best-performing network framework was selected based on the experimental results. The experimental comparison results are shown in Table 4. Adding an attention

model behind the DenseNet is already very effective, and the improved CBAM attention model integrates the advantages of both attention models. However, the improved effect is limited because the shape of the feature map is too small. The results show that the DenseNet + improved CBAM attention network framework works better. Therefore, we finally chose the DenseNet + improved CBAM attentional network framework model.

3.5 Performance of iEnhancer-DCSV on the training dataset

To verify the performance of the iEnhancer-DCSV classifier, we cycled through 10 times of fivefold cross-validation, and the experimental results are shown in Table 5. We found that the values of the evaluation metrics fluctuated relatively steadily on the first (enhancer recognition) and second (enhancer strength prediction) layer tasks, indicating that the iEnhancer-DCSV model has good generalization capability. Figure 5 shows the ROC curves of the first layer (enhancer recognition) with a mean AUC value of 0.8527 in 10 experiments, and Figure 6 shows the ROC curves of the second layer (enhancer strength prediction) with a mean AUC value of 0.8686 in 10 experiments. The results show that our proposed iEnhancer-DCSV has good performance.

3.6 Comparison of iEnhancer-DCSV with existing predictors

The iEnhancer-DCSV predictor proposed in this study is compared with seven existing predictors. The performance of independent datasets under different methods is shown in Table 6. The iEnhancer-DCSV predictor has better Acc and MCC metrics compared with others. The improvement ranges for ACC and MCC in the first layer (enhancer recognition) were 1.95%–5.95% and 0.0202–0.1205, respectively, and the improvement ranges for ACC and MCC in the second layer (enhancer strength prediction) were 7.2%–25.7% and 0.1218–0.5588, respectively. Meanwhile, in the first and second layers, the SN and SP metrics also have some advantages, indicating that iEnhancer-DCSV is more balanced and has more stable and superior performance in identifying positive and negative samples. The iEnhancer-DCSV predictor is expected to be the most advanced and representative tool for predicting enhancement and its strengths and weaknesses.

4 Conclusion

In this study, we propose a new predictor of enhancer recognition and its strength called iEnhancer-DCSV. It is based on DenseNet and an improved CBAM attention module approach. The experimental results demonstrate that the MCC value for enhancer identification on the independent test set is 0.5809, and the MCC value for enhancer strength prediction is 0.6609. This indicates that the iEnhancer-DCSV predictor has good performance and generalization ability, which is better than the existing prediction tools. We combine deep learning methods with enhancer research to innovate computational methods in the field of bioinformatics and enrich enhancer research. In the future, the iEnhancer-DCSV predictor not only is applicable to enhancer classification tasks but can also be used in different prediction tasks, making its use convenient for researchers.

Of course, some deficiencies must be overcome in our proposed model. The current enhancer sample of data is small and fails to sufficiently promote the performance of the iEnhancer-DCSV model using a big data-driven approach. In addition, data enhancement strategies were not employed to augment our data samples, such as generative adversarial networks (GANs) (Li and Zhang, 2021). This will be our future work issue to address. However, as the research on enhancers progresses, the disadvantage of a small amount of data will gradually disappear, and better deep learning methods will be used in the research, creating more possibilities for future enhancer recognition and strength prediction.

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 authors.

Author contributions

JJ and RL conceived and designed the experiments. RL implemented feature extraction, model construction, model training, and performance evaluation. RL, LQ, and XW drafted the manuscript and revised the manuscript. JJ supervised this study. All authors contributed to the content of this paper and approved the final manuscript.

Funding

This work was partially supported by the National Natural Science Foundation of China (nos 61761023, 62162032, and 31760315), the Natural Science Foundation of Jiangxi Province, China (nos 20202BABL202004 and 20202BAB202007), and the Scientific Research Plan of the Department of Education of Jiangxi Province, China (GJJ190695 and GJJ212419). These funders had no role in the study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Acknowledgments

The authors are grateful for the constructive comments and suggestions made by the reviewers.

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

Bauer, E., and Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Mach. Learn.* 36, 105–139. doi:10.1023/a:1007515423169

Boyd, M., Thodberg, M., Vitezic, M., Bornholdt, J., Vitting-Seerup, K., Chen, Y., et al. (2018). Characterization of the enhancer and promoter landscape of inflammatory bowel disease from human colon biopsies. *Nat. Commun.* 9, 1661. doi:10.1038/s41467-018-03766-z

Bu, H., Gan, Y., Wang, Y., Zhou, S., and Guan, J. (2017). A new method for enhancer prediction based on deep belief network. *BMC Bioinforma.* 18, 418. doi:10.1186/s12859-017-1828-0

Buccitelli, C., and Selbach, M. (2020). mRNAs, proteins and the emerging principles of gene expression control. *Nat. Rev. Genet.* 21, 630–644. doi:10.1038/s41576-020-0258-4

Cai, L., Ren, X., Fu, X., Peng, L., Gao, M., and Zeng, X. (2021). iEnhancer-XG: interpretable sequence-based enhancers and their strength predictor. *Bioinformatics* 37, 1060–1067. doi:10.1093/bioinformatics/btaa914

Chen, L., Zhang, H., Xiao, J., Nie, L., Shao, J., Liu, W., et al. (Year). "SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning", in: 2017 Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)), 6298–6306.

Chou, K. C. (1984). Low-frequency vibrations of DNA molecules. *Biochem. J.* 221, 27–31. doi:10.1042/bj2210027

Cvetesic, N., and Lenhard, B. (2017). Core promoters across the genome. *Nat. Biotechnol.* 35, 123–124. doi:10.1038/nbt.3788

Dai, Q., Bao, C., Hai, Y., Ma, S., Zhou, T., Wang, C., et al. (2018). MTGIpick allows robust identification of genomic islands from a single genome. *Brief. Bioinform* 19, 361–373. doi:10.1093/bib/bbw118

Emison, E. S., Mccallion, A. S., Kashuk, C. S., Bush, R. T., Grice, E., Lin, S., et al. (2005). A common sex-dependent mutation in a RET enhancer underlies Hirschsprung disease risk. *Nature* 434, 857–863. doi:10.1038/nature03467

Erwin, G. D., Oksenberg, N., Truty, R. M., Kostka, D., Murphy, K. K., Ahituv, N., et al. (2014). Integrating diverse datasets improves developmental enhancer prediction. *PLoS Comput. Biol.* 10, e1003677. doi:10.1371/journal.pcbi.1003677

Fernández, M., and Miranda-Saavedra, D. (2012). Genome-wide enhancer prediction from epigenetic signatures using genetic algorithm-optimized support vector machines. *Nucleic Acids Res.* 40, e77. doi:10.1093/nar/gks149

Firpi, H. A., Ucar, D., and Tan, K. (2010). Discover regulatory DNA elements using chromatin signatures and artificial neural network. *Bioinformatics* 26, 1579–1586. doi:10.1093/bioinformatics/btq248

Ghandi, M., Lee, D., Mohammad-Noori, M., and Beer, M. A. (2014). Enhanced regulatory sequence prediction using gapped k-mer features. *PLoS Comput. Biol.* 10, e1003711. doi:10.1371/journal.pcbi.1003711

He, W., and Jia, C. (2017). EnhancerPred2.0: Predicting enhancers and their strength based on position-specific trinucleotide propensity and electron–ion interaction potential feature selection. *Mol. Biosyst.* 13, 767–774. doi:10.1039/c7mb00054e

Huang, Y., Niu, B., Gao, Y., Fu, L., and Li, W. (2010). CD-HIT suite: A web server for clustering and comparing biological sequences. *Bioinformatics* 26, 680–682. doi:10.1093/bioinformatics/btq003

Jia, C., and He, W. (2016). EnhancerPred: A predictor for discovering enhancers based on the combination and selection of multiple features. *Sci. Rep.* 6, 38741. doi:10.1038/srep38741

Kim, T.-K., Hemberg, M., Gray, J. M., Costa, A. M., Bear, D. M., Wu, J., et al. (2010). Widespread transcription at neuronal activity-regulated enhancers. *Nature* 465, 182–187. doi:10.1038/nature09033

Kleftogiannis, D., Kalnis, P., and Bajic, V. B. (2015). Deep: A general computational framework for predicting enhancers. *Nucleic Acids Res.* 43, e6. doi:10.1093/nar/gku1058

Kong, R., Xu, X., Liu, X., He, P., Zhang, M. Q., and Dai, Q. (2020). 2SigFinder: The combined use of small-scale and large-scale statistical testing for genomic island detection from a single genome. *BMC Bioinforma.* 21, 159. doi:10.1186/s12859-020-3501-2

Li, M., and Zhang, W. (2021). Phiaf: Prediction of phage-host interactions with GAN-based data augmentation and sequence-based feature fusion. *Briefings Bioinforma.* 23, bbab348. doi:10.1093/bib/bbab348

Li, X., Han, P., Chen, W., Gao, C., Wang, S., Song, T., et al. (2022). Marppi: Boosting prediction of protein–protein interactions with multi-scale architecture residual network. *Briefings Bioinforma.* 24, bbac524. doi:10.1093/bib/bbac524

Liu, B., Fang, L., Long, R., Lan, X., and Chou, K.-C. (2015). iEnhancer-2L: a two-layer predictor for identifying enhancers and their strength by pseudo k-tuple nucleotide composition. *Bioinformatics* 32, 362–369. doi:10.1093/bioinformatics/btv604

Liu, B., Li, K., Huang, D.-S., and Chou, K.-C. (2018a). iEnhancer-EL: identifying enhancers and their strength with ensemble learning approach. *Bioinformatics* 34, 3835–3842. doi:10.1093/bioinformatics/bty458

Liu, G., Zhang, Y., Wang, L., Xu, J., Chen, X., Bao, Y., et al. (2018b). Alzheimer's disease rs11767557 variant regulates EPHA1 gene expression specifically in human whole blood. *J. Alzheimer's Dis.* 61, 1077–1088. doi:10.3233/JAD-170468

Lu, Y., Qu, W., Shan, G., and Zhang, C. (2015). Delta: A distal enhancer locating tool based on AdaBoost algorithm and shape features of chromatin modifications. *PLoS One* 10, e0130622. doi:10.1371/journal.pone.0130622

Maston, G. A., Evans, S. K., and Green, M. R. (2006). Transcriptional regulatory elements in the human genome. *Annu. Rev. Genomics Hum. Genet.* 7, 29–59. doi:10.1146/annurev.genom.7.080505.115623

Min, S., Lee, B., and Yoon, S. (2016). Deep learning in bioinformatics. *Briefings Bioinforma.* 18, bbw068–869. doi:10.1093/bib/bbw068

Nguyen, Q. H., Nguyen-Vo, T. H., Le, N. Q. K., Do, T. T. T., Rahardja, S., and Nguyen, B. P. (2019). iEnhancer-ECNN: identifying enhancers and their strength using ensembles of convolutional neural networks. *BMC Genomics* 20, 951. doi:10.1186/s12864-019-6336-3

Niu, K., Luo, X., Zhang, S., Teng, Z., Zhang, T., and Zhao, Y. (2021). iEnhancer-EBLSTM: Identifying enhancers and strengths by ensembles of bidirectional long short-term memory. *Front. Genet.* 12, 665498. doi:10.3389/fgene.2021.665498

Pennacchio, L. A., Bickmore, W., Dean, A., Nobrega, M. A., and Bejerano, G. (2013). Enhancers: Five essential questions. *Nat. Rev. Genet.* 14, 288–295. doi:10.1038/nrg3458

Shang, Y., Ye, X., Futamura, Y., Yu, L., and Sakurai, T. (2022). Multiview network embedding for drug-target Interactions prediction by consistent and complementary information preserving. *Briefings Bioinforma.* 23, bbac059. doi:10.1093/bib/bbac059

Shlyueva, D., Stampfel, G., and Stark, A. (2014). Transcriptional enhancers: From properties to genome-wide predictions. *Nat. Rev. Genet.* 15, 272–286. doi:10.1038/nrg3682

Shrinivas, K., Sabari, B. R., Coffey, E. L., Klein, I. A., Boija, A., Zamudio, A. V., et al. (2019). Enhancer features that drive formation of transcriptional condensates. *Mol. Cell* 75, 549–561.e7. doi:10.1016/j.molcel.2019.07.009

Sinden, R. R., Pearson, C. E., Potaman, V. N., and Ussery, D. W. (1998). "Dna: Structure and function," in *Advances in genome biology*. Editor R. S. VermaJAI, 1–141.

Singh, M., Rajagopal, N., Xie, W., Li, Y., Wagner, U., Wang, W., et al. (2013). Rfecs: A random-forest based algorithm for enhancer identification from chromatin state. *PLoS Comput. Biol.* 9, e1002968. doi:10.1371/journal.pcbi.1002968

Sokolova, M., and Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Inf. Process. Manag.* 45, 427–437. doi:10.1016/j.ipm.2009.03.002

Vacic, V., Iakoucheva, L. M., and Radivojac, P. (2006). Two sample logo: A graphical representation of the differences between two sets of sequence alignments. *Bioinformatics* 22, 1536–1537. doi:10.1093/bioinformatics/btl151

Wang, C., Zou, Q., Ju, Y., and Shi, H. (2022a). Enhancer-FRL: Improved and robust identification of enhancers and their activities using feature representation learning. *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, 1–9. doi:10.1109/TCBB.2022.3204365

Wang, Y., Peng, Q., Mou, X., Wang, X., Li, H., Han, T., et al. (2022b). A successful hybrid deep learning model aiming at promoter identification. *BMC Bioinforma.* 23, 206. doi:10.1186/s12859-022-04735-6

Wang, Y., Xu, Y., Yang, Z., Liu, X., and Dai, Q. (2021). Using recursive feature selection with random forest to improve protein structural class prediction for low-

similarity sequences. *Comput. Math. Methods Med.* 2021, 5529389. doi:10.1155/2021/5529389

Wu, S., Ou, T., Xing, N., Lu, J., Wan, S., Wang, C., et al. (2019). Whole-genome sequencing identifies ADGRG6 enhancer mutations and FRS2 duplications as angiogenesis-related drivers in bladder cancer. *Nat. Commun.* 10, 720. doi:10.1038/s41467-019-08576-5

Xiao, X., Xu, Z. C., Qiu, W. R., Wang, P., Ge, H. T., and Chou, K. C. (2019). iPSW(2L)-PseKNC: A two-layer predictor for identifying promoters and their strength by hybrid features via pseudo K-tuple nucleotide composition. *Genomics* 111, 1785–1793. doi:10.1016/j.ygeno.2018.12.001

Xiao, Z., Zou, Q., Liu, Y., and Yang, X. (2016). Genome-wide assessment of differential translations with ribosome profiling data. *Nat. Commun.* 7, 11194. doi:10.1038/ncomms11194

Yang, B., Liu, F., Ren, C., Ouyang, Z., Xie, Z., Bo, X., et al. (2017). BiRen: Predicting enhancers with a deep-learning-based model using the DNA sequence alone. *Bioinformatics* 33, 1930–1936. doi:10.1093/bioinformatics/btx105

Yang, S., Wang, Y., Chen, Y., and Dai, Q. (2020). Masqc: Next generation sequencing assists third generation sequencing for quality control in N6-methyladenine DNA identification. *Front. Genet.* 11, 269. doi:10.3389/fgene.2020.00269

Yang, Z., Yi, W., Tao, J., Liu, X., Zhang, M. Q., Chen, G., et al. (2022). HPVMD-C: A disease-based mutation database of human papillomavirus in China. *Database J. Biol. Databases Curation* 2022. doi:10.1093/database/baac018

Zhang, T., Tang, Q., Nie, F., Zhao, Q., and Chen, W. (2022). DeepLncPro: An interpretable convolutional neural network model for identifying long non-coding RNA promoters. *Briefings Bioinforma.* 23, bbac447. doi:10.1093/bib/bbac447

frontiers | Frontiers in Genetics

TYPE Technology and Code
PUBLISHED 27 March 2023
DOI 10.3389/fgene.2023.1138582

OPEN ACCESS

EDITED BY
Nathan Olson,
National Institute of Standards and Technology (NIST), United States

REVIEWED BY
Vladimir Potapov,
New England Biolabs, United States
Mark Akeson,
University of California, Santa Cruz, United States

*CORRESPONDENCE
Matt Loose,
matt.loose@nottingham.ac.uk

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

RECEIVED 06 January 2023
ACCEPTED 06 March 2023
PUBLISHED 27 March 2023

CITATION
Munro R, Holmes N, Moore C, Carlile M, Payne A, Tyson JR, Williams T, Alder C, Snell LB, Nebbia G, Santos R and Loose M (2023), A framework for real-time monitoring, analysis and adaptive sampling of viral amplicon nanopore sequencing.
Front. Genet. 14:1138582.
doi: 10.3389/fgene.2023.1138582

A framework for real-time monitoring, analysis and adaptive sampling of viral amplicon nanopore sequencing

Rory Munro[1], Nadine Holmes[2], Christopher Moore[2], Matthew Carlile[2], Alexander Payne[1], John R. Tyson[3], Thomas Williams[4], Christopher Alder[4], Luke B. Snell[4], Gaia Nebbia[4], Roberto Santos[5] and Matt Loose[2]*

[1]School of Life Sciences, University of Nottingham, Nottingham, United Kingdom, [2]DeepSeq, University of Nottingham, Nottingham, United Kingdom, [3]BCCDC Public Health Laboratory, Vancouver, BC, Canada, [4]Department of Infection, Guy's and St Thomas' NHS Foundation Trust, London, United Kingdom, [5]Microsoft Research, São Paulo, Brazil

The ongoing SARS-CoV-2 pandemic demonstrates the utility of real-time sequence analysis in monitoring and surveillance of pathogens. However, cost-effective sequencing requires that samples be PCR amplified and multiplexed *via* barcoding onto a single flow cell, resulting in challenges with maximising and balancing coverage for each sample. To address this, we developed a real-time analysis pipeline to maximise flow cell performance and optimise sequencing time and costs for any amplicon based sequencing. We extended our nanopore analysis platform MinoTour to incorporate ARTIC network bioinformatics analysis pipelines. MinoTour predicts which samples will reach sufficient coverage for downstream analysis and runs the ARTIC networks Medaka pipeline once sufficient coverage has been reached. We show that stopping a viral sequencing run earlier, at the point that sufficient data has become available, has no negative effect on subsequent down-stream analysis. A separate tool, SwordFish, is used to automate adaptive sampling on Nanopore sequencers during the sequencing run. This enables normalisation of coverage both within (amplicons) and between samples (barcodes) on barcoded sequencing runs. We show that this process enriches under-represented samples and amplicons in a library as well as reducing the time taken to obtain complete genomes without affecting the consensus sequence.

KEYWORDS

bioinformactics, software, pipeline, viral sequence analysis, genomics, nanopore sequencing, oxford nanopore minION, oxford nanopore technologies (ONT)

1 Introduction

Oxford Nanopore Technologies (ONT) sequencers (MinION, GridION, Promethion) have allowed sequencing to become a dynamic, real-time process (Jain et al., 2016). By writing batches of sequenced reads to disk after DNA has finished translocating a pore, these data become available immediately, enabling parallel data analysis and so reducing the time required to provide insight into the sequenced sample. Even prior to the ongoing SARS-CoV-2 pandemic, the benefits of real-time analysis of

sequence data have been demonstrated (Quick and Loman, 2016; Gardy and Loman, 2018), and rapid lineage assignment and Variant of Concern/Variant under Investigation (VoC/VuI) status can be time sensitive when tracking a new variant (O'Toole et al., 2021).

The ARTIC Network (Quick et al., 2017; Tyson, 2020) (https://artic.network) provides comprehensive protocols for both wet lab and downstream best practice informatics analyses for SARS-CoV-2, amidst other pathogenic viruses. The use of PCR amplification can lead to unequal coverage of individual amplicons in a sequencing library such that some reach sufficient coverage for reliable analysis faster than others. Further sequencing of these amplicons with sufficient coverage will not benefit the final down-stream analysis. Even using 96 barcodes to multiplex samples, the average ONT MinION/PromethION flowcell is capable of providing more data than required. Ideally, sequencing would be stopped as soon as sufficient data are available for analysis with balanced coverage of amplicons in the library. Aside from wet lab optimisations, ONT sequencers offer Run Until, the ability to stop sequencing once some pre-defined condition has been met, and adaptive sampling (Payne 2020), the ability to stop sequencing and unblock off target DNA from the pore, which may help address these problems.

As part of the COG-UK network (Cog, 2020; Nicholls et al., 2021) we generated thousands of SARS-CoV-2 consensus sequences using ONT sequencers. To test the utility of run until in this context, we incorporated the ARTIC pipeline into our minoTour tool (Munro et al., 2021) (https://github.com/looselab/minotourapp) and developed a model to predict if sufficient coverage will be obtained for each barcoded sample on a flowcell, stopping sequencing when all samples predicted to achieve sufficient coverage do so. We demonstrate this has no effect on the ability to assign lineages (O'Toole et al., 2021) to samples and minimal impact on SNP calls. The resultant shorter sequencing runs preserve flow cell health, allowing them to be flushed and reused for other experiments, reducing the effective cost per sample for sequencing.

To determine if adaptive sampling could be used to select individual amplicons from one or more samples to improve and balance the coverage across SARS-CoV-2 genomes we developed SwordFish. This tool enables truly "dynamic" adaptive sampling by providing feedback between minoTour and the ReadFish pipeline (https://github.com/looselab/swordfish). SwordFish couples ReadFish to minoTour by querying minoTour for information on specific sequencing runs and updating ReadFish (Payne et al., 2021) with new barcode/amplicon targets in response to ongoing data generation (see Figure 1). Using a custom 1,200 base pair amplicon scheme (Supplementary File S3.5) we show adaptive sampling can filter out over abundant samples and individual amplicons, and coupled with run until, results in time savings and an increase in the number of amplicons reaching median 20× coverage.

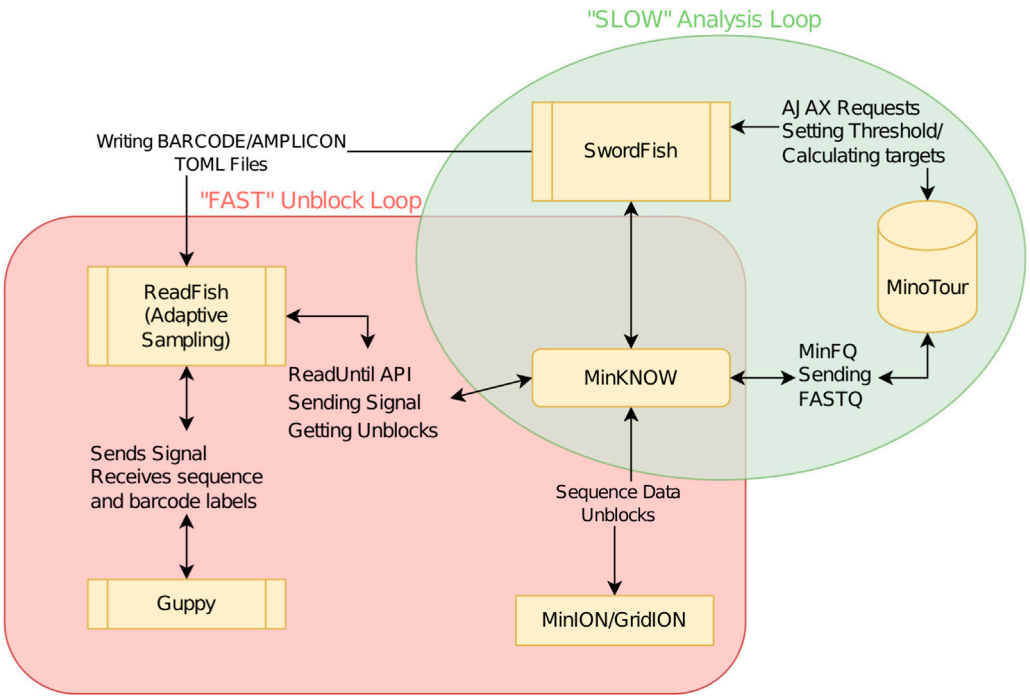


FIGURE 1
Flowchart demonstrating ReadFish/SwordFish/minoTour interactions. The slow analysis loop (green) is used to update ReadFish's target TOML file. The loop is run once every 60 s. MinFQ uploads FASTQ sequence data to minoTour, which tracks coverage for each amplicon on each barcode. SwordFish queries minoTour for the set of amplicon coordinates on each barcode to unblock. These are defined as those exceeding a specified level of coverage (e.g., 50×). SwordFish updates a TOML file that can be read by ReadFish. The fast analysis loop (red) is run every read batch (approximately 0.8 s). ReadFish updates its coordinates from the TOML file, base calls and demultiplexes all reads in the batch using Guppy, and then sends unblock signals to MinKNOW for any reads that align inside any amplicon with sufficient coverage.

2 Implementation

2.1 MinoTour ARTIC pipeline implementation

The standard ARTIC pipeline uses Nanopolish (Loman et al., 2015; Quick et al., 2017) for signal level analysis of raw ONT data during variant calling. Unfortunately, signal level data are unavailable within minoTour. Instead we integrated ARTIC's alternate Medaka (https://github.com/nanoporetech/medaka) workflow to enable real-time generation of consensus genomes during sequencing. MinoTour's workflow contrasts with other web based analysis platforms which either do not exploit the real-time features of the nanopore platform or do not have access to the sequence data themselves for further analysis (Bruno et al., 2021; Ferguson et al., 2021). The ARTIC network does provide a tool, RAMPART, which can monitor a run over time and complete analysis for individual samples, but does not provide many of the other features shown here at this time, such as lineage analysis or adaptive sampling. (https://artic.network/rampart). We integrated the ARTIC SARS-CoV-2 Medaka pipeline into minoTour as a custom python script, which is run as a Celery task (https://docs.celeryproject.org/en/stable/), processing read batches as they are uploaded. The pipeline is asynchronous, preventing blocking of any other analyses being performed. Uploaded reads are filtered by length, with the minimum and maximum read lengths permissible calculated from the underlying amplicon scheme. Reads are further filtered by the QC score assigned by Guppy (assigned pass by Guppy) and then mapped to an appropriate SARS-CoV-2 reference using minimap2 (Li, 2018). Per base coverage is tracked in optimised numpy arrays using the mapped reads in real-time (Walt and Varoquaux, 2011).

Coverage is tracked for each individual amplicon on a sample as defined by the primer scheme in use. Default parameters for triggering the analysis of a specific sample are at least 90% of the amplicons (completeness) covered at a median depth of at least 20×, though these are user configurable. Once triggered, the accumulated mapped reads for that sample are passed to the ARTIC network's Medaka pipeline. Numerous primer schemes can be chosen, including custom schemes, simply by creating the appropriate primer scheme and reference files and uploading them to minoTour.

MinoTour uses pangolin (O'Toole et al., 2021) to assign a PANGO lineage from the most recent lineage classifications. Consensus sequences are also compared with current VoC/VuI definitions as defined at https://github.com/phe-genomics/variant_definitions, using the Aln2Type tool (https://github.com/connor-lab/aln2type). Both PANGO lineages and VoC/VuI designations are automatically updated daily by minoTour. A report is generated for each sample (see Supplementary Figures S1A, B) and optionally users can be notified of VoC/VuI identifications *via* the minoTour Twitter API. Sequences within each run are also globally aligned using MAFFT (Katoh et al., 2002), with an illustrative tree generated using iQ-Tree (Minh et al., 2020) and visualised with figtree.js or ToyTree (Rambaut, 2021). Additional background sequences can be included in these trees if desired and the distribution of SNPs within consensus sequences from the run, compared with the reference are displayed in a SNIPIT plot (https://github.com/aineniamh/snipit) (Supplementary Figure S1A).

Results from the pipeline are maintained for historical record, with files stored on disk and metadata and metrics about the ARTIC sequencing experiment stored in a SQL database. These results are then visualised in the minoTour web server. Once a run has completed, which is automatically recognised by the fact that no further data are added to the flow cell within a fixed period of time, all analyses are automatically re-run to ensure maximum coverage for consensus generation. A retention policy for sequence data is set globally for the site and all read data can be automatically scrubbed from the server after consensus generation, if desired.

2.2 ARTIC visualisations and reports

If running an ARTIC analysis on a flow cell, minoTour provides a custom page containing all ARTIC data and visualisations (Supplementary Figures S1A–D). This page shows the performance of all samples in the run and then visualises detailed performance and information available for an individual sample. A sortable and searchable summary table shows users metrics about each sample in the run, with average coverage, number of amplicons at different depths and basic statistics such as mean read length and read count. If the sample had sufficient data to be run through the ARTIC pipeline, we display the assigned lineage and VoC/VoI status.

Further details can be seen for a chosen sample such as per base coverage plots for the sample genome. Assigned PANGO lineage information is provided in tabular form, with links out to further information describing each lineage (https://cov.lineages.org and https://outbreak.info). The VoC/VuI report generated by Aln2Type is visualised and the final status assigned displayed. A PDF report for each barcoded sample and the overall run can be exported, showing all above metrics for each sample. An example can be found in Supplementary File S3.1.

Pass and fail VCF files, BAM files and pangolin lineages can be downloaded. Optionally, these features can be disabled and minoTour will remove all files that may contain identifiable sequence information from the server. By maintaining compatibility with standard ARTIC bioinformatics pipelines, this tool can be adapted to run any ARTIC compatible pathogen analysis simply by uploading the appropriate reference files.

2.3 Amplicon coverage prediction model

To predict if individual samples are likely to result in an informative genome sequence, providing the basis for minoTour's decision on when to stop the run, minoTour assumes the user is seeking minimal useful genome completeness (default 90% amplicons with at least 20× median "pass" read coverage). Using median coverage depth reduces the impact of small insertions/deletions on monitoring amplicon coverage. In addition, median coverage is only calculated for unique regions of each amplicon, removing any overlap between amplicons. This prevents amplicons with more than 50% overlap being incorrectly labelled as complete due to the coverage of a neighbouring amplicon. MinoTour then assumes that each ONT flow cell can generate a minimum of 100,000 reads for each sample detected and so projects whether

each sample will reach minimal useful completeness using a simple model (Equation 1). A sample is projected to finish if 90% of the amplicons have a predicted final coverage over the minimum required coverage (default 20×). All sequencing runs gather data for 1 h before any of our strategies are used to ensure reasonable sampling of the loaded library.

$$\left(\frac{\text{Amplicon median coverage}}{\text{Total mapped reads}} \times \text{Barcodes identified} \times 100,000\right) \geq \text{Min. required coverage} \quad (1)$$

2.4 SwordFish—real-time readfish target updating software

Swordfish provides a python based command line interface to connect minoTour to ReadFish *via* minoTour's Representational State Transfer (REST) Application Programming Interface (API), querying for updates at a user specified interval. In the context of amplicon based sequencing, SwordFish receives a list of barcodes and amplicon genomic coordinates for each barcode from minoTour, where the median coverage for any returned amplicon exceeds the user defined threshold. SwordFish then adds the coordinates of these over coverage amplicons to the rejection targets for the correct barcode in ReadFish's configuration file. ReadFish will then reject any future reads corresponding to that amplicon. If a barcoded sample has completed analysis, SwordFish can switch off that barcode entirely for the remainder of the experiment. The relationship between minoTour, swordfish and ReadFish is shown more clearly in Figure 1. It is worth noting that whilst this manuscript focuses on SARS-CoV-2, this approach is applicable to any viral amplicon primer scheme that can be used with the ARTIC field bioinformatics pipeline provided the amplicons are sufficiently long and ligation, not rapid, sequencing is used. If rapid kit based sequencing were to be used, the amplicons would have to be of sufficient size to generate a library with a long enough mean read length that the software would have time to unblock them.

2.5 Post run genome analysis

To determine how manipulating run time affects results, we defined three time points of interest for a sample during a sequencing run. The Full Run time point, the Run Complete time point and the Sample Complete time point. Full Run is defined as the time at which the run completed with no intervention. Run Complete is the point in a run where all samples our algorithm predicted would complete (90% completeness, 20×) had done so. Finally, Sample Complete is the point at which an *individual* sample in a run reached sufficient completeness and is automatically put through the ARTIC pipeline by minoTour, whilst the run continues. A sequencing run will have only one Full run time point, one Run complete time point, but will have many Sample Complete time points. This concept is visualised in Figure 2.

To create consensus genomes from time points equivalent to our ARTIC pipeline and compare the results of both Medaka and Nanopolish we had to calculate the sets of both the signal (FAST5) and FASTQ files equivalent to those that would have been uploaded to minoTour at each of the time points. We iteratively mapped all reads from each barcode across 13 reference ARTIC runs using minimap2 (Li, 2018), in FASTQ file creation order, creating cumulative alignment files. Using mosdepth (Pedersen and Aaron, 2018) we determined cumulative coverage at each base across the reference genome, for each FASTQ file creation time point, and then the median coverage for each amplicon using the same primer scheme based approach as in minoTour. This identifies the time points in each run when sufficient data are available to trigger minoTour to analyse the samples, as well as the points that minoTour would have recommended stopping the run based on it is amplicon coverage predictions. The creation time point for the FASTQ file that results in sufficient coverage to meet any appropriate thresholds was used to identify the time in the sequencing run when analysis would occur. Using this method, we can identify the equivalent FAST5 file for that FASTQ file from the ONT sequencing summary file, enabling us to analyse the data with both Medaka and Nanopolish (code available from https://github.com/LooseLab/artic_minotour_analyses). For each time-point, we generated consensus FASTA files to calculate genome recovery, defined as the proportion of non N positions in the final sequence. This is a close approximation of the minoTour completeness metric, as any base that has 20× coverage going into the ARTIC Medaka pipeline will most likely be called as non N.

3 Results and discussion

3.1 Amplicon coverage prediction model

The amplicon prediction model performed well across all runs (Figure 3A, R^2 = 0.991). The model proved to be conservative, slightly under-predicting against final coverage, which prevented minoTour from waiting for genomes to complete which would never do so. After an hour of data, predicted genome recovery collection compares well with that observed at the calculated Run Complete times for the 13 runs (Figure 3B). The strong correlation (R^2 = 0.993) between predicted values and values actually recovered provides confidence in our algorithm. Comparing the genome recovery achieved at the Run Complete time point with the genome recovery seen at Full Run (Figure 3C) shows some small further benefits in recovery (R^2 = 0.996) when allowing the run to reach natural completion. This is expected as continuing the run for longer allows the missing 10% of each genome to acquire some further coverage. However the longer a run continues the more it is information return diminishes, so stopping earlier accelerates time to answer as well as allowing the flow cell to be reused and save costs. This can be seen more clearly in Figure 3D, (R^2 = 0.994) when filtering out those runs where no time is saved by our model, as these runs have the same time defined for Run Complete and Full Run.

Our model confidently predicts if a sample will generate sufficient data to provide useful information with enough accuracy to support a decision on whether or not to continue

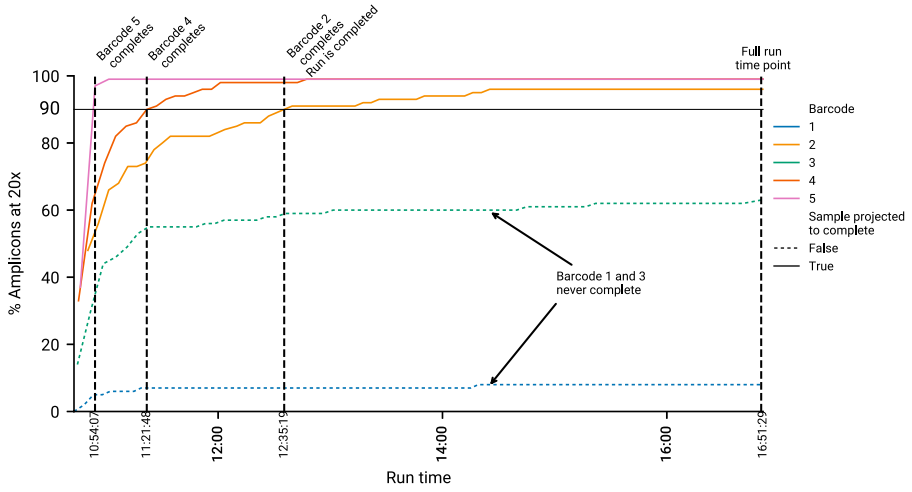


FIGURE 2
An example run, showing how timepoints are calculated. Five barcodes are displayed. Black horizontal line indicates the 20× coverage on 90% of amplicons threshold. The time at which a barcode reaches this threshold is recorded as the "Sample Completion" (SC) time point. Two illustrative samples, barcode 1 and 3, are not predicted to finish, and do not cross this threshold. Hence they have no associated SC time. Once all barcodes predicted to finish are complete, we record the "Run Completed" (RC) time point. This is the time minoTour would recommend stopping the run. The Full Run time is when the run stopped without any early intervention.

sequencing. There is a potential small loss in data as a consequence of reducing the sequencing time. We therefore quantify the consequences of this on time saved, lineage assignment and SNP calling below.

3.2 Run until time savings

To quantify whether this approach results in useful time savings, we tracked metrics and predicted amplicon coverages per barcoded genome sample using minoTour for 13 sequencing runs. We visualised a comparison of calculated Run Complete time and Full Run time in Figure 4A. Runs 9 through 13 were actively monitored with minoTour and manually stopped at earlier run times in response to the model predictions, resulting in the shorter Full Run length and the similarly quick Run Complete time point. Time savings using this approach are dependent on the sample composition, but are often significant (for example, Run 4, Figure 4A). By plotting all barcodes on every run, we can visualise the point in time in a run when all barcodes predicted to finish in a run cross the threshold. Negative controls are treated as a sample, and are predicted to fail, so they do not prevent a run from completing. Time savings are greater in runs with fewer samples as each has relatively more sequencing capacity available as can be seen in Figures 4B–N.

3.3 SNPS and lineages

Finally we investigated whether stopping early affected the information you can retrieve from consensus genomes, and compared whether minoTour loses SNP accuracy by using the Medaka pipeline rather than Nanopolish.

3.3.1 Lineage assignment to consensus genomes

Across all 13 sequencing runs, a total of 508 SARS-CoV-2 samples were sequenced (including negative and positive controls). The number of genomes produced by the ARTIC Pipeline at each time point were: Full run, 456 genomes; Run Completed, 454 genomes; Sample Completed, 334 genomes. The two additional genomes produced at the Full Run time over the Run Complete time are both extremely low completeness genomes (only 1% of the genome has consensus sequence) that failed to call at the Run Complete time. Across all time points for any given sample in any run, we observe complete concordance in lineage assignment between either Medaka or Nanopolish generated genomes (Supplementary File S3.2). Any loss of data seen by stopping sequencing early did not impact PANGO lineage assignment in a SARS-CoV-2 sequencing run. We note that these sequences are predominantly from the B.1.1.7 lineage due to the time periods in which they were collected, but given our observations on SNP calling below do not envisage this being an issue.

3.3.2 Comparing SNPs between Medaka and Nanopolish consensus genomes

We compared Nanopolish and Medaka consensus genome sequences for all genomes in our data set (1,245 genomes across Full Run, Run Complete and Sample Complete time points from 508 unique samples). The SNPs were called using nextclade (https://clades.nextstrain.org) with the output data available in Supplementary Files S3.3, S3.4.

Of the 456 genomes generated at the Full Run time-point, 341 called SNPs identically whether they were generated by Medaka or Nanopolish. The majority of the remaining genomes either Medaka or Nanopolish are unable to confidently call a site and so assigns an ambiguous base (N), altering the SNP call. Of more concern, there are some sites which are incorrectly assigned as a

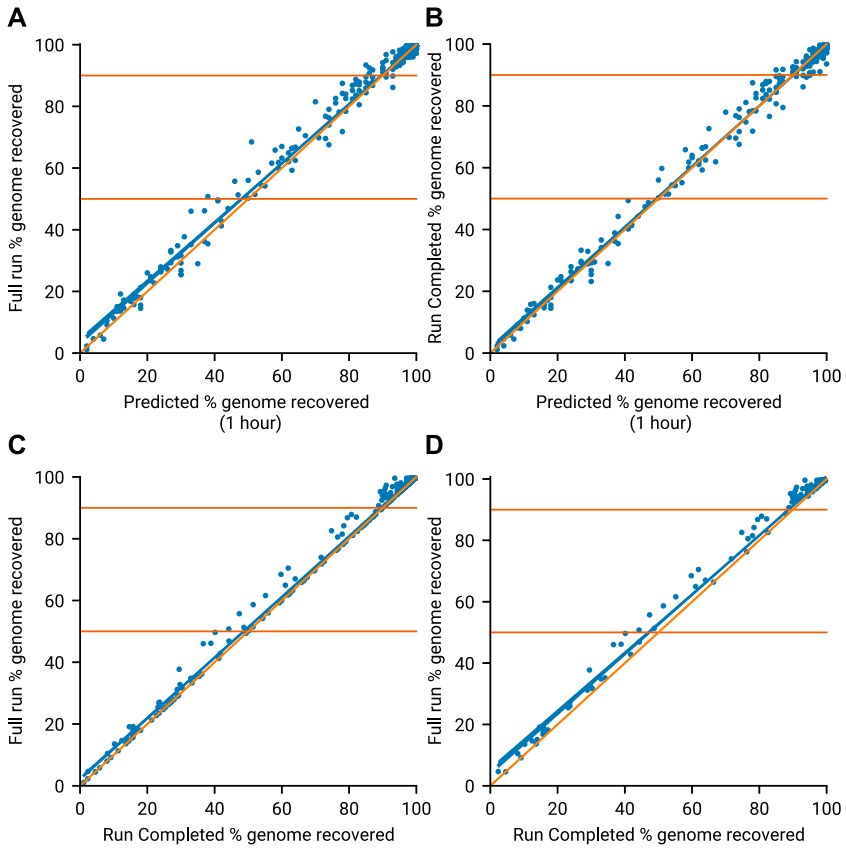


FIGURE 3
Genome recovery throughout all 13 runs. Blue line (line of best fit), Orange line ($x = y$). **(A)** Compares the predicted genome recovery based on 1 hours sequencing (X-axis) with the actual coverage seen at the end of the run (Y-axis). Predicted recovery is the proportion of amplicons in a sample expected to reach 20× coverage. Actual coverage is the proportion of non N bases in the consensus genomes obtained at the end of the run (Full Run). **(B)** Compares the same predicted genome recovery with the actual coverage observed at the Run Complete time as defined by minoTour. **(C)** Compares the actual coverages reported in A (Full Run) and B (Run Complete). **(D)** Is the same as C but ignores those runs where the Run Complete time is the same as the Full Run time.

reference call by Medaka (Table 1), 27 total. Upon inspection, the majority of these are for one single site in the genome at position 28,111 (Figure 5A). We also note one site, 913, for which Nanopolish rarely can call a SNP at lower coverage, but changes to an ambiguous call at higher depth, however this is a very unusual case. It is very infrequent that an increase in coverage over 20× alters a call. An example of this is illustrated at Figure 5B.

At the Run Complete time point, there is an increase in the number of ambiguous (N) sites called by both pipelines, most likely a consequence of the slightly lower coverage data available (Tables 1, 2). However overall the difference between calls made in consensus genomes generated by both pipelines at this time point is very slight (99.7% identical calls). It is worth mentioning that as this is the time point that genomes would finish in a minoTour ARTIC run, we conclude that there is very little effect in using Medaka in our pipeline. There is a very slight increase in the number of SNPs being called by one pipeline being called as an N in the other (1 for Nanopolish SNPs and 3 for Medaka SNPs), although again this is likely due to slightly lower coverage.

The Sample Complete time genomes are of lower quality, with approximately a 5 fold increase in the number of Ns seen in a generated consensus genome (Table 2). However we note that only one sample finishes at this time point in an actual run (the last to reach our completion threshold). When comparing Nanopolish and Medaka genomes at the Sample Complete time point, we can see that there is a very small increase over the Run Complete time point generated genomes in disagreement between the SNP calls (0.00013% of all calls). However the calls are effectively concordant even at this earliest time point, and as previously noted, only one generated genome actually finishes at this time point in an actual run.

Finally we compared the genomes that did not reach our completion threshold in our run, thus lacking a Sample Complete time point. As shown in Table 3, these genomes are of much lower quality, and do not improve by allowing the run to continue to the Full Run length. They are approximately 48% Ambiguous N calls on average, and there is no gain in the average number of SNPs called.

Thus we conclude the majority of SNP call differences between Medaka and Nanopolish are differences in ambiguous calls. Overall, we conclude that Medaka is sufficient for variant calling and lineage assignment, but in our downstream analysis workflows we routinely run both pipelines for confirmation.

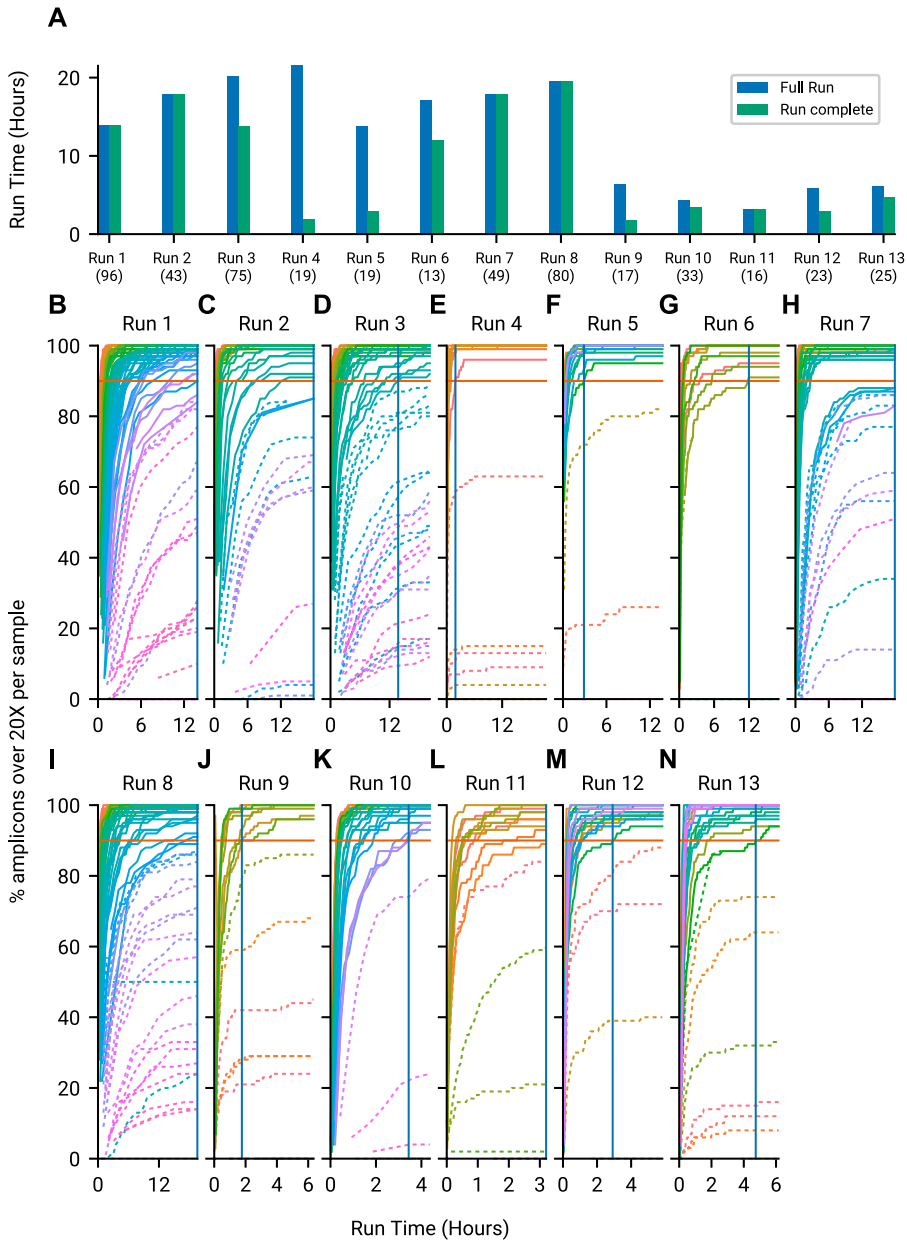


FIGURE 4
Time savings by using minoTours ARTIC Pipeline and amplicon coverages for each run, across the course of the run. **(A)** The Full Run time point plotted against minoTours Run complete time point as hours since the run started, for each run. Number of samples shown in brackets below the run label. **(B–N)** Samples across 13 runs showing the percentage of amplicons at 20× over time. Barcodes that we project to finish are displayed with solid lines, whilst barcodes we project not to finish are dashed. 90% (Our threshold for firing) is marked on each plot. Once all barcodes that are projected to finish cross the 90% threshold, we would instruct MinKNOW to stop the run. This time is marked by a solid blue vertical line.

3.4 SwordFish based adaptive sampling

Sequencing libraries were prepared using the standard ARTIC protocol and a custom set of 1,200 base pair primers (Freed, 2020) (BED file available in Supplementary File S3.5). Sequencing was monitored in real-time using minoTour (Munro et al., 2021) and SARS-CoV-2 samples analysed using minoTours ARTIC pipeline. ReadFish at commit 0ccb5932 (https://github.com/LooseLab/readfish/tree/0ccb59324906635a0d077f94d7f82388039885cb) was used to perform targetted sampling, as unlike ONTs adaptive sampling, experimental configurations can be updated during a run (Payne et al., 2020). MinKNOW was configured to provide data in 0.8 s chunks. Sequencing was performed on a GridION Mk1 (ONT). The method requires Guppy version 4.2 or later for barcode de-multiplexing. Basecalling was performed using the HAC model for final analysis with “require both ends” for de-multiplexing set to true. ReadFish was configured to use fast base calling, requiring barcodes at one end. Starting configuration TOMLs and commands can be found at https://github.com/looselab/swordfish-experiments.

10.3389/fgene.2023.1138582

TABLE 1 Contingency table comparing SNP calling between Medaka and Nanopolish for all three time points. Displayed are total counts across all sites called as either reference (Ref), SNP or unknown (N). Only genomes present in each category (Full Run, Run Completed and Sample Completed) are included in the analysis.

	Medaka								
		Full Run			Run completed			Sample completed		
		N	SNP	Ref	N	SNP	Ref	N	SNP	Ref
Nanopolish	N	151,292	29	418	183,709	32	412	837,973	116	253
	SNP	25	10,727	5	26	10,697	6	78	9,984	13
	Ref	24	0	9,818,580	33	0	9,786,167	37	2	9,132,765

FIGURE 5
SNIPIT plots demonstrating particularly divergent positions for SNP calls between Medaka and Nanopolish. The tracks from top to bottom show SNPs as called from consensus genomes for the Sample Complete time, Run Until and Full run time points (in this order) for both the Medaka and Nanopolish pipelines. **(A)** SNIPIT plot showing an example pair of consensus genomes with Nanopolish calling a SNP at position 28111 but Medaka calling reference. **(B)** SNIPIT plot, showing the Nanopolish pipeline switch from a SNP to an N at position 913 with more data, on a single sample across our three time points.

In our first demonstrative experiment, we selected a range of representative clinical samples (see Supplementary Table S1) with Cycle threshold (Ct) values ranging from 14 to 30, as well as some samples for which no Ct values were available. We utilised the standard ARTIC requirement for reads being barcoded at each end and sequenced using the LSK109 library protocol (ONT). Barcoding at both ends undoubtedly favours downstream analysis as rejected reads will only possess a single starting barcode, and so are assigned as unclassified, Even so, this approach provides an ideal test for throughput and the performance of ReadFish. We ran three separate experiments on our flow cell, visualised in Supplementary Figure S2. The library was not normalised prior to loading, and four barcodes were clearly abundant 58, 64, 76, 88, when no adaptive sampling was applied to the library Supplementary Figure S2A. We began by unblocking based purely on barcode assignment Supplementary

TABLE 2 Contingency table displaying the mean count for sites called as either reference (Ref), SNP or unknown (N), for all three time points (Full Run, Run Completed and Sample Completed), for each SNP calling pipeline. Note, The total of each column, excluding (N. genomes) represents every position in a SARS-CoV-2 genome. Only genomes present in each category (Full Run, Run Completed and Sample Completed) are included in the analysis.

	Medaka			Nanopolish		
	Full run	Run complete	Sample completed	Full run	Run complete	Sample completed
N	453	550	2,509	454	551	2,510
SNP	32	32	30	32	32	30
Ref	29,398	29,301	27,345	29,397	29,300	27,343
N. Genomes	334	334	334	334	334	334

TABLE 3 Contingency table displaying the mean count for sites called as either reference (Ref), SNP or unknown (N). Note, The total of each column, excluding (N. genomes) represents every position in a SARS-CoV-2 genome. Only genomes NOT present in the Sample completed category are included in the analysis.

	Medaka		Nanopolish	
	Full Run	Run Until	Full Run	Run Until
N	14,519	14,813	14,495	14,813
SNP	19	19	19	19
Ref	15,353	15,059	15,374	15,056
N. Genomes	123	120	123	120

Figure S2B. Unexpectedly, even given the 1200bp read lengths, this resulted in the ability to detect 16 more amplicons at 50× coverage on the less abundant barcodes, compared with the control run after 110 min of sequencing, as shown in Supplementary Table S2.

We next used SwordFish to update ReadFish's targets in real-time, based on real-time analysis by minoTour, to provide granular control of individual amplicon/barcode combinations. Using the same library as our previous experiment, we applied a simple threshold approach rejecting reads from amplicon/barcode combinations once coverage exceeded 50×. More sophisticated algorithms for normalisation, for example probabilistic discard, could be considered but would have to account for the large dynamic range of amplicon concentration in samples. As shown in Supplementary Table S2, it is possible to individually address each amplicon/barcode combination to ensure the total coverage does not exceed a predetermined threshold. Inspection of the relative change in amplicon/barcode proportion reveals that some amplicons within abundant barcodes are themselves effectively enriched, suggesting that this targeted approach is better than simple inactivation of entire barcodes. The relative change in proportion of classified amplicon/barcode combinations is slight, as expected for the short amplicons sequenced here (Supplementary Figure S3C, Supplementary Figure S3E and Supplementary Figure S3G). Enrichment efficiency is further reduced by short fragments present within these libraries (Supplementary Figures S4, S5).

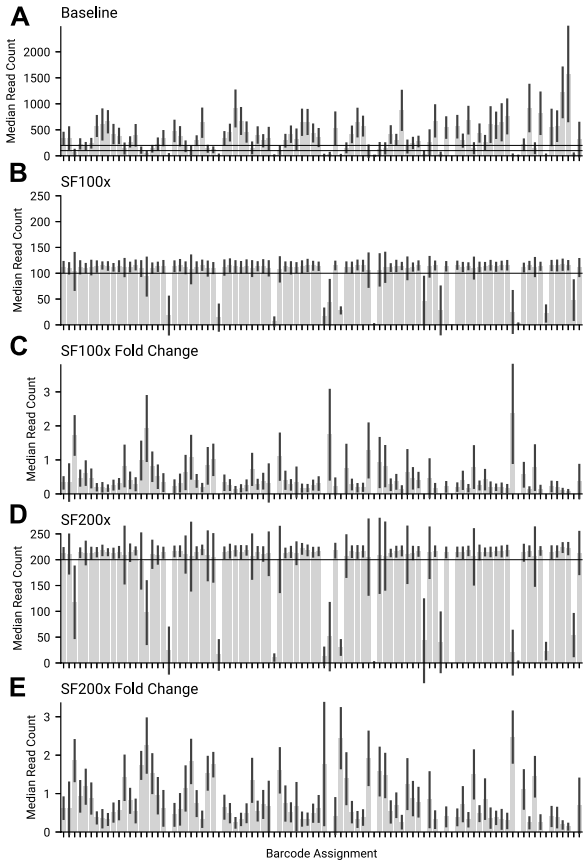


FIGURE 6
The median amplicon read count for each barcode, for the same library across different swordfish threshold targets. **(A)** Baseline, where no adaptive sampling was applied. 100 and 200 are marked as the targets for the other 2 experiments. **(B)** SF100 had a 100× coverage swordfish threshold target. 100× is marked on the graph. **(C)** Fold change for the median amplicon read count per barcode, between SF100x and Baseline. **(D)** SF200 had a 200× coverage threshold target. 200 is marked on the graph. **(E)** Fold change for the median amplicon read count per barcode, between SF200x and Baseline.

The current maximum number of barcodes in a library available for nanopore sequencers is 96, at the time of writing. We proceeded to test our approach against the maximum number of samples, targeting 200× coverage of each sample, running for 6 h on a

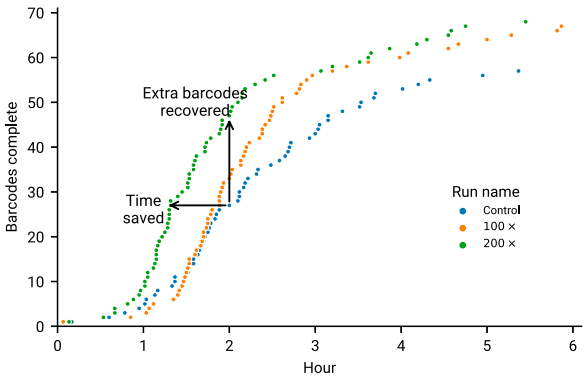


FIGURE 7
Overlapped runs in our second experiment (6 h of sequencing) marking the time at which a barcode reached 90% of amplicons at 20×. It is worth noting the increased performance of 200× is likely due to the increased yield, as this was the first run to go on. Time saved is considered to be points for a run that are shifted left of their equivalents on the *y*-axis. Any points above the Control run (at the same time) are considered as extra barcodes recovered.

MinION Mk1b. Given that our amplicon primer scheme has 29 amplicons, we are tracking a total 2,784 unique amplicon/barcode combinations. We then ran the same library targeting amplicon coverage of 100× for a further 6 h. Finally we ran a 6 h control experiment, with no adaptive sampling. The median amplicon coverages for each sample achieved are displayed in Figure 6. As each condition was run on the same flowcell, after a nuclease flush and reload, there was a small decrease in total sequence yield for each experiment (Supplementary Table S3). Potential maximum enrichment was again brought down by the presence of some short material in the library (Supplementary Figure S6), but ReadFish displayed sufficient performance to keep up with 96 barcodes, with the unblocked read length falling far short of the sequenced. Inspecting some illustrative barcodes (unclassified, 32, 62, 92, 93), we see that indeed our analysis bins all unblocked reads into unclassified, as shown in Supplementary Figure S7. Figure 7 illustrates that using adaptive sampling with ReadFish/Swordfish resulted in an increase in the number of amplicons reaching useful coverages, as well as accelerating the time at which these amplicons reached this coverage. We also see an increase in the number of amplicons that we recover in the SwordFish enabled runs, recovering up to 108 more amplicons at 50× coverage when compared to the control run, as shown in Supplementary Table S4. It is worth noting that these amplicons may have reached this coverage in the control run with more time, as there was a smaller yield in our control run due to having had two experiments run on the flowcell beforehand. Thus, although the effects are relatively small, this approach of individually addressing each amplicon on each barcode in a 96 barcode library will benefit the sequencing run.

In a third set of experiments, Swordfish/Readfish was applied to the midnight protocol 1,200 base pair amplicon scheme using the RBK110.96 rapid library preparation kit, and sequenced on a GridION Mk1. As anticipated, the rapid protocol results in reads shorter than the amplicon length and so we saw no benefit as either a filter to balance barcodes or the speed at which amplicons reached completion Supplementary Figures S8A–C, S9. This experiment was run in triplicate.

Overall, applying adaptive sampling to ARTIC SARS-CoV-2 sequencing reveals the fundamental challenges of enriching short material. Reads must be long enough to benefit from time saved by rejecting unwanted reads. Effectively this application is more of a simple filter to remove unnecessary excess reads with minimal enrichment benefits. Longer read lengths would improve enrichment capabilities, but are less useful for viral amplicon sequencing due to the risk of drop out. In the future, as flow cell yield increases, and these features become available on the PromethION, this approach will enable dynamic adjustment of yields obtained from individual samples in barcoded libraries. The model presented here relies on real-time analysis of the data obtained to determine if an experimental objective has been achieved. Any method that does not consider the final data risks bias as a result of unexpected read length distribution differences between barcoded samples.

4 Conclusion

We demonstrate that by reducing the run time for a SARS-CoV-2 sequencing run using real-time analysis to calculate the best stopping point, it is possible to balance flow-cell health and time to answer while minimising any information loss. Significant time savings are possible using this approach; this has previously been described as “Run Until”, a method described by Oxford Nanopore Technologies but to date, not widely used. We show that stopping a run at the earliest point where sufficient data are available does not negatively affect subsequent downstream analysis. In addition, Read Until provides further benefits to a SARS-CoV-2 sequencing run, by reducing the number of unnecessary reads in the analysis, reducing the time taken to complete individual genomes and focusing sequencing capacity on incomplete samples. Real-time analysis in conjunction with adaptive sampling demonstrates powerful balancing of amplicon coverage on up to 96 samples, even providing limited enrichment in some cases. In order for this method to work, amplicons must be sufficiently long. Whilst the current maximum barcode number is 96, we anticipate this approach being able to handle many more samples.

Data availability statement

The original contributions presented in the study are publicly available. Source code and documentation is available at https://github.com/LooseLab/minotourapp and https://github.com/LooseLab/swordfish Supplementary data are available from https://github.com/LooseLab/artic_minotour_analyses.

Author contributions

RM, ML, and AP wrote and performed analysis for this Manuscript. RS worked substantially on the MinoTour platform. NH, MC, CM, LS, GN, TW, CA, and JT gathered data, created sequencing libraries and performed sequencing for this manuscript.

Funding

Work on minoTour has been funded by BBSRC (BB/M020061/1) as well as additional support from the Defence Science and Technology Laboratory (DSTLX-1000138444). RM is supported by a BBSRC iCASE studentship. The sequencing data used to develop the ARTIC components of minoTour were generated as part of COG-UK, itself supported by funding from the Medical Research Council (MRC) part of UK Research and Innovation (UKRI), the National Institute of Health Research (NIHR) (grant code: MC_PC_19027), and Genome Research Limited, operating as the Wellcome Sanger Institute.

Acknowledgments

We thank the molecular, virology, and bacteriology staff and microbiologists of the British Columbia Centre for Disease Control Public Health Laboratory (BCCDC PHL) for their contribution toward the implementation and testing of SARS-CoV-2 genomic sequencing in British Columbia. We thank Rebecca Hickman for sequencing library production for testing.

Conflict of interest

ML was a member of the MinION access program and has received free flow cells and sequencing reagents in the past. ML has received reimbursement for travel, accommodation and conference fees to speak at events organised by Oxford Nanopore Technologies. JT was a member of the MinION access program and has received free flow cells and sequencing reagents in the past. JT has received reimbursement for travel, accommodation and conference fees to speak at events organised by Oxford Nanopore Technologies.

Publisher's note

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fgene.2023.1138582/full#supplementary-material

References

Bruno, A., Aury, J. M., and Engelen, S. (2021). BoardION: Real-time monitoring of Oxford nanopore sequencing instruments. *BMC Bioinforma.* 22 (1), 245. issn: 245. doi:10.1186/s12859-021-04161-0

Cog (2020). "An integrated national scale SARS-CoV-2 genomic surveillance network". In: *Lancet Microbe* 1 (3), e99–e100. issn: 2666-e100. doi:10.1016/S2666-5247(20)30054-9

Ferguson, J. M., Gamaarachchi, H., Nguyen, T., Gollon, A., Tong, S., Aquilina-Reid, C., et al. (2021). InterARTIC: An interactive web application for whole-genome nanopore sequencing analysis of SARS-CoV-2 and other viruses. *Bioinformatics* 38 (5), 1443–1446. issn: 1367-4803. doi:10.1093/bioinformatics/btab846

Freed, N. E., Vlkova, M., Faisal, M. B., and Silander, O. K. (2020). "Rapid and inexpensive whole-genome sequencing of SARS-CoV-2 using 1200 bp tiled amplicons and Oxford Nanopore Rapid Barcoding". In: *Biol. Methods Protoc.* 5 (1), bpaa014. doi:10.1093/biomethods/bpaa014

Gardy, J. L., and Loman, N. J. (2018). Towards a genomics-informed, real-time, global pathogen surveillance system. *Nat. Rev. Genet.* 19, 9–20. issn: 1471-0064. doi:10.1038/nrg.2017.88

Jain, M., Olsen, H. E., Paten, B., and Akeson, M. (2016). "The Oxford nanopore MinION: Delivery of nanopore sequencing to the genomics community. In: *Genome Biol.* 17, p. 239 (Electronic). doi:10.1186/s13059-016-1103-0

Li, H. (2018). Minimap2: Pairwise alignment for nucleotide sequences. *Bioinformatics* 34 (18), 3094–3100. issn: 1367-4803. doi:10.1093/bioinformatics/bty191

Loman, N. J., Quick, J., and Simpson, J. T. (2015). A complete bacterial genome assembled de novo using only nanopore sequencing data. *Nat. Methods* 12 (8), 733–735. issn: 1548-7105. doi:10.1038/nmeth.3444

Minh, B. Q., Schmidt, H. A., Chernomor, O., Schrempf, D., Woodhams, M. D., von Haeseler, A., et al. (2020). IQ-TREE 2: New models and efficient methods for phylogenetic inference in the genomic era. *Mol. Biol. Evol.* 37 (5), 1530–1534. issn: 0737-4038. doi:10.1093/molbev/msaa015

Munro, R., Payne, A., Forey, T., Osei, S., Holmes, N., et al. (2021). minoTour, real-time monitoring and analysis for nanopore sequencers. *Bioinformatics* 38 (4), 1133–1135. issn: 1367-4803. doi:10.1093/bioinformatics/btab780

Nicholls, S. M., Poplawski, R., Bull, M. J., Underwood, A., Chapman, M., Abu-Dahab, K., et al. (2021). "CLIMB-COVID: Continuous integration supporting decentralised sequencing for SARS-CoV-2 genomic surveillance". In: *Genome Biol.* 22 (1), p. 196. issn: 1474-760X. doi:10.1186/s13059-021-02395-y

O'Toole, A., Scher, E., Underwood, A., Jackson, B., Hill, V., McCrone, J. T., et al. (2021). "Assignment of epidemiological lineages in an emerging pandemic using the pangolin tool". In: *Virus Evol.* 7.2. issn: veab064-1577. doi:10.1093/ve/veab064

Payne, A., HolmesNov, N., Munro, R., Debebe, B. J., and Loose, M. (2020). Readfish enables targeted nanopore sequencing of gigabase-sized genomes. *Nat. Biotechnol.* 39 (4), 442–450. doi:10.1038/s41587-020-00746-x

Payne, A., Munro, R., Holmes, N., Moore, C., Matt, C., et al. (2021). Barcode aware adaptive sampling for Oxford Nanopore sequencers. *BioRXiv*. doi:10.1101/2021.12.01.470722

Pedersen, B. S., and Aaron, R. Q. (2018). Mosdepth: Quick coverage calculation for genomes and exomes. *Bioinformatics* 34 (5), 867–868. issn: 1367-4803. doi:10.1093/bioinformatics/btx699

Quick, J., Nathan, D. G., Pullan, S. T., Claro, I. M., Smith, A. D., Gangavarapu, K., et al. (2017). Multiplex PCR method for MinION and Il-lumina sequencing of Zika and other virus genomes directly from clinical samples. *Nat. Protoc.* 12, 1261–1276. issn: 1750-2799. doi:10.1038/nprot.2017.066

Quick, J., Loman, N. J., Duraffour, S., Simpson, J. T., Severi, E., Cowley, L., et al. (2016). "Real-time, portable genome sequencing for Ebola surveillance". In: *Nature* 530, pp. 228–232. issn: 1476-4687. doi:10.1038/nature16996

Rambaut, A. (2021). GitHub - rambaut/figtree.js: Phylogenetic tree library for JavaScript/Node.js. Available at: https://github.com/rambaut/figtree.js/.

Tyson, J. R., James, P., Stoddart, D., Sparks, N., Wickenhagen, A., Hall, G., et al. (2020). Improvements to the ARTIC multiplex PCR method for SARS-CoV-2 genome sequencing using nanopore. *bioRxiv* 2020, 283077. doi:10.1101/2020.09.04.283077

Walt, S., and Varoquaux, G. (2011). The NumPy array: A structure for efficient numerical computation. *Comput. Sci. Eng.* 13, 22–30. doi:10.1109/MCSE.2011.37

frontiers | Frontiers in Genetics

TYPE Original Research
PUBLISHED 03 April 2023
DOI 10.3389/fgene.2023.1181956

OPEN ACCESS

EDITED BY
Nathan Olson,
National Institute of Standards and Technology (NIST), United States

REVIEWED BY
Zhibin Lv,
Sichuan University, China
Feifei Cui,
Hainan University, China

*CORRESPONDENCE
Wen Zhu,
syzhuwen@163.com

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

RECEIVED 08 March 2023
ACCEPTED 24 March 2023
PUBLISHED 03 April 2023

CITATION
Zhang X, Zhu W, Sun H, Ding Y and Liu L (2023), Prediction of CTCF loop anchor based on machine learning.
Front. Genet. 14:1181956.
doi: 10.3389/fgene.2023.1181956

Prediction of CTCF loop anchor based on machine learning

Xiao Zhang[1,2,3], Wen Zhu[1,3]*, Huimin Sun[4], Yijie Ding[3] and Li Liu[2]

[1]School of Mathematics and Statistics, Hainan Normal University, Haikou, China, [2]Yangtze Delta Region Institute (Quzhou), University of Electronic Science and Technology of China, Quzhou, China, [3]Key Laboratory of Computational Science and Application of Hainan Province, Haikou, China, [4]School of Physical Science and Technology, Inner Mongolia University, Hohhot, China

Introduction: Various activities in biological cells are affected by three-dimensional genome structure. The insulators play an important role in the organization of higher-order structure. CTCF is a representative of mammalian insulators, which can produce barriers to prevent the continuous extrusion of chromatin loop. As a multifunctional protein, CTCF has tens of thousands of binding sites in the genome, but only a portion of them can be used as anchors of chromatin loops. It is still unclear how cells select the anchor in the process of chromatin looping.

Methods: In this paper, a comparative analysis is performed to investigate the sequence preference and binding strength of anchor and non-anchor CTCF binding sites. Furthermore, a machine learning model based on the CTCF binding intensity and DNA sequence is proposed to predict which CTCF sites can form chromatin loop anchors.

Results: The accuracy of the machine learning model that we constructed for predicting the anchor of the chromatin loop mediated by CTCF reached 0.8646. And we find that the formation of loop anchor is mainly influenced by the CTCF binding strength and binding pattern (which can be interpreted as the binding of different zinc fingers).

Discussion: In conclusion, our results suggest that The CTCF core motif and it's flanking sequence may be responsible for the binding specificity. This work contributes to understanding the mechanism of loop anchor selection and provides a reference for the prediction of CTCF-mediated chromatin loops.

KEYWORDS

CTCF, Chromatin Loop, Machine Learning, DNA sequence, 3D Genome

1 Introduction

High-order chromatin structure influences a variety of biological processes in the nucleus, including gene transcription, gene regulation and DNA replication. The structure of interphase chromatin has been extensively researched with the development of various chromatin conformation capture techniques (Fullwood et al., 2009a; Fullwood et al., 2009b; Lieberman-Aiden et al., 2009; Hsieh et al., 2015), unveiling the functional units. For example, extensive researches on chromosome compartments (Dixon et al., 2012), topologically associated domains (TADs) (Rao et al., 2014) and loops (Narendra et al., 2015) have been carried out. Chromatin loops usually form between the locus that separated by hundreds of thousands base pairs. These long-range interactions usually form a local chromatin structure. According to previous studies, the destruction of these loops leads to a significant imbalance in nearby gene expression (Lupiáñez et al., 2015; Hnisz et al.,

2016). The binding sites of CTCF (an 11 zinc finger DNA binding protein) frequently occur on the boundaries of loops and topologically associated domains, which highlights the importance of CTCF binding for the loop formation.

In the process of gene expression, the gene regulatory elements work in order. These regulatory elements can be classified as promoters, enhancers, insulators, and other regulatory sequences (West et al., 2002; Kellis et al., 2014). Insulators protect genes in cells from inappropriate regulatory signals from adjacent chromatin environments, and play an important role in cell type-specific gene expression (Liu et al., 2019). CTCF was originally thought to be an active chromatin-labeled insulator. As an evolutionarily conserved zinc finger family transcription factor, CTCF was discovered for the first time in the chicken gene promoter (Bell and Felsenfeld, 2000). CTCF was found to be related to blocking the activity of enhancers in the process of transcription (Liu et al., 2021). Changes in the CTCF protein and its binding sites on insulators are linked to a variety of human diseases. For example, deletion of CTCF in the domain may result in an interaction between the enhancer and a glioma oncogene (Katainen et al., 2015); the binding site of CTCF is the main mutation hot spot of the non-coding cancer genome (Ohlsson et al., 2001); zinc finger mutation or abnormal target selective methylation destroy the spectrum of target specificity and is related to cancer (Phillips and Corces, 2009).

CTCF was later found to play an important role in chromatin organization. Paired CTCFs binding act as loop anchors to limit the interaction between remote regulatory elements (de Wit et al., 2015; Rowley and Corces, 2018). As a result, how to distinguish the interacting CTCF pair and the non-interacting CTCF pair is a critical issue. Many experiments have revealed that the interaction between CTCF and cohesin is crucial for loop formation (Wutz et al., 2017). This interaction establishes a dynamic chromatin loop between remote CTCF binding sites to drive the formation of TADs. The chromatin loops may form through the process of loop extrusion (Alipour and Marko, 2012; Barbieri et al., 2012; Fudenberg et al., 2016; Haarhuis et al., 2017; Rao et al., 2017; Davidson et al., 2019; Kim et al., 2019). Cohesin can pass through and extrude DNA to form chromatin loops until it is blocked by CTCF. In addition, the formation of the loop can also be realized through other mechanisms (Brackley et al., 2013; Bianco et al., 2018; Conte et al., 2020). Although the formation mechanism of the loop has been deeply studied, the ability of the model based on polymer physics to predict a single CTCF loop has not been systematically evaluated (Di Pierro et al., 2016; Kai et al., 2018). The machine learning model named Lollipop uses 77 features of the genome and epigenome to predict the interaction of CTCF pairs (Lv et al., 2021). Deep-loop uses only DNA sequences to predict CTCF-mediated chromatin loops (Xi and Beer, 2021). The loop extrusion and competition model can predict the specificity of CTCF interaction through four characteristics. These four characteristics are chromatin loop competition, CTCF binding site distance, CTCF motif and CTCF binding intensity. The aforementioned experiments aim to predict the loops formed between pairs of CTCF binding sites and require the CTCF ChIP-seq data as input. In mammalian cells, there are approximately 50,000 CTCF binding sites, corresponding to more than one million possible CTCF pairs separated by less than 1 Mb. However, Hi-C or ChIA-PET measurements revealed that only approximately 2%–5% of CTCF pairs are directly interacting. This increases the difficulty of *de novo* prediction task. We notice that only a portion of CTCF binding sites are used as loop anchors. Can we first distinguish the loop anchor and non-anchor to reduce the search space for loop identification? Motivated by this idea, we intend to determine if a single CTCF binding site may serve as the anchor of loop by using sequence and binding intensity features. We find the binding intensity of CTCF, the core motif and the flanking sequence of the motif all have an important influence. Previous models ignore the flanking sequence features of the CTCF motif. In this paper, we developed support vector machine (SVM) (Guo et al., 2008; Zhang and Liu, 2017), convolutional neural network (CNN) (Li and Liu, 2020; Cui et al., 2021), random forest (RF) (Xu et al., 2019; Dao et al., 2022), linear discriminant analysis (LDA), Naive Bayes (NB), logistic regression (LR) (Yang et al., 2021) and stochastic gradient descent (SGD) model to predict the potential of CTCF binding to form chromatin loop anchors. We considered the binding intensity of CTCF, the sequence characteristics of the CTCF core motif and the flanking sequence as input features. These features performed well in almost all the models, indicating that they are important for the formation of loop anchors.

2 Materials and methods

2.1 Data source

We download the public ChIP-seq data of CTCF from the ENCODE database (Ecker et al., 2012). The detection method is ChIP-seq, the target set is transcription factors, the biological sample term is GM12878, the reference genome is hg19, and the file type is bed narrowPeak. We also downloaded ChIA-PET data of CTCF from ENCODE. The detection method is ChIA-PET, the target set is transcription factors, the biological sample term is GM12878, the reference genome is hg19, and the file type is fastq.

2.2 Data processing

The positive and negative set was constructed as follows: First, ChIA-PET data of CTCF were preprocessed by using ChIA-PET2 (Li et al., 2017). ChIA-PET2 can significantly improve the sensitivity and reproducibility of detecting chromatin loops while maintaining the same false discovery rate. We can calculate the false discovery rate of each ChIA-PET data by ChIA-PET2. The false discovery rate refers to the expected value of the proportion of the number of falsely rejected true assumptions compared to the number of rejected original assumptions. The false discovery rate offers several advantages, including flexible adjustment of its value, clear meaning, and its ability to be used as an evaluation metric for screened different variables. The ChIA-PET data of CTCF will give a pair of DNA anchors that can form chromatin loops. Here, the data with FDR < 0.05 are considered as the CTCF-mediated chromatin loops. In addition, we focus on whether single CTCF site can form loop anchor. ChIA-PET data consists of a combination of two anchors, which can result in a single anchor corresponding to multiple other anchors. Therefore, when extracting location data of anchors,

repeated anchors may be generated. Thus, after removing duplicate anchors in the data with the cutoff of FDR < 0.05, we obtained the location data of all CTCF loop anchors in the GM12878 cell line. Secondly, the ChIP-seq data of CTCF were used to obtain the location of the core motif with a length of 19 bp at the corresponding data by storm (Schones et al., 2007). Storm scans the input sequences and find the fragment with the highest motif score. Importantly, it also provides the information about whether sequence fragment occurs on the sense chain or antisense chain. This data will later be crucial for extracting sequences of CTCFs in the same binding direction. Finally, The motif location data were compared with the CTCF loop anchors. If there is an overlap between them, it is considered that the binding site of the CTCF can form loop. Then, the position data of these motifs are taken as the positive set and add the label "1". The position data of non-overlapping motifs are taken as the negative set that cannot form a loop, and the label "0"is added. In the GM12878 cell line, the number of samples that cannot form loops is 26,765, the number of samples that can form loops is 22,191, and the total number of samples is 48,956.

The context sequence of CTCF, in addition to its core motif, is crucial for controlling gene expression. Huang et al. (2021) used the SOX2 gene reporting system of mouse embryonic stem cells to study how the context sequence of the CTCF binding site regulates insulator function. They discovered the following: 1) The 10–20 bp sequence upstream of the core motif of CTCF rather than the core motif itself determines whether CTCF can perform the insulator function 2) The insulating effect depends on the number of CTCF tandem binding sites. These findings provide new insights into the classification of CTCF binding sites. The binding and dissociation of CTCF on the genome is a dynamic process. The residence time of CTCF is determined by the binding stability. CTCF has 11 zinc finger structures. The zinc finger ZF3-ZF7 binds with the core motif, and ZF9-ZF11 binds 10–20 bp upstream of the core motif. The existence of ZF8 as a linker also plays an important role in promoting the overall binding stability (Soochit et al., 2021). In the above experiment, when each flanking sequence of the motif gradually decreases from 60 bp to 20 bp, the insulation effect does not decrease significantly, and the strong insulation effect of CTCF always exists. However, when the flanking sequence of the core motif gradually decreases to 10 bp, the strong insulation effect of CTCF is significantly reduced. This demonstrates that the flanking sequences 10–20 bp from core motif has a significant effect on the insulation effect. Furthermore, the bases upstream and downstream of the motif will have a great impact on the function of CTCF. Therefore, we added 20 bp upstream and downstream to the CTCF motif and obtained the location data of the 59 bp sequence. Because the binding of CTCF is directional, the sequence direction should be taken into account when extracting sequences. One-hot encoding is used to make the 59 bp sequence fragment into a matrix consisting of 0 and 1, where base A corresponds to (1,0,0,0), base T corresponds to (0,1,0,0), and bases C and G correspond to (0,0,1,0) and (0,0,0,1), respectively. Then, a 48,956 × 236 one-hot matrix is obtained.

The binding intensity of CTCF can affect the movement of cohesin, thereby affecting the formation of the loops. The narrowPeak ChIP-seq data gives the CTCF binding intensity at the corresponding position. There is a large variation among the CTCF binding intensity values. It will greatly affect the training of the model. Due to the unique characteristics of each assessment index, a multi-index evaluation system typically has different dimensions and orders of magnitude. If the original indicator values are used for analysis when there is significant variation between the indicators, the importance of the indicators with higher values will be accentuated, while the significance of the indicators with lower values will be substantially diminished. Therefore, data normalization is required for the CTCF binding intensity to reduce the impact of the large variation in the training model. Here, we took the logarithm base two of the CTCF binding intensity value to narrow the gap between the value of CTCF binding intensity with the sequence data. Finally, the normalized value of the CTCF binding intensity and one-hot matrix were merged to construct the feature matrix (Supplementary Material).

2.3 Summary of the machine learning model

The basic motivation of the support vector machine (SVM) is to find a decision hyper plane to maximize the interval between the two types of data, construct an objective function according to the maximum interval, and then transform it into its dual problem for solution. For non-linear problems, first use a transformation $z = \varphi(x)$ to map x to a new feature space z, then transform it into the dual problem of support vector machine, and we use radial basis functions as kernel functions.

The random forest (RF) algorithm is an ensemble algorithm composed of multiple decision tree classifiers, with each subclassifier being a CART classification regression tree. Therefore, random forest can perform both classification and regression. The risk of overfitting can be reduced by averaging the decision trees.

Convolutional Neural Network (CNN) is a specialized type of artificial neural network commonly used in deep learning for analyzing visual imagery. It is designed to automatically and adaptively learn spatial hierarchies of features from input images or other two-dimensional data, such as audio spectrograms. CNNs are composed of multiple convolutional layers that apply mathematical operations called convolution to the input data, followed by pooling layers that reduce the dimensionality of the output from the convolutional layers. The output of the pooling layers is then fed into fully connected layers, which perform the final classification or regression of the input data.

We also use other machine learning models to train on the same dataset, including the linear discriminant analysis (LDA), Naive Bayes (NB): The Naive Bayes method is a classification technique that is based on Bayes' theorem and the assumption of independently occurring features, logistic regression (LR): logistic regression is a generalized linear regression that utilizes logistic functions, and stochastic gradient descent (SGD): stochastic gradient descent is an iterative optimization algorithm used to update a model's parameters based on the steepest descent direction of the loss function.

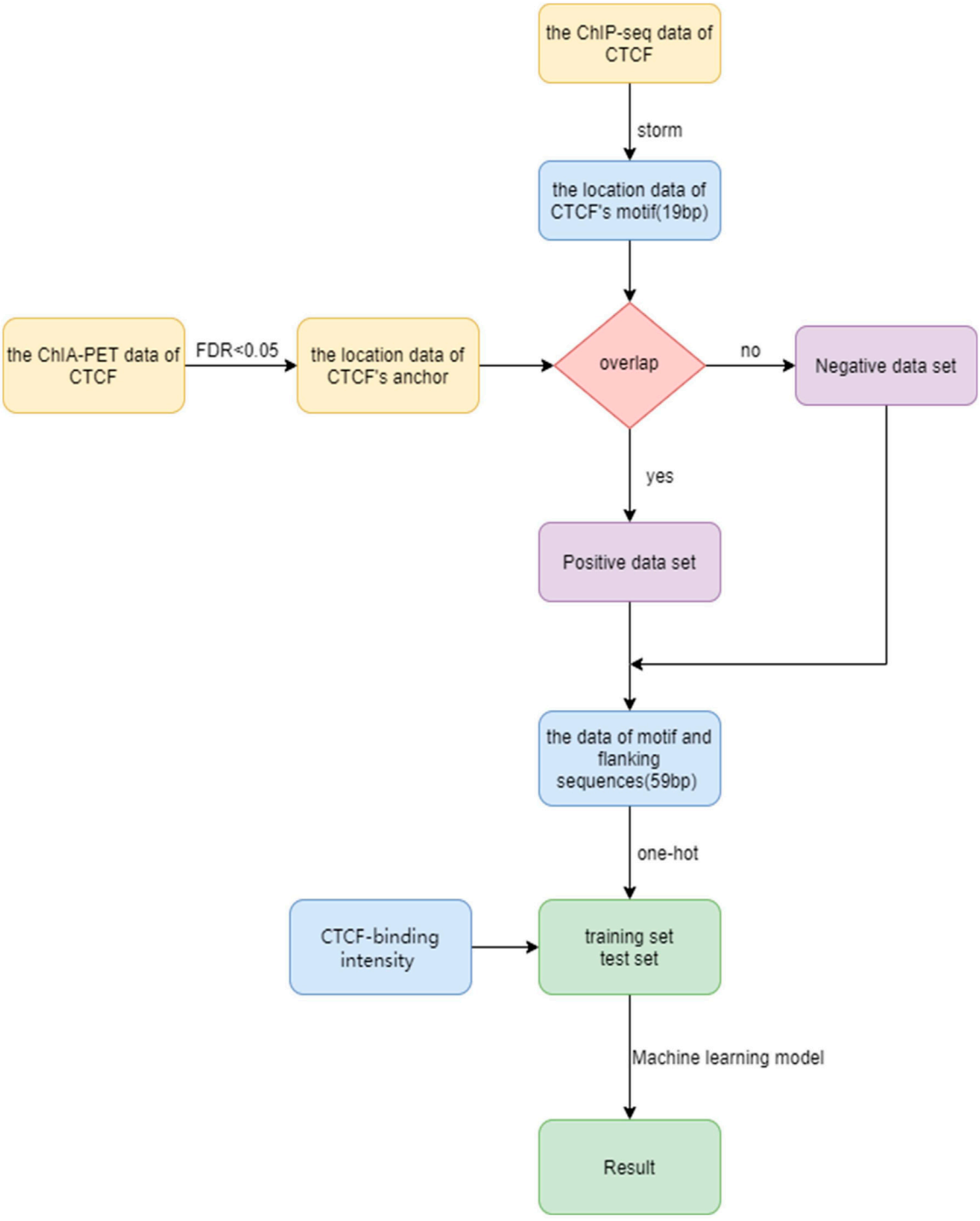


FIGURE 1
Flow chart for CTCF loop anchor prediction.

2.4 Performance assessment

We have employed several machine learning models to train on the same dataset, including linear discriminant analysis (LDA), support vector machines, Random Forest (RF)), logistic regression (LR), stochastic gradient descent (SGD), Naive Bayes (NB) models, convolutional neural networks (CNN), support vector machine models (SVM). We then compared the results of each model.

To evaluate the prediction performance of the model, Accuracy (Acc), Precision (Pre), F1-score (F1), Area Under ROC Curve (AUROC), Area Under PRC Curve (AUPRC),

Specificity (Sp), Sensitivity (Sn) and Matthews correlation coefficient (MCC) are used as evaluation indicators (Zhang Q. et al., 2022; Zhang Z. Y. et al., 2022; Han et al., 2022; Yang et al., 2022). TP (True Positive): successful prediction of positive samples as positive. FP (False-Positive): incorrectly predicts negative samples to be positive. TN and FN correspond to the value of the negative set.

The ratio of correctly classified positive samples in the total number of positive samples:

$$S_n = \frac{TP}{TP + FN} \tag{1}$$

The ratio of correctly classified negative samples in the total number of negative samples:

$$S_p = \frac{TN}{TN + FP} \tag{2}$$

MCC, which has a value range of [- 1,1], is simply a correlation coefficient that describes the relationship between actual classification and prediction classification. A score of 1 denotes the subject's perfect prediction, a value of 0 denotes that the prediction result is less accurate than a random prediction, and a value of −1 denotes that there is no consistency between the predicted classification and the actual classification:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{3}$$

The ratio of the sample size of correctly classified positive samples to the total number of samples predicted by the model as positive samples:

$$\mathrm{Pr}\,ecision = \frac{TP}{TP + FP} \tag{4}$$

The F1 score is the harmonic average of precision and recall:

$$F1 = 2 \times \frac{\mathrm{Pr}\,ecision \times \mathrm{Re}call}{\mathrm{Pr}\,ecision + \mathrm{Re}call} \tag{5}$$

3 Results

3.1 Overview of CTCF loop anchor prediction

In order to predict the CTCF loop anchor, we propose a computational framework (Figure 1). The framework includes dataset construction, feature extraction, and machine learning algorithm selection. We first establish the precise location of the CTCF binding sites based on ChIP-seq data and motif scanning. The positive and negative sets are then generated based on ChIA-PET data. The feature matrix is constructed by extracting sequence of the core motif, flanking sequence, and CTCF binding strength. The machine learning methods are implemented on the feature matrix to distinguish the loop anchor and non-anchor. More details of the framework are discussed in the Materials and methods section.

3.2 Comparison of prediction performance

We trained these machine learning models by using tenfold cross-validation, and then tested the prediction performance on a separated independent testing set. The training dataset is randomly divided into K subgroups of the same size for the K-fold cross validation test. The remaining K-1 folds are utilized as the training dataset for the machine learning model, while one fold is used as the validation dataset. Each fold serves as the validation dataset once this procedure has been repeated K times. One-10th of the dataset is used as an independent testing set, and the rest is considered as a training set. The training set was used to perform ten-fold cross-validation and train the model, and then the model's performance is verified on the test set. We compared the predictive performance of seven machine learning models on independent test sets by evaluating Sn, Sp, Pre, Acc, MCC, F1, AUROC, and AUPRC (Figure 2). Except for the Naive Bayes model, the accuracy rates of the other models are greater than 0.85, with the support vector machine model having the highest accuracy rate of 0.8646. As shown in Figure 3, the AUROC and AUPRC values of the other models (in addition to the naive bayes model) are around 0.92, and they perform well in terms of the remaining F1 score, precision, and other evaluation criteria. This demonstrates that the three types of features we selected have a good predictive effect across a variety of different machine learning models. The good performance of the selected features indicates they have an important influence on the process of CTCF binding to form chromatin loops.

3.3 Importance of features

We found no significant difference in the prediction results obtained from the SVM, SGD, RF, LR, LDA, and CNN models. So we selected the SVM model, which had a slight advantage in results, and used different features and feature combinations for prediction. To assess the contribution of various features or feature combinations to the prediction, we used the core motif, flanking sequence, and their combination with CTCF binding intensity as the features to perform prediction (Table 1). We have discovered that CTCF binding intensity alone has good predictive performance, indicating its important role in the loop formation process. Although the flanking sequence has slightly lower prediction accuracy compared to the core motif, the combinations of the flanking sequence and CTCF binding intensity yields a marginally better outcome than the combination of core motif and binding intensity. This demonstrates that the CTCF binding intensity and core motif features are somewhat redundant. Adding the flanking sequence feature to the model can increase prediction accuracy.

We next try to find the key sites that play an important role in the classification. We calculated the information content of each site by using weblogo (Crooks et al., 2004). From Figure 4, we can see that in the flanking sequence of the core motif, there are obvious differences between the positive set and negative set of sequence data, which is consistent with previous views (Huang et al., 2021; Soochit et al., 2021). The flanking sequence also plays an important role in the process of CTCF binding, and it will affect whether CTCF can be used as the loop anchor. Comparing

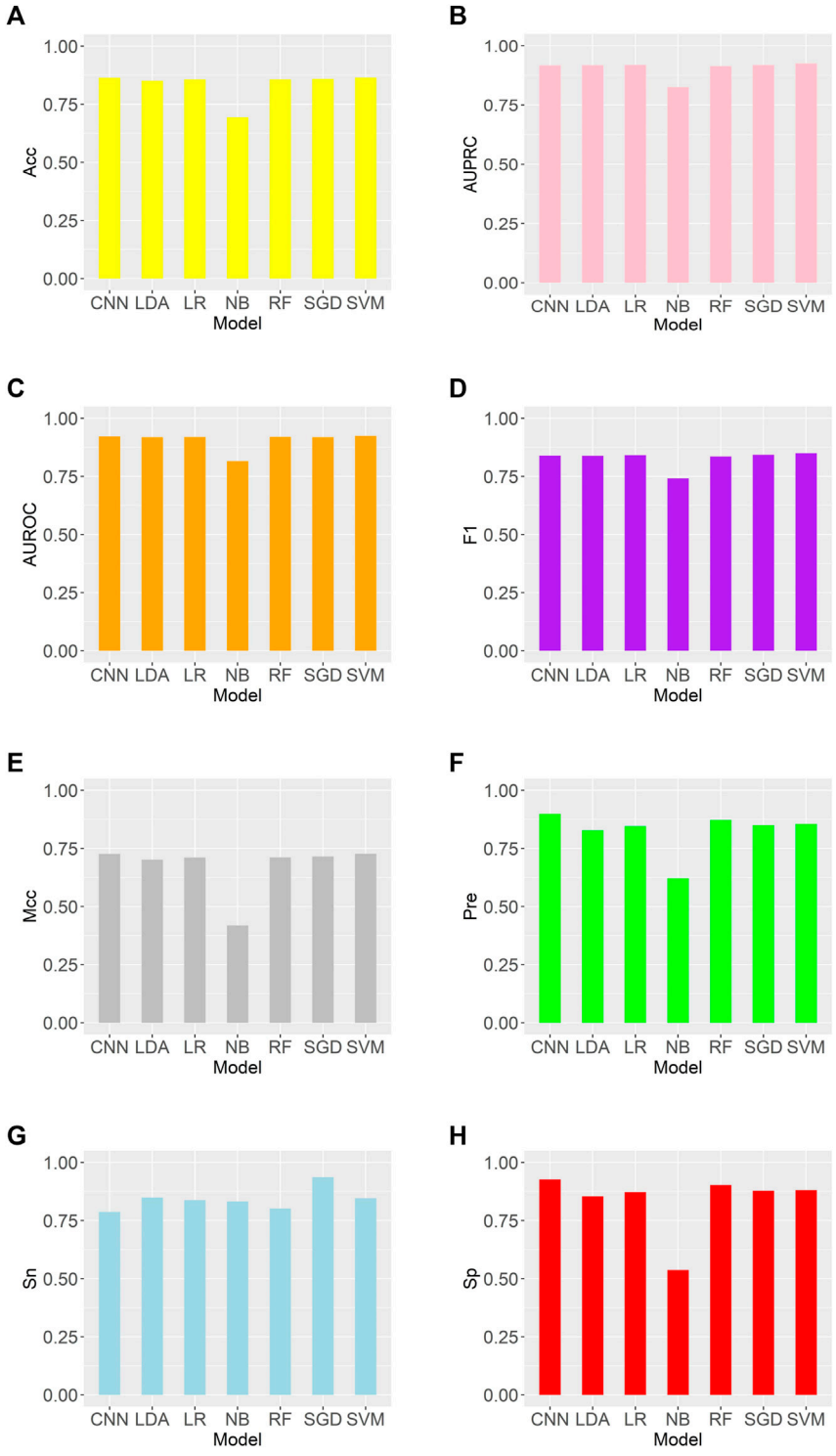


FIGURE 2
Comparison of evaluation criteria between support vector machine and other machine learning models **(A–H)** are as follows: Acc, AUROC, AUPRC, F1, MCC, Pre, Sn, Sp.

the information content shows that positive sets prefer certain flanking sequence sites: 7–9, 12, 14, and 42–48. This further reveals that the flanking sequence feature can effectively distinguish chromatin loop anchor.

In order to determine the importance of each feature more precisely, we performed feature selection, sometimes referred to as feature subset selection (FSS). It alludes to the process of choosing N characteristics from the already-existing M features to optimize the

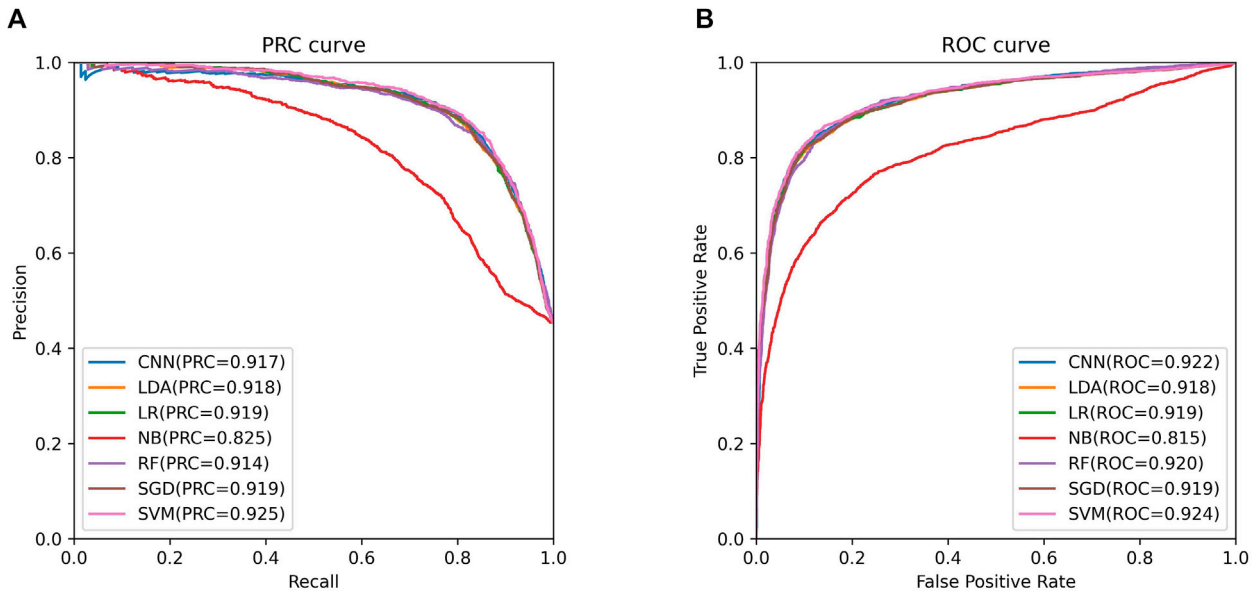


FIGURE 3
ROC and PRC curves of support vector machine model. **(A)**. The PRC curve of support vector machine model. **(B)**. The ROC curve of support vector machine model.

TABLE 1 Prediction performance of different feature and combinations of features by SVM model.

	Acc	AUROC	AUPRC
Core motif	0.6945	0.7491	0.7184
Flanking sequence	0.6467	0.6855	0.6422
CTCF binding intensity	0.8375	0.8954	0.8916
Core motif and CTCF binding intensity	0.8458	0.9118	0.9061
Flanking sequence and CTCF binding intensity	0.8599	0.9225	0.9226

system's particular indicators. To decrease the dimension of the dataset and enhance the efficiency of the learning algorithm, it is necessary to choose some of the most useful characteristics from the original features. The generating process, evaluation function, stop criteria, and verification procedure are the four main components of the feature selection process. We selected the top 20 features in order of importance, and the most important feature was the value of the CTCF binding intensity corresponding to the sequence. We ranked the significance of the features and find that the flanking sequence sites 12, 45, 46 and 47 significantly contribute to classification. The sites 45 to 47 correspond to CTCF zinc fingers 1 to 3. Based on the analysis

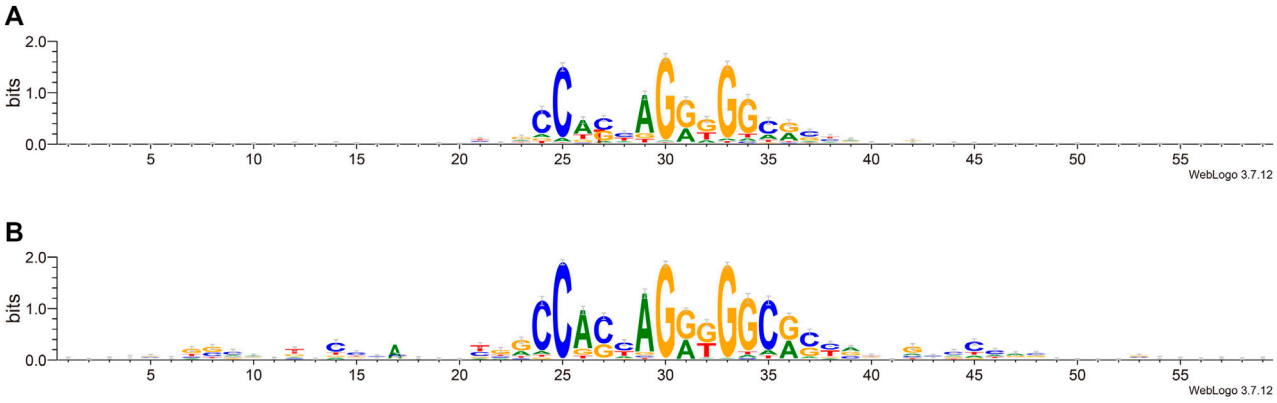


FIGURE 4
Sequence alignment of positive and negative sets. **(A)**. The negative set. **(B)**. The positive set.

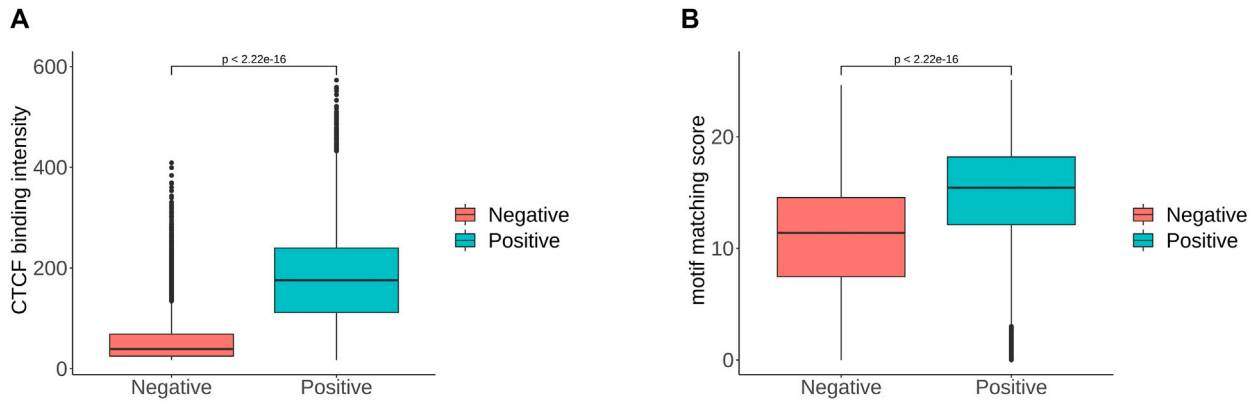


FIGURE 5
(A). CTCF binding intensity of positive and negative sets. **(B)**. CTCF motif matching score of positive and negative sets, where the *p*-value is given by t-test.

combined with Figure 4, there were significant sequence differences observed between the positive and negative sets at the flanking sites 12–17. These sites corresponded to the binding region of zinc finger 8–11 of CTCF. There are 11 zinc fingers in CTCF, and not all of them bind to DNA at the same time. According to the research of Soochit et al. (2021), the removal of zinc finger 8 results in a decrease of chromatin residence time. Our result also suggests that the flanking sequence may influence the residence time of CTCF on DNA. Additionally, we compared the CTCF binding intensity and motif matching score for positive and negative sets The CTCF binding intensity of the positive set was mostly greater than that of the negative set (Figure 5A). The same is true for the CTCF motif matching score calculated by storm (Figure 5B). The motif matching score is a method used to measure the similarity between a query motif and a target motif. The positive set had a higher matching score, suggesting that the binding of CTCF would be more stable. The results support that the stronger the binding intensity, the more it can prevent the movement of cohesin and thus form loops. Therefore, the CTCF binding intensity is indeed an important feature to reflect whether CTCF can be used as the anchor for forming loops.

From the research results, it can be concluded that the most important factor affecting the formation of loop anchor is the CTCF binding intensity. The sequence preference of flanking sequence and the motif matching score are consistent with the different distributions of CTCF binding strength in the positive and negative sets. A more suitable sequence context is favorable to the stable binding of CTCF and makes it simpler to prevent the sliding of cohesin and thus form a loop anchor. The statistical analysis of these three characteristics revealed that CTCF binding strength, core motif, and flanking sequence are the most important factors in predicting loop anchor.

4 Discussion

As an important transcriptional regulation mechanism in organisms, the process of chromatin looping has been widely studied. Previous studies have shown that this process can be interpreted by the loop extrusion model (Xi and Beer, 2021). The details of mechanism are gradually dissected. For instance, the recent study demonstrates that the flanking sequence of CTCF motif have a major impact on the TAD border formation (Huang et al., 2021). Motivated by the experimental results and our statistical analysis, we try to answer which CTCF binding sites may form loop anchors. Our analyses indicate that the CTCF binding intensity, the core motif sequence and the flanking sequence have a certain difference between CTCF loop anchors and non-anchors. Using these features, we employed machine learning models to predict CTCF loop anchors. We conducted ten-fold cross-validation and independent testing, both of which demonstrated the ability of these characteristics to produce accurate prediction results, The statistical analysis showed a significant difference in CTCF binding strength between the positive and negative sets, as well as in the motif matching score. These results indicate that CTCF binding strength can be used as a classification feature. Moreover, this difference may be influenced by the motif and flanker sequences, highlighting their importance as features for predicting CTCF loop anchors. Specifically, based on feature importance ranking, we have identified the flanking sequence sites 12 and 45 to 47, which are likely bound by CTCF ZF8 and ZF1-3, make a significant contribution. This is consistent with other study (Soochit et al., 2021) that the upstream and downstream motifs determine the stability of CTCF binding to DNA. In conclusion, our results suggest that a better sequence context is favorable to the stable binding of CTCF and makes it easier to block loop extrusion by cohesin. Our study provides new insights into the functional classification of CTCF and might even be helpful for the prediction of CTCF-mediated chromatin loops.

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.

Author contributions

XZ: Investigation, Methodology, Writing-Original draft preparation. WZ: Conceptualization, Project administration, Funding acquisition. HS: Investigation, Methodology. YD: Methodology, Writing-Review & Editing, LL: Conceptualization, Funding acquisition, Writing-Review & Editing.

Funding

This work was supported by the National Nature Science Foundation of China (Grant Nos 61863010, 11926205, 11926412, 61873076 and 62272085), National Key R and D Program of China (No. 2020YFB2104400), Natural Science Foundation of Hainan, China (Grant Nos 121RC538, 119MS036, and 120RC588), Key Laboratory of Computational Science and Application of Hainan Province (No. JSKX202201), and the Municipal Government of Quzhou (No. 2022D017).

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/fgene.2023.1181956/full#supplementary-material

References

Alipour, E., and Marko, J. F. (2012). Self-organization of domain structures by DNA-loop-extruding enzymes. *Nucleic Acids Res.* 40, 11202–11212. doi:10.1093/nar/gks925

Barbieri, M., Chotalia, M., Fraser, J., Lavitas, L. M., Dostie, J., Pombo, A., et al. (2012). Complexity of chromatin folding is captured by the strings and binders switch model. *Proc. Natl. Acad. Sci. U. S. A.* 109, 16173–16178. doi:10.1073/pnas.1204799109

Bell, A. C., and Felsenfeld, G. (2000). Methylation of a CTCF-dependent boundary controls imprinted expression of the Igf2 gene. *Nature* 405, 482–485. doi:10.1038/35013100

Bianco, S., Lupiáñez, D. G., Chiariello, A. M., Annunziatella, C., Kraft, K., Schöpflin, R., et al. (2018). Polymer physics predicts the effects of structural variants on chromatin architecture. *Nat. Genet.* 50, 662–667. doi:10.1038/s41588-018-0098-8

Brackley, C. A., Taylor, S., Papantonis, A., Cook, P. R., and Marenduzzo, D. (2013). Nonspecific bridging-induced attraction drives clustering of DNA-binding proteins and genome organization. *Proc. Natl. Acad. Sci. U. S. A.* 110, E3605–E3611. doi:10.1073/pnas.1302950110

Conte, M., Fiorillo, L., Bianco, S., Chiariello, A. M., Esposito, A., and Nicodemi, M. (2020). Polymer physics indicates chromatin folding variability across single-cells results from state degeneracy in phase separation. *Nat. Commun.* 11, 3289. doi:10.1038/s41467-020-17141-4

Crooks, G. E., Hon, G., Chandonia, J. M., and Brenner, S. E. (2004). WebLogo: A sequence logo generator. *Genome Res.* 14, 1188–1190. doi:10.1101/gr.849004

Cui, F., Zhang, Z., and Zou, Q. (2021). Sequence representation approaches for sequence-based protein prediction tasks that use deep learning. *Brief. Funct. Genomics* 20, 61–73. doi:10.1093/bfgp/elaa030

Dao, F. Y., Lv, H., Fullwood, M. J., and Lin, H. (2022). Accurate identification of DNA replication origin by fusing epigenomics and chromatin interaction information. *Res. (Wash D C)* 2022, 9780293. doi:10.34133/2022/9780293

Davidson, I. F., Bauer, B., Goetz, D., Tang, W., Wutz, G., and Peters, J. M. (2019). DNA loop extrusion by human cohesin. *Science* 366, 1338–1345. doi:10.1126/science.aaz3418

De Wit, E., Vos, E. S., Holwerda, S. J., Valdes-Quezada, C., Verstegen, M. J., Teunissen, H., et al. (2015). CTCF binding polarity determines chromatin looping. *Mol. Cell* 60, 676–684. doi:10.1016/j.molcel.2015.09.023

Di Pierro, M., Zhang, B., Aiden, E. L., Wolynes, P. G., and Onuchic, J. N. (2016). Transferable model for chromosome architecture. *Proc. Natl. Acad. Sci. U. S. A.* 113, 12168–12173. doi:10.1073/pnas.1613607113

Dixon, J. R., Selvaraj, S., Yue, F., Kim, A., Li, Y., Shen, Y., et al. (2012). Topological domains in mammalian genomes identified by analysis of chromatin interactions. *Nature* 485, 376–380. doi:10.1038/nature11082

Ecker, J. R., Bickmore, W. A., Barroso, I., Pritchard, J. K., Gilad, Y., and Segal, E. (2012). Genomics: ENCODE explained. *Nature* 489, 52–55. doi:10.1038/489052a

Fudenberg, G., Imakaev, M., Lu, C., Goloborodko, A., Abdennur, N., and Mirny, L. A. (2016). formation of chromosomal domains by loop extrusion. *Cell Rep.* 15, 2038–2049. doi:10.1016/j.celrep.2016.04.085

Fullwood, M. J., Liu, M. H., Pan, Y. F., Liu, J., Xu, H., Mohamed, Y. B., et al. (2009a). An oestrogen-receptor-alpha-bound human chromatin interactome. *Nature* 462, 58–64. doi:10.1038/nature08497

Fullwood, M. J., Wei, C. L., Liu, E. T., and Ruan, Y. (2009b). Next-generation DNA sequencing of paired-end tags (PET) for transcriptome and genome analyses. *Genome Res.* 19, 521–532. doi:10.1101/gr.074906.107

Guo, Y., Yu, L., Wen, Z., and Li, M. (2008). Using support vector machine combined with auto covariance to predict protein-protein interactions from protein sequences. *Nucleic Acids Res.* 36, 3025–3030. doi:10.1093/nar/gkn159

Haarhuis, J. H. I., Van Der Weide, R. H., Blomen, V. A., Yáñez-Cuna, J. O., Amendola, M., Van Ruiten, M. S., et al. (2017). The cohesin release factor WAPL restricts chromatin loop extension. *Cell* 169, 693–707.e14. doi:10.1016/j.cell.2017.04.013

Han, Y. M., Yang, H., Huang, Q. L., Sun, Z. J., Li, M. L., Zhang, J. B., et al. (2022). Risk prediction of diabetes and pre-diabetes based on physical examination data. *Math. Biosci. Eng.* 19, 3597–3608. doi:10.3934/mbe.2022166

Hnisz, D., Day, D. S., and Young, R. A. (2016). Insulated neighborhoods: Structural and functional units of mammalian gene control. *Cell* 167, 1188–1200. doi:10.1016/j.cell.2016.10.024

Hsieh, T. H., Weiner, A., Lajoie, B., Dekker, J., Friedman, N., and Rando, O. J. (2015). Mapping nucleosome resolution chromosome folding in yeast by micro-C. *Cell* 162, 108–119. doi:10.1016/j.cell.2015.05.048

Huang, H., Zhu, Q., Jussila, A., Han, Y., Bintu, B., Kern, C., et al. (2021). CTCF mediates dosage- and sequence-context-dependent transcriptional insulation by forming local chromatin domains. *Nat. Genet.* 53, 1064–1074. doi:10.1038/s41588-021-00863-6

Kai, Y., Andricovich, J., Zeng, Z., Zhu, J., Tzatsos, A., and Peng, W. (2018). Predicting CTCF-mediated chromatin interactions by integrating genomic and epigenomic features. *Nat. Commun.* 9, 4221. doi:10.1038/s41467-018-06664-6

Katainen, R., Dave, K., Pitkänen, E., Palin, K., Kivioja, T., Välimäki, N., et al. (2015). CTCF/cohesin-binding sites are frequently mutated in cancer. *Nat. Genet.* 47, 818–821. doi:10.1038/ng.3335

Kellis, M., Wold, B., Snyder, M. P., Bernstein, B. E., Kundaje, A., Marinov, G. K., et al. (2014). Defining functional DNA elements in the human genome. *Proc. Natl. Acad. Sci. U. S. A.* 111, 6131–6138. doi:10.1073/pnas.1318948111

Kim, Y., Shi, Z., Zhang, H., Finkelstein, I. J., and Yu, H. (2019). Human cohesin compacts DNA by loop extrusion. *Science* 366, 1345–1349. doi:10.1126/science.aaz4475

Li, C. C., and Liu, B. (2020). MotifCNN-fold: Protein fold recognition based on fold-specific features extracted by motif-based convolutional neural networks. *Brief. Bioinform* 21, 2133–2141. doi:10.1093/bib/bbz133

Li, G., Chen, Y., Snyder, M. P., and Zhang, M. Q. (2017). ChIA-PET2: A versatile and flexible pipeline for ChIA-PET data analysis. *Nucleic Acids Res.* 45, e4. doi:10.1093/nar/gkw809

Lieberman-Aiden, E., Van Berkum, N. L., Williams, L., Imakaev, M., Ragoczy, T., Telling, A., et al. (2009). Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science* 326, 289–293. doi:10.1126/science.1181369

Liu, L., Li, Q. Z., Jin, W., Lv, H., and Lin, H. (2019). Revealing gene function and transcription relationship by reconstructing gene-level chromatin interaction. *Comput. Struct. Biotechnol. J.* 17, 195–205. doi:10.1016/j.csbj.2019.01.011

Liu, L., Zhang, L. R., Dao, F. Y., Yang, Y. C., and Lin, H. (2021). A computational framework for identifying the transcription factors involved in enhancer-promoter loop formation. *Mol. Ther. Nucleic Acids* 23, 347–354. doi:10.1016/j.omtn.2020.11.011

Lupiáñez, D. G., Kraft, K., Heinrich, V., Krawitz, P., Brancati, F., Klopocki, E., et al. (2015). Disruptions of topological chromatin domains cause pathogenic rewiring of gene-enhancer interactions. *Cell* 161, 1012–1025. doi:10.1016/j.cell.2015.04.004

Lv, H., Dao, F. Y., Zulfiqar, H., Su, W., Ding, H., Liu, L., et al. (2021). A sequence-based deep learning approach to predict CTCF-mediated chromatin loop. *Brief. Bioinform* 22, bbab031. doi:10.1093/bib/bbab031

Narendra, V., Rocha, P. P., An, D., Raviram, R., Skok, J. A., Mazzoni, E. O., et al. (2015). CTCF establishes discrete functional chromatin domains at the Hox clusters during differentiation. *Science* 347, 1017–1021. doi:10.1126/science.1262088

Ohlsson, R., Renkawitz, R., and Lobanenkov, V. (2001). CTCF is a uniquely versatile transcription regulator linked to epigenetics and disease. *Trends Genet.* 17, 520–527. doi:10.1016/s0168-9525(01)02366-6

Phillips, J. E., and Corces, V. G. (2009). Ctcf: Master weaver of the genome. *Cell* 137, 1194–1211. doi:10.1016/j.cell.2009.06.001

Rao, S. S., Huntley, M. H., Durand, N. C., Stamenova, E. K., Bochkov, I. D., Robinson, J. T., et al. (2014). A 3D map of the human genome at kilobase resolution reveals principles of chromatin looping. *Cell* 159, 1665–1680. doi:10.1016/j.cell.2014.11.021

Rao, S. S. P., Huang, S. C., Glenn St Hilaire, B., Engreitz, J. M., Perez, E. M., Kieffer-Kwon, K. R., et al. (2017). Cohesin loss eliminates all loop domains. *Cell* 171, 305–320.e24. doi:10.1016/j.cell.2017.09.026

Rowley, M. J., and Corces, V. G. (2018). Organizational principles of 3D genome architecture. *Nat. Rev. Genet.* 19, 789–800. doi:10.1038/s41576-018-0060-8

Schones, D. E., Smith, A. D., and Zhang, M. Q. (2007). Statistical significance of cis-regulatory modules. *BMC Bioinforma.* 8, 19. doi:10.1186/1471-2105-8-19

Soochit, W., Sleutels, F., Stik, G., Bartkuhn, M., Basu, S., Hernandez, S. C., et al. (2021). CTCF chromatin residence time controls three-dimensional genome organization, gene expression and DNA methylation in pluripotent cells. *Nat. Cell Biol.* 23, 881–893. doi:10.1038/s41556-021-00722-w

West, A. G., Gaszner, M., and Felsenfeld, G. (2002). Insulators: Many functions, many mechanisms. *Genes Dev.* 16, 271–288. doi:10.1101/gad.954702

Wutz, G., Várnai, C., Nagasaka, K., Cisneros, D. A., Stocsits, R. R., Tang, W., et al. (2017). Topologically associating domains and chromatin loops depend on cohesin and are regulated by CTCF, WAPL, and PDS5 proteins. *Embo J.* 36, 3573–3599. doi:10.15252/embj.201798004

Xi, W., and Beer, M. A. (2021). Loop competition and extrusion model predicts CTCF interaction specificity. *Nat. Commun.* 12, 1046. doi:10.1038/s41467-021-21368-0

Xu, L., Liang, G., Liao, C., Chen, G. D., and Chang, C. C. (2019). K-skip-n-gram-RF: A random forest based method for alzheimer's disease protein identification. *Front. Genet.* 10, 33. doi:10.3389/fgene.2019.00033

Yang, H., Luo, Y., Ren, X., Wu, M., He, X., Peng, B., et al. (2021). Risk prediction of diabetes: Big data mining with fusion of multifarious physical examination indicators. *Inf. Fusion* 75, 140–149. doi:10.1016/j.inffus.2021.02.015

Yang, Y., Gao, D., Xie, X., Qin, J., Li, J., Lin, H., et al. (2022). DeepIDC: A prediction framework of injectable drug combination based on heterogeneous information and deep learning. *Clin. Pharmacokinet.* 61, 1749–1759. doi:10.1007/s40262-022-01180-9

Zhang, Q., Li, H., Liu, Y., Li, J., Wu, C., and Tang, H. (2022a). Exosomal non-coding RNAs: New insights into the biology of hepatocellular carcinoma. *Curr. Oncol.* 29, 5383–5406. doi:10.3390/curroncol29080427

Zhang, X., and Liu, S. (2017). RBPPred: Predicting RNA-binding proteins from sequence using SVM. *Bioinformatics* 33, 854–862. doi:10.1093/bioinformatics/btw730

Zhang, Z. Y., Ning, L., Ye, X., Yang, Y. H., Futamura, Y., Sakurai, T., et al. (2022b). iLoc-miRNA: extracellular/intracellular miRNA prediction using deep BiLSTM with attention mechanism. *Brief. Bioinform* 23, bbac395. doi:10.1093/bib/bbac395

frontiers | Frontiers in Genetics

TYPE Technology and Code
PUBLISHED 30 May 2023
DOI 10.3389/fgene.2023.1082782

OPEN ACCESS

EDITED BY
Joanna Szyda,
Wroclaw University of Environmental and Life Sciences, Poland

REVIEWED BY
Arvind Sonwane,
Indian Council of Agricultural Research (ICAR), India
Nathan Olson,
National Institute of Standards and Technology (NIST), United States

*CORRESPONDENCE
Dörte Wittenburg,
wittenburg@fbn-dummerstorf.de

RECEIVED 28 October 2022
ACCEPTED 12 May 2023
PUBLISHED 30 May 2023

CITATION
Melzer N, Qanbari S, Ding X and Wittenburg D (2023), CLARITY: a Shiny app for interactive visualisation of the bovine physical-genetic map.
Front. Genet. 14:1082782.
doi: 10.3389/fgene.2023.1082782

CLARITY: a Shiny app for interactive visualisation of the bovine physical-genetic map

Nina Melzer, Saber Qanbari, Xi Ding and Dörte Wittenburg*

Research Institute for Farm Animal Biology (FBN), Dummerstorf, Germany

The arrangement of markers on the genome can be defined in either physical or linkage terms. While a physical map represents the inter-marker distances in base pairs, a genetic (or linkage) map pictures the recombination rate between pairs of markers. High-resolution genetic maps are key elements for genomic research, such as fine-mapping of quantitative trait loci, but they are also needed for creating and updating chromosome-level assemblies of whole-genome sequences. Based on published results on a large pedigree of German Holstein cattle and newly obtained results with German/Austrian Fleckvieh cattle, we aim at providing a platform that allows users to interactively explore the bovine genetic and physical map. We developed the R Shiny app CLARITY available online at https://nmelzer.shinyapps.io/clarity and as R package at https://github.com/nmelzer/CLARITY that provides access to the genetic maps built on the Illumina Bovine SNP50 genotyping array with markers ordered according to the physical coordinates of the most recent bovine genome assembly ARS-UCD1.2. The user is able to interconnect the physical and genetic map for a whole chromosome or a specific chromosomal region and can inspect a landscape of recombination hotspots. Moreover, the user can investigate which of the frequently used genetic-map functions locally fits best. We further provide auxiliary information about markers being putatively misplaced in the ARS-UCD1.2 release. The corresponding output tables and figures can be downloaded in various formats. By ongoing data integration from different breeds, the app also facilitates comparison of different genome features, providing a valuable tool for education and research purposes.

KEYWORDS

single nucleotide polymorphism, linkage, recombination rate, education, mapping function

1 Introduction

Genomic research involving gene mapping of economically important traits, population-specific genetic structure and evolutionary history relies heavily on genetic maps built on the extent of linkage disequilibrium (LD) between genomic markers (e.g., Georges et al., 2019; Johnsson and Jungnickel, 2021). For example, to what extent LD persists in a certain genomic region determines the number of markers required to fine-map a quantitative trait loci with succinct power and precision (for review see Qanbari, 2020). Moreover, genetic maps are valuable resources for comparative genomic analyses among breeds or species (e.g., Everts-van der Wind et al., 2005; Womack, 2005). Of utmost topical importance, however, is the contribution of genetic maps (also known as linkage maps) to measuring haplotype similarity in the

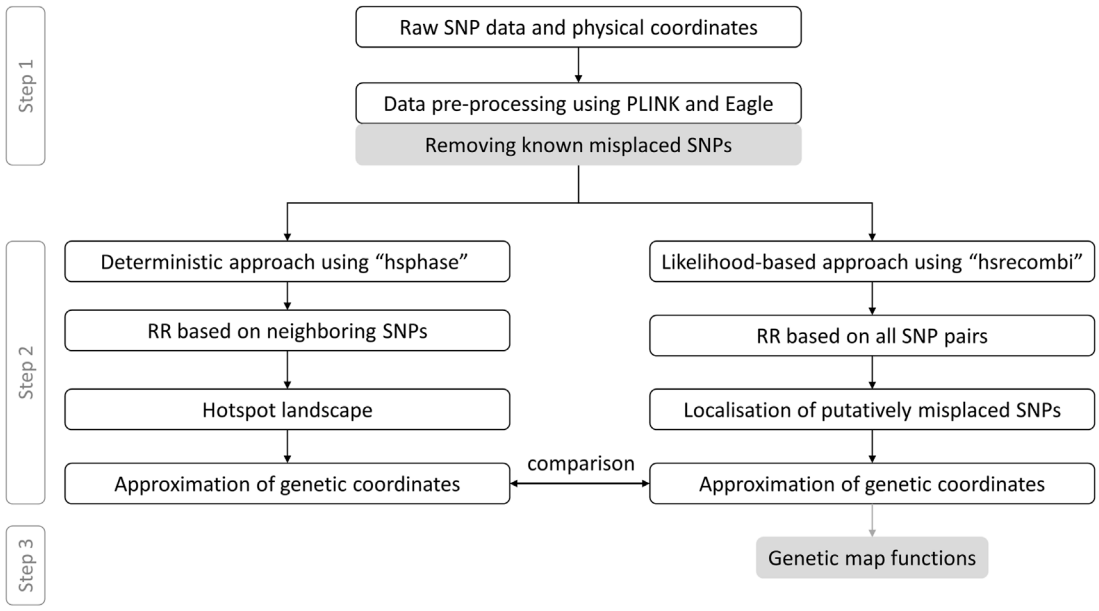


FIGURE 1
Data processing from raw genotypes to genetic-map functions. This workflow with grey coloured boxes excluded follows Qanbari and Wittenburg (2020) RR, recombination rate.

context of genomic selection (Musa, 2021) and to chromosome-level assemblies of whole-genome sequences (e.g., De los Ríos-Pérez et al., 2020; Rosen et al., 2020).

Given the value of cattle in sustaining the world food security, the bovine genome is subject of vast amount of ongoing research. We recently updated the genetic map of German Holstein breed (Qanbari and Wittenburg, 2020) and compared it with physical coordinates of the most recent bovine reference genome assembly ARS-UCD1.2 (Rosen et al., 2020). As an extension to this resource, here we introduce a Shiny app CLARITY which facilitates interactive visualisation of the bovine genetic and physical map. CLARITY illustrates the details of male recombination across the bovine genome of selected breeds and suggests suitable genetic-map functions. In addition to published findings, results have been updated by taking most recent knowledge about putatively misplaced markers in the bovine genome assembly into account (Qanbari et al., 2022). Moreover, a linkage map for German/Austrian Fleckvieh cattle has been created. The CLARITY app can therefore serve as a toolkit for both educational and research purposes for the genome of bovine and related species.

2 Data preparation

The app provides access to linkage maps built based on the 50K genotypes of a large pedigree of German Holstein cattle (1,053 half-sib families which comprise 367,056 genotyped animals) as well as German/Austrian Fleckvieh cattle (298,850 genotyped animals pedigreed across 6,866 half-sib families) and based on estimates of recombination rate between intra-chromosomal marker pairs. In what follows, we briefly describe the workflow towards genetic coordinates as depicted in Figure 1 according to Qanbari and Wittenburg (2020). As some of the steps require several days of computing or visual inspection, the workflow was executed once in advance; the app itself dynamically processes physical and genetic coordinates as well as pairwise recombination rates. If not stated otherwise, data processing was executed in R v4.1.3 (R Core Team, 2022).

Step 1: Genotype data were filtered for minor allele frequency >1% and for Mendelian inconsistencies both on marker and individual level using PLINK v1.9 (Purcell et al., 2007) with recommended settings. Genotypes with a Mendelian inheritance error were set to "NA" and missing values were imputed using Eagle v2.4.1 program (Loh et al., 2016). Putatively misplaced markers, which have recently been reported (Qanbari et al., 2022), were discarded. Data passed on to Step 2 comprised 876 half-sib families with 366,565 progeny genotyped at 44,696 single nucleotide polymorphisms (SNPs) in Holstein and 1,577 half-sib families with 270,636 progeny genotyped at 40,003 SNPs in Fleckvieh.

Step 2: Paternal recombination rates to build the genetic map were derived from two different methods, see Figure 1. First, the *deterministic approach* developed by Ferdosi et al. (2014), which is implemented in the R package hsphase v2.0.2, yielded estimates of recombination rate between adjacent markers. These estimates were later used to form the landscape of recombination hotspots. Genetic coordinates were estimated as cumulative sum of recombination rates between neighbouring markers. Second, the *likelihood-based approach*, as implemented in the R package hsrecombi v0.3.4 (Wittenburg, 2020), was applied to estimate recombination rates between all intra-chromosomal marker pairs. These estimates of recombination rate also enabled the identification of candidate misplaced markers in the current assembly release ARS-UCD1.2 (Qanbari and Wittenburg, 2020). Genetic coordinates were obtained with a smoothing approach, in which

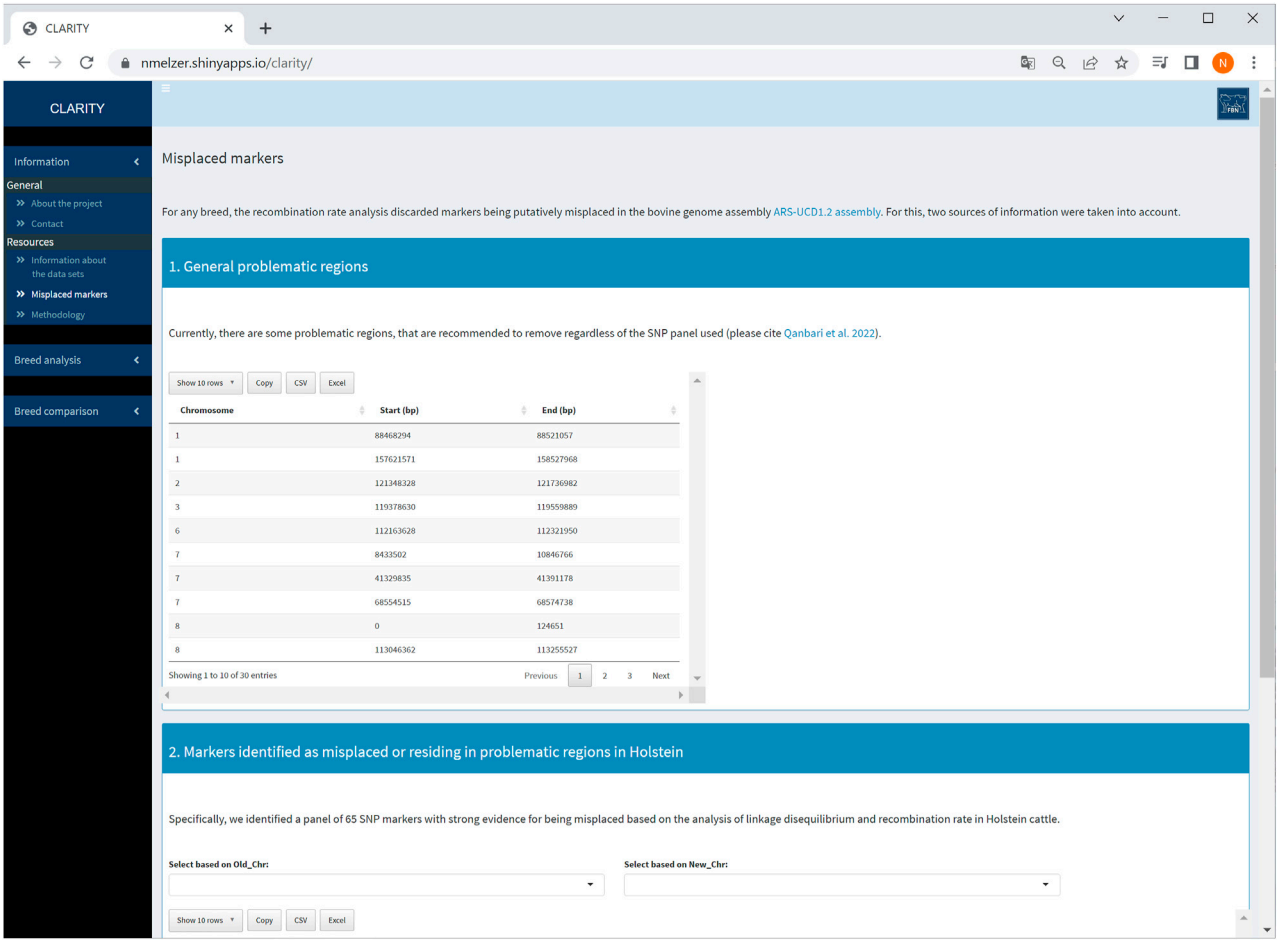


FIGURE 2
Screenshot of information about problematic regions and list of misplaced markers detected in Holstein cattle.

all recombination rates < 0.05 between any intra-chromosomal marker pairs were taken into account.

Step 3: As a novel contribution, the relationship between recombination rate and genetic distance between all intra-chromosomal marker pairs was investigated for a set of commonly used genetic-map functions (Haldane, 1919; Rao et al., 1977; Felsenstein, 1979; Liberman and Karlin, 1984). Given the estimates of recombination rate $\hat{\theta}_{i,j}$ between two markers i and j and its genetic distances $d_{i,j}$ derived in Step 2 (Qanbari and Wittenburg, 2020), a genetic-map function $f(d_{i,j}|a) = \theta_{i,j}$ was fitted to the data by solving the following minimisation problem in terms of the model parameter a (where p is the total number of markers per chromosome):

$$\sum_{\substack{i,j=1 \\ i<j}}^{p} \left(\hat{\theta}_{i,j} - f\left(d_{i,j}|a\right)\right)^2 \rightarrow \min \text{ s.t. } a \in \mathbb{R}$$

We solved this optimisation problem using the R function optim with "Brent" option allowing to specify restrictions on a. Rao's system of mapping function requires $a \in (0, 1)$. Furthermore, instead of Haldane's original map function, we investigated a scaled version thereof, i.e.

$$f\left(d_{i,j}|a\right) = \frac{1}{2}\left(1 - e^{-2ad_{i,j}}\right) \text{ with } a > 0$$

For the Binomial map function of Liberman and Karlin (1984), we employed a grid search over $a \in \{2, 3, 4, 5\}$ seeking the minimum squared deviation as described above. The fitted function leading to the least squared deviation constituted the "best" genetic-map function.

3 Implementation

The CLARITY app is an R Shiny web GUI for various operating systems. It relies on several R packages to enable the outcome and visualisation functionalities. CLARITY was implemented in R v4.1.3 (R Core Team, 2022) with help of the R packages shiny v1.7.1 (Chang et al., 2021) and shinydashboard v0.7.2 (Chang and Borges Ribeiro, 2021) to create a dashboard. The graphical output was produced using the R packages ggplot2 v3.3.5 (Wickham, 2016), plotly v4.10.0 (Sievert, 2020) and ggVennDiagram v1.2.0 (Gao,

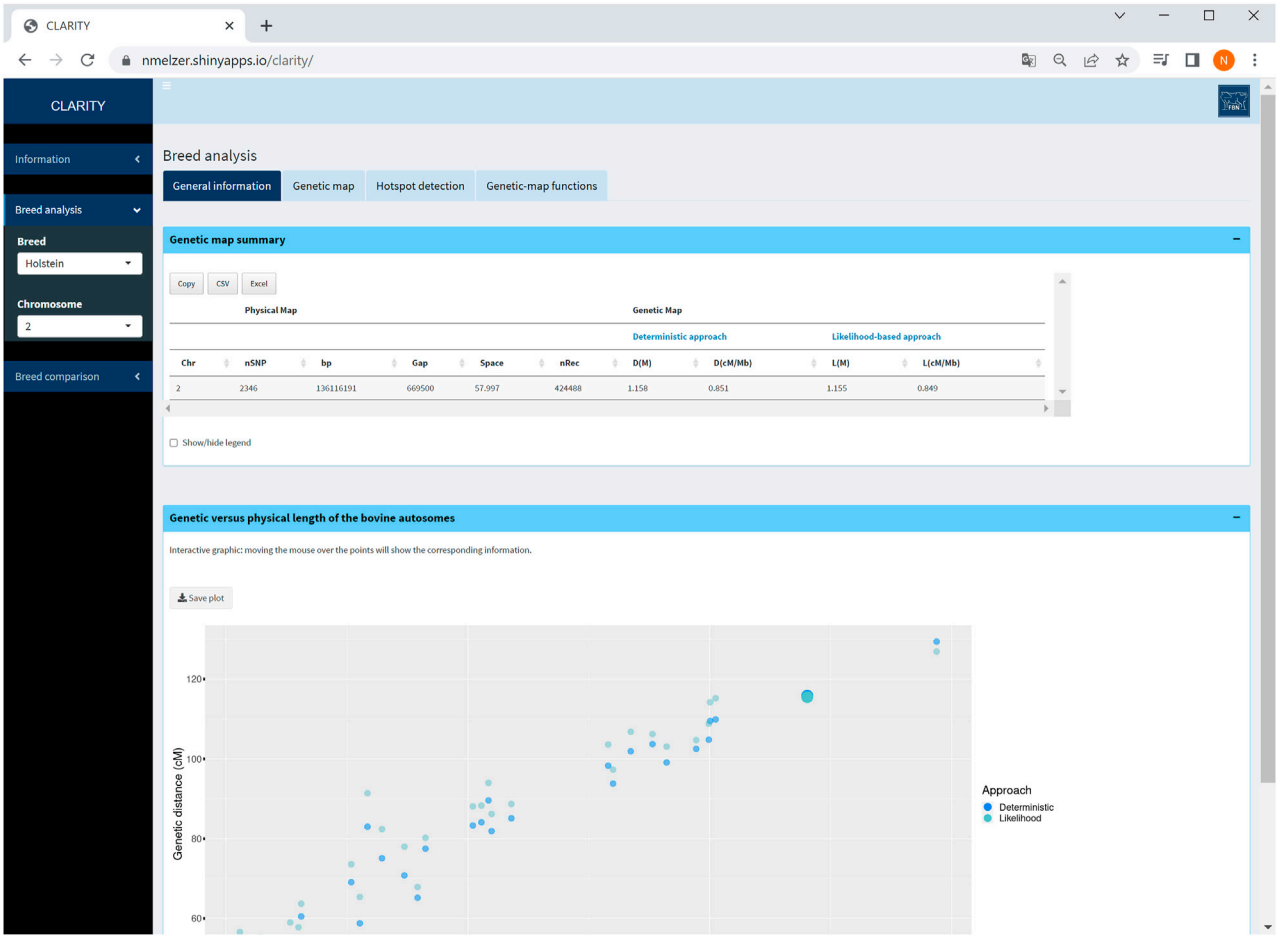


FIGURE 3
Screenshot of general information on selected chromosome 2. The graphic displays the relationship between genetic and physical map length of all autosomes in Holstein cattle and highlights the selected chromosome.

2021). To provide additional helpful features for the app, such as hiding or toogle of elements, the R packages shinyjs v2.1.0 (Attali, 2021) and shinycssloaders v1.0.0 (Sali and Attali, 2020) were incorporated. Tables were generated using the R package DT v0.22 (Xie et al., 2022). Further R packages were employed: cachem v1.0.6 (Chang, 2021), config v0.3.1 (Allaire, 2020), dplyr v1.0.10 (Wickham et al., 2022), gridExtra v2.3 (Auguie, 2017), htmltools v0.5.2 (Cheng et al., 2021), magrittr v2.0.3 (Bache and Wickham, 2022), metathis v1.1.2 (Aden-Buie, 2022), rlang v1.0.6 (Henry and Wickham, 2022a), sf v1.0.8 (Pebesma, 2018), RVenn v1.1.0 (Akyol, 2019) and purrr v0.3.5 (Henry and Wickham, 2022b). Eventually, an R package was built from the CLARITY app with use of the R package golem v0.3.2 (Fay et al., 2022), which offers default R files for creating the package as well as for deploying the app. The R package roxygen2 v7.1.2 (Wickham et al., 2021) was employed for package documentation. The processed data (i.e., recombination rates, genetic coordinates and parameters of genetic-map functions) were included as Rdata files in the folder "extdata".

We optimized the app following recommendations for best practice with lighthouse (Google LLC, 2022). Figures were compressed with the tool Squoosh (Google Chrome Team, 2022), and caching of those figures requiring longer loading was enabled.

The structure of the app relies on modules, which eases a clear and concise organisation. The use of modules and corresponding interfaces to the main shiny ui and shiny server enables a straightforward maintenance of the software. Furthermore, since each module is an independent app with its own interface and server (Di Filippo et al., 2019), future modification and extension of the app are supported.

4 Realisation and features

The app has three sidebar menus: "Information", "Breed analysis" and "Breed comparison" which are described in the following.

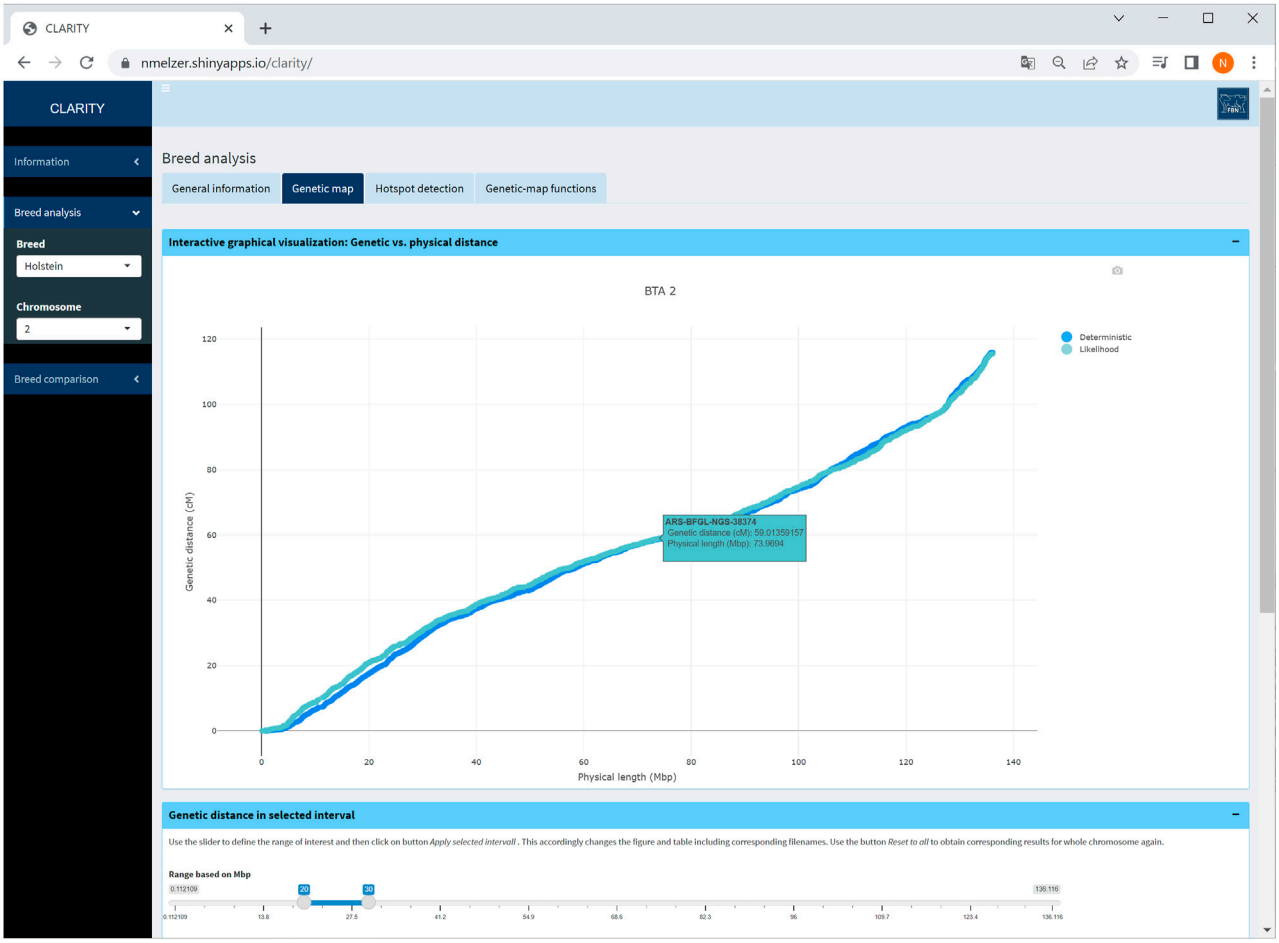


FIGURE 4
Screenshot of genetic map obtained with the likelihood-based and the deterministic approach on selected chromosome 2 in Holstein cattle. A table below (not shown) contains detailed information on each marker.

4.1 Information

The first sidebar menu comprises two subitems: (1) general information about the project and contact options as well as (2) details of resources used. More specifically, subitem (2) contains a brief data description for each breed and outlines the methodology used for data analysis and parameter estimation. This subitem also provides auxiliary information about candidate markers identified as being putatively misplaced and/or residing in problematic regions of the ARS-UCD1.2 release (Qanbari et al., 2022, see also Figure 2). These markers were recommended to be excluded from subsequent genomic analyses, such as phasing, imputation or genome-wide association studies.

4.2 Breed analysis

Under the second sidebar menu "Breed analysis", all tabulated and graphical outcomes are presented for the available breeds. So far, options "Holstein" and "Fleckvieh" are available. The user can select a single chromosome or all chromosomes to interconnect the physical and linkage map. The results are divided into different tabs (implemented as separate modules) within the main panel: general information, genetic map, hotspot detection and genetic-map functions. Generally, outputs including tables and figures for a certain interval or for the entire data can also be downloaded for being locally stored. The properties of each tab are explained in more detail below.

4.2.1 General information

For all chromosomes, summary statistics are provided about the number of markers considered, total number of recombination events, length of physical and both genetic maps (from Step 2) in tabulated format. This table reduces if a single chromosome is selected (Figure 3). An interactive graphical output displays physical *versus* genetic length per chromosome.

FIGURE 5
Screenshot of recombination rate between adjacent markers on selected chromosome 2 in Holstein cattle. The coloured points are indicative for putative recombination hotspot intervals using the default threshold. The table below (not shown) provides detailed information on each marker in a hotspot interval.

4.2.2 Genetic map

Physical and genetic map coordinates are listed for all markers in table or graphical format. If a specific chromosome is selected on the sidebar menu, the user can zoom into relevant chromosomal regions (Figure 4). In each graphic, the genetic-map coordinates of the deterministic and likelihood-based approach appear. The graphical as well as the table output is adaptable to a user specified chromosome window. Selecting the option "all chromosomes" provides a static overview of 29 single graphics.

4.2.3 Hotspot detection

The CLARITY app offers a landscape of putative recombination hotspots, in which marker intervals with an elevated recombination rate are colour-marked across the bovine genome or a selected chromosome (Figure 5). The default threshold for the recombination rate is adopted from Ma et al. (2015) who defined a hotspot region with a recombination rate exceeding 2.5 standard deviations from the genome-wide average. The threshold is adjustable by the user. Changing the threshold accordingly affects the interactive graphic as well as the corresponding table listing all markers within the hotspot intervals.

4.2.4 Genetic-map functions

The user can investigate the suitability of frequently used genetic-map functions and their overall and local fit to the observed recombination activity (Figure 6). The parameter specifying a genetic-map function was estimated in Step 3 which took all intra-chromosomal marker pairs into account.

The fitted genetic-map functions are illustrated together with a reduced scatterplot of recombination rate *versus* genetic distance for computational reasons. Especially for chromosomes with

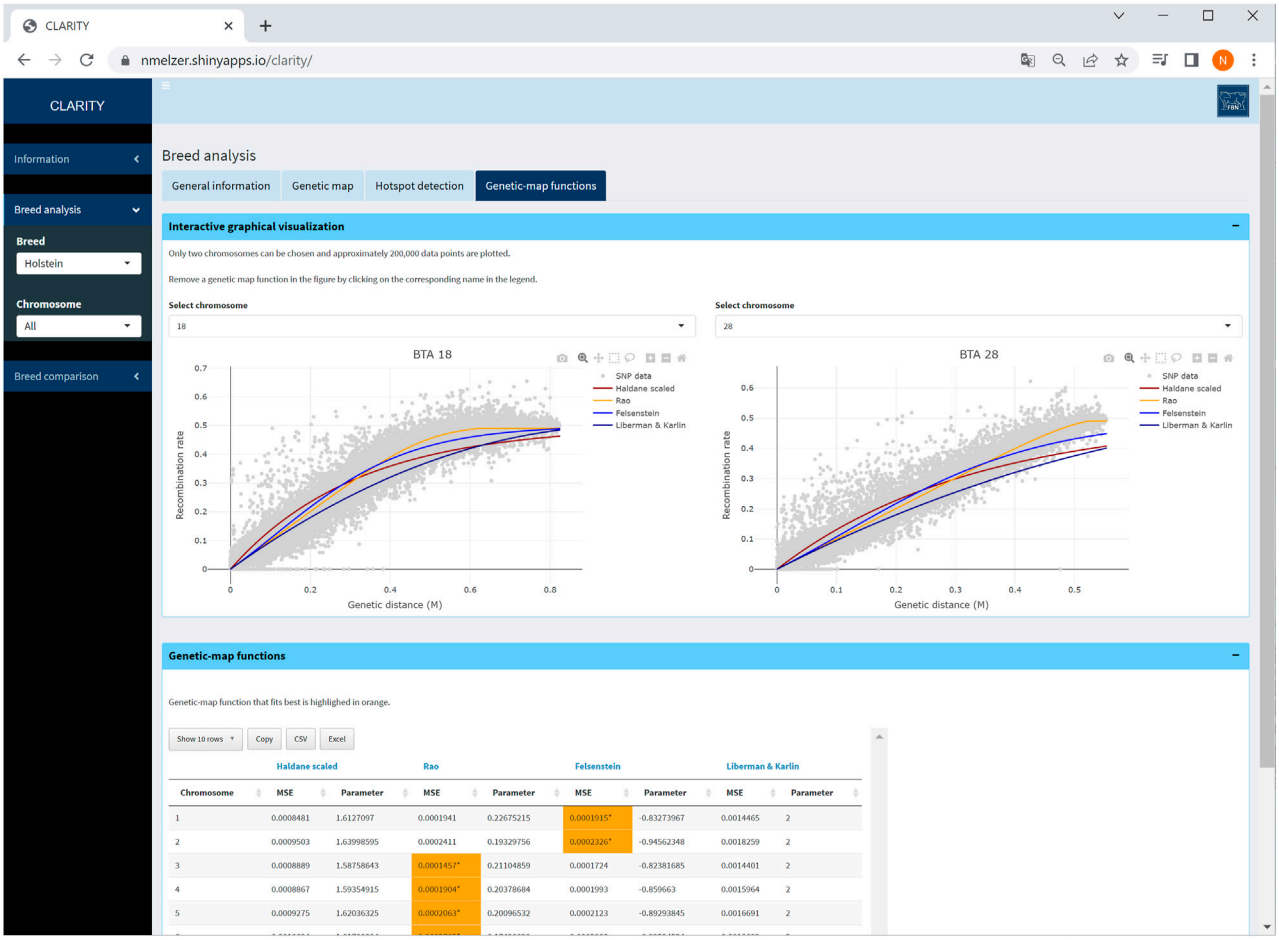


FIGURE 6
Screenshot of genetic-map functions and its overall fit on two selected chromosomes in Holstein cattle. The table provides details on estimated parameters of genetic-map functions and mean squared error of fitted curves.

$p \geq 1{,}415$ (i.e., >1,000,000 marker pairs), the computing time is drastically increased retarding the visualisation. To ensure smooth processing, data were thinned based on the Euclidian distance among consecutive data points; a data point is a pair of recombination rate and genetic coordinate ordered according to a vectorised triangular matrix of SNP identifiers. In total, 200,000 data points with largest Euclidian distance were kept. This reduction of data did not impair the visual appearance of the scatter plot.

4.3 Breed comparison

The third sidebar menu "Breed comparison" contains comparative analyses between breeds separated into the same tabs as described above. In addition, a Venn diagram summarises numbers of breed-specific and shared SNPs on a selected chromosome or over the entire autosome as well as in hotspot intervals. In the tabs "genetic map" and "hotspot detection", the Venn diagram is interactively linked with the corresponding table—this allows the user to retrieve particular information of a selected Venn set. The Venn diagram also dynamically adapts to a user defined range and threshold, respectively. Furthermore, since the fit of genetic-map functions might differ between breeds and chromosomes, a barplot displays counts of "best" genetic-map functions in each breed if the option "all chromosomes" is chosen.

Particularly, a comparison of Holstein and Fleckvieh cattle suggested similar recombination activity genome-wide. As an example, an inspection of hotspot intervals on chromosome 3 (Figure 7) underlined regions of increased recombination rates at the chromosome ends that coincided well in both breeds. Furthermore, for each chromosome, genetic-map functions were almost overlapping. Small deviations of genetic-map functions were observed on chromosomes 5, 6, 9, 15, 16, and 18 with a slightly steeper curve in Fleckvieh.

5 Discussion and outlook

The CLARITY app provides an environment to interactively explore the physical and genetic map in selected cattle breeds. Importantly, processing of genotype data and presenting results via

FIGURE 7
Screenshot of recombination rate between adjacent markers on selected chromosome 3 in a comparison of Holstein and Fleckvieh cattle. The yellow triangles and orange rectangles highlight markers in hotspot intervals of Holstein and Fleckvieh, respectively.

Shiny app rely on the current bovine genome assembly ARS-UCD1.2. In case of new assembly releases, Steps 1–3 need to be re-run and an app update becomes necessary. (Especially, both approaches in Step 2 depend on the ordering of markers for inferring phases of sire genotypes and recombination events.) Though a pipeline for Steps 1–3 is available at github, and it could theoretically be part of the Shiny app, we do not recommend its inclusion for computational matters as mentioned in Section 2.

Further work on the integration of data from other breeds (beef, dairy, dual-purpose) is underway and will facilitate complex comparative analyses of map features (e.g., hotspot intervals and assembly flaws) in different genomes. Our Shiny app will be extended accordingly, certainly increasing its value for educational and research purposes.

Data availability statement

CLARITY is a publicly available Shiny app that can be accessed via web interface at https://nmelzer.shinyapps.io/clarity. The corresponding R package CLARITY v1.0.1 including the source code and processed data can be downloaded from https://github.com/nmelzer/CLARITY under the terms of GPL (≥ 2.0). A pipeline for processing genotype data and an R script for composing the app input data are available at https://github.com/wittenburg/hsrecombi. Restrictions apply to the availability of the original data supporting the findings of this study due to thirdparty ownership. Genotype data are available from the Association for Bioeconomy Research (FBF, Bonn) and ZuchtData (Vienna) upon agreement. Requests to access the original datasets should be directed to www.fbf-forschung.de/kontakt.html; www.rinderzucht.at/zuchtdata/team.html.

Author contributions

NM developed the Shiny app package. SQ, XD, and DW analysed the data. NM, SQ, and DW wrote the manuscript and contributed to the online documentation of the R Shiny app. DW

conceived the project. All authors contributed to the article and approved the final version.

Funding

This study was supported by the grant from the German Federal Ministry of Education and Research (BMBF, FKZ 031L0166 CompLS). The publication of this article was funded by the Open Access Fund of the FBN.

Acknowledgments

We gratefully acknowledge the support of the Association for Bioeconomy Research (FBF, Bonn, Germany) as representative of German Holstein cattle breeders for participating in this project and the German Evaluation Center (VIT, Verden, Germany) for composing the Holstein data. We thank ZuchtData (Vienna, Austria) for providing the German/Austrian Fleckvieh data.

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

Aden-Buie, G. (2022). Metathis: HTML metadata tags for 'R markdown' and 'shiny'. Available at: https://CRAN.R-project.org/package=metathis (Accessed October 20, 2022).

Akyol, T. Y. (2019). RVenn: Set operations for many sets. Available at: https://CRAN.R-project.org/package=RVenn (Accessed October 20, 2022).

Allaire, J. J. (2020). config: Manage environment specific configuration values. Available at: https://CRAN.R-project.org/package=config (Accessed October 20, 2022).

Attali, D. (2021). shinyjs: Easily improve the user experience of your shiny apps in seconds. Available at: https://CRAN.R-project.org/package=shinyjs (Accessed October 20, 2022).

Auguie, B. (2017). gridExtra: Miscellaneous functions for "grid" graphics. Available at: https://CRAN.R-project.org/package=gridExtra (Accessed October 20, 2022).

Bache, S. M., and Wickham, H. (2022). Magrittr: A forward-pipe operator for R. Available at: https://CRAN.R-project.org/package=magrittr (Accessed October 20, 2022).

Chang, W., and Borges Ribeiro, B. (2021). shinydashboard: Create dashboards with 'shiny'. Available at: https://CRAN.R-project.org/package=shinydashboard (Accessed October 20, 2022).

Chang, W. (2021). cachem: Cache R objects with automatic pruning. Available at: https://CRAN.R-project.org/package=cachem (Accessed October 20, 2022).

Chang, W., Cheng, J., Allaire, J. J., Sievert, C., Schloerke, B., Xie, Y., et al. (2021). shiny: Web application framework for R. Available at: https://CRAN.R-project.org/package=shiny (Accessed October 20, 2022).

Cheng, J., Sievert, C., Schloerke, B., Chang, W., Xie, Y., and Allen, J. (2021). htmltools: Tools for HTML. Available at: https://CRAN.R-project.org/package=htmltools (Accessed October 20, 2022).

De los Ríos-Pérez, L., Verleih, M., Rebl, A., Brunner, R., Nguinkal, J. A., Klosa, J., et al. (2020). An ultra-high density SNP-based linkage map for enhancing the pikeperch (*Sander lucioperca*) genome assembly to chromosome-scale. *Sci. Rep.* 10, 22335. doi:10.1038/s41598-020-79358-z

Di Filippo, L., Righelli, D., Gagliardi, M., Matarazzo, M. R., and Angelini, C. (2019). HiCeekR: A novel shiny app for hi-C data analysis. *Front. Genet.* 10, 1079. doi:10.3389/fgene.2019.01079

Everts-van der Wind, A., Larkin, D. M., Green, C. A., Elliott, J. S., Olmstead, C. A., Chiu, R., et al. (2005). A high-resolution whole-genome cattle–human comparative map reveals details of mammalian chromosome evolution. *PNAS* 102, 18526–18531. doi:10.1073/pnas.0509285102

Fay, C., Guyader, V., Rochette, S., and Girard, C. (2022). Golem: A framework for robust shiny applications. Available at: https://CRAN.R-project.org/package=golem (Accessed October 20, 2022).

Felsenstein, J. (1979). A mathematically tractable family of genetic mapping functions with different amounts of interference. *Genetics* 91, 769–775. doi:10.1093/genetics/91.4.769

Ferdosi, M. H., Kinghorn, B. P., van der Werf, J. H., Lee, S. H., and Gondro, C. (2014). hsphase: an R package for pedigree reconstruction, detection of recombination events, phasing and imputation of half-sib family groups. *BMC Bioinf* 15, 172. doi:10.1186/1471-2105-15-172

Gao, C.-H. (2021). ggVenndiagram: A 'ggplot2' implement of Venn diagram. Available at: https://CRAN.R-project.org/package=ggVennDiagram (Accessed October 20, 2022).

Georges, M., Charlier, C., and Hayes, B. (2019). Harnessing genomic information for livestock improvement. *Nat. Rev. Genet.* 20, 135–156. doi:10.1038/s41576-018-0082-2

Google Chrome Team (2022). Squoosh!. Available at: https://github.com/GoogleChromeLabs/squoosh (Accessed June 21, 2022).

Google LLC (2022). Lighthouse. Available at: https://github.com/GoogleChrome/lighthouse (Accessed June 21, 2022).

Haldane, J. B. S. (1919). The combination of linkage values and the calculation of distances between the loci of linked factors. *J. Genet.* 8, 299–309.

Henry, L., and Wickham, H. (2022b). purrr: Functional programming tools. Available at: https://CRAN.R-project.org/package=purrr (Accessed October 20, 2022).

Henry, L., and Wickham, H. (2022a). rlang: Functions for base types and Core R and 'tidyverse' features. Available at: https://CRAN.R-project.org/package=rlang (Accessed October 20, 2022).

Johnsson, M., and Jungnickel, M. K. (2021). Evidence for and localization of proposed causative variants in cattle and pig genomes. *Genet. Sel. Evol.* 53, 67. doi:10.1186/s12711-021-00662-x

Liberman, U., and Karlin, S. (1984). Theoretical models of genetic map functions. *Theor. Popul. Biol.* 25, 331–346. doi:10.1016/0040-5809(84)90013-3

Loh, P.-R., Danecek, P., Palamara, P. F., Fuchsberger, C., Reshef, Y. A., Finucane, H. K., et al. (2016). Reference-based phasing using the haplotype reference consortium panel. *Nat. Genet.* 48, 1443–1448.

Ma, L., O'Connell, J. R., VanRaden, P. M., Shen, B., Padhi, A., Sun, C., et al. (2015). Cattle sex-specific recombination and genetic control from a large pedigree analysis. *PLoS Genet.* 11, e1005387. doi:10.1371/journal.pgen.1005387

Musa, A. A. (2021). "A similarity matrix and its application in genomic selection for hedging haplotype diversity (Dissertation)" (Germany: University Kiel).

Pebesma, E. (2018). Simple features for R: Standardized support for spatial vector data. *R J.* 10 (1), 439–446. doi:10.32614/RJ-2018-009

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.

Qanbari, S. (2020). On the extent of linkage disequilibrium in the genome of farm animals. *Front. Genet.* 10, 1304. doi:10.3389/fgene.2019.01304

Qanbari, S., Schnabel, R. D., and Wittenburg, D. (2022). Evidence of rare misassemblies in the bovine reference genome revealed by population genetic metrics. *Anim. Genet.* 53, 498–505. doi:10.1111/age.13205

Qanbari, S., and Wittenburg, D. (2020). Male recombination map of the autosomal genome in German Holstein. *Genet. Sel. Evol.* 52, 73. doi:10.1186/s12711-020-00593-z

R Core Team (2022). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. Available at: https://www.R-project.org.

Rao, D. C., Morton, N. E., Lindsten, J., Hultén, M., and Yee, S. (1977). A mapping function for man. *Hum. Hered.* 27, 99–104. doi:10.1159/000152856

Rosen, B. D., Bickhart, D. M., Schnabel, R. D., Koren, S., Elsik, C. G., Tseng, E., et al. (2020). De novo assembly of the cattle reference genome with single-molecule sequencing. *GigaScience* 9, giaa021. doi:10.1093/gigascience/giaa021

Sali, A., and Attali, D. (2020). shinycssloaders: Add loading animations to a 'shiny' output while it's recalculating. Available at: https://CRAN.R-project.org/package=shinycssloaders (Accessed October 20, 2022).

Sievert, C. (2020). *Interactive web-based data visualisation with R, plotly, and shiny*. Boca Raton: Chapman and Hall/CRC. Available at: https://plotly-r.com.

Wickham, H., Danenberg, P., Csárdi, G., and Eugster, M. (2021). roxygen2: In-Line Documentation for R. Available at: https://CRAN.R-project.org/package=roxygen2 (Accessed October 20, 2022).

Wickham, H., François, R., Henry, L., and Müller, K. (2022). dplyr: A grammar of data manipulation. Available at: https://CRAN.R-project.org/package=dplyr (Accessed October 20, 2022)

Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis*. New York, NY: Springer.

Wittenburg, D. (2020). hsrecombi: Estimation of recombination rate and maternal LD in half-sibs. Available at: https://cran.r-project.org/package=hsrecombi (Accessed October 20, 2022).

Womack, J. E. (2005). Advances in livestock genomics: Opening the barn door. *Genome Res.* 15, 1699–1705. doi:10.1101/gr.3809105

Xie, Y., Cheng, J., and Tan, X. (2022). DT: A wrapper of the JavaScript library 'DataTables'. Available at: https://CRAN.R-project.org/package=DT (Accessed October 20, 2022).

frontiers | Frontiers in Genetics

TYPE Original Research
PUBLISHED 07 June 2023
DOI 10.3389/fgene.2023.1211020

OPEN ACCESS

EDITED BY
Lei Chen,
Shanghai Maritime University, China

REVIEWED BY
Yongqiang Xing,
Inner Mongolia University of Science and Technology, China
Cangzhi Jia,
Dalian Maritime University, China

*CORRESPONDENCE
Chengbing Huang,
abtchcb@qq.com
Zhaoyue Zhang,
zyzhang@uestc.edu.cn

RECEIVED 24 April 2023
ACCEPTED 24 May 2023
PUBLISHED 07 June 2023

CITATION
Su W, Qian X, Yang K, Ding H, Huang C and Zhang Z (2023), Recognition of outer membrane proteins using multiple feature fusion.
Front. Genet. 14:1211020.
doi: 10.3389/fgene.2023.1211020

Recognition of outer membrane proteins using multiple feature fusion

Wenxia Su[1], Xiaojun Qian[2], Keli Yang[3], Hui Ding[2], Chengbing Huang[4]* and Zhaoyue Zhang[2,5]*

[1]College of Science, Inner Mongolia Agriculture University, Hohhot, China, [2]School of Life Science and Technology, Center for Information Biology, University of Electronic Science and Technology of China, Chengdu, China, [3]Nonlinear Research Institute, Baoji University of Arts and Sciences, Baoji, China, [4]School of Computer Science and Technology, Aba Teachers University, Aba, China, [5]School of Healthcare Technology, Chengdu Neusoft University, Chengdu, China

Introduction: Outer membrane proteins are crucial in maintaining the structural stability and permeability of the outer membrane. Outer membrane proteins exhibit several functions such as antigenicity and strong immunogenicity, which have potential applications in clinical diagnosis and disease prevention. However, wet experiments for studying OMPs are time and capital-intensive, thereby necessitating the use of computational methods for their identification.

Methods: In this study, we developed a computational model to predict outer membrane proteins. The non-redundant dataset consists of a positive set of 208 outer membrane proteins and a negative set of 876 non-outer membrane proteins. In this study, we employed the pseudo amino acid composition method to extract feature vectors and subsequently utilized the support vector machine for prediction.

Results and Discussion: In the Jackknife cross-validation, the overall accuracy and the area under receiver operating characteristic curve were observed to be 93.19% and 0.966, respectively. These results demonstrate that our model can produce accurate predictions, and could serve as a valuable guide for experimental research on outer membrane proteins.

KEYWORDS

outer membrane protein, pseudo amino acid composition, support vector machine, jackknife test, prediction model

1 Introduction

Outer membrane proteins (OMPs) are a special type of proteins that are found in the outermost membranes of Gram-negative bacteria, mitochondria, and chloroplasts (Rollauer et al., 2015; Qi et al., 2022). OMPs serve a wide range of functions, including acting as adhesion factors in virulence, channels for small hydrophilic molecules, enzymes in biochemical reactions, and antigens in immune responses. They also work in concert with other substances to enhance the bacteria pathogenicity. Recent research on OMPs has revealed their potential for clinical diagnosis and disease prevention. Several published studies have explored OMPs as potential vaccine candidates (Budiardjo et al., 2021; Fahie et al., 2021; Cheng et al., 2022; Yu et al., 2022). The functions are determined by the OMP's structure and the way it interacts with other molecules. OMPs are typically composed of a transmembrane β-barrel architecture, providing permeability to the outer membrane and

maintaining structural stability. Among different types of OMPs, β-buckets consist of varying even numbers of β-folding sheets, ranging from 8 to 26 (Rollauer et al., 2015). The specific composition of the β-barrel architecture is determined by the amino acid sequence of the OMPs. Mutations in sequences can impact the stability and function of the protein.

Distinguishing OMPs from non-OMPs can aid researchers in identifying promising vaccine targets, developing new antibiotics and therapeutics, and understanding the evolution of Gram-negative bacteria. Despite their distinctive β-barrel structure, OMPs are exposed to numerous charged and polar residues in the membrane, making it challenging to distinguish them from non-OMPs. This is a primary challenge and a significant obstacle in the research process, given the considerable time and capital costs associated with laboratory studies of OMPs. As a result, OMP prediction has tremendous significance for the scientific community. Currently, various machine learning methods have been used for the identification of OMPs, such as support vector machine (SVM) (Park et al., 2005; Gromiha et al., 2006; Hu et al., 2017; Zhang et al., 2021), k-nearest neighbor (K-NN) method (Yan et al., 2008), neural networks (NN) (Hu et al., 2017). These methods utilize the amino acid composition, and physical and chemical properties of the amino acid sequences to construct the prediction models. Gromiha and Suwa (2003); Gromiha and Suwa (2005) developed multiple OMP prediction methods based on amino acid composition, residue pair preference, and motif sequence. However, these methods only achieved prediction accuracies of 80%–90%. Subsequently, a machine learning algorithm was proposed with a higher accuracy ranging from 90% to 94% (Gromiha et al., 2005; Gromiha et al., 2006). Lin (2008) further improved the OMP prediction model by introducing the Incremental Diversity with Quality Distinctness analysis, which combines the Markov discriminant method and the pseudo amino acid composition (Pse-AAC). Despite the progress made in OMP predictions, there is still room for further improvement in prediction quality.

In this article, we proposed a novel method for predicting OMPs that combines Pse-AAC and SVM. To extract the features for amino acid composition and physical and chemical characteristics of amino acids, we used the Pse-AAC feature extraction method. Additionally, we introduced multi-level amino acid residue index correlation coefficients such as hydrophobic value, average polarity, and solvation-free energy to enhance the accuracy of our prediction model. To assess the effectiveness and reliability of our approach, we also conducted a comprehensive comparison and analysis of our proposed model with existing methods for predicting OMPs. Our developed approach will be useful for distinguishing OMPs from non-OMPs.

2 Materials and methods

2.1 Datasets

The construction of a reliable dataset is the basis for developing an accurate outer membrane protein prediction model (Su et al., 2021). A well-designed dataset is crucial for developing effective algorithms and an objective evaluation and prediction system. In this paper, membrane proteins were extracted from the PSORT-B database (https://www.psort.org/) (Gardy et al., 2003), and globular proteins were extracted from the PDB40D of SCOP_1.37 database (http://scop.mrc-lmb.cam.ac.uk/scop/) (Andreeva et al., 2020). As a result, a total of 208 OMPs were selected as the positive set, while 879 non-OMPs were chosen as the negative set. The negative set included 206 inner membrane proteins and 673 globular proteins. The globular protein dataset contained 154 complete α proteins, 156 complete β proteins, 184 $\alpha+\beta$ proteins, and 179 α/β proteins. Since the sequence homology of each protein class was less than 40%, proteins in each database were not similar and were de-redundant.

2.2 Feature encoding

To construct a prediction model, it is necessary to represent the protein sequences as mathematical vectors. This conversion is commonly known as feature extraction (Basith et al., 2020; Dao et al., 2022b; Zhang Z.-Y. et al., 2022; Hunt et al., 2022; Karuna Nidhi et al., 2022; Sun et al., 2022; Tran and Nguyen, 2022; Wang et al., 2022; Yang et al., 2022). The amino acid composition (ACC) of the protein has a great impact on protein classification research (Awais et al., 2021; Shoombuatong et al., 2022b; Manavalan and Patra, 2022; Rout et al., 2022; Zhu et al., 2022). By using the ACC, a protein sequence can be represented as a 20-D (dimension) vector as follows:

$$V_{AAC}(S) = (v_1, v_2, v_3, \cdots, v_{20})^T \tag{1}$$

In Eq. 1, $v_i = f_i / \sum f_i$, f_i represented the number of the i ($i = 1, 2, \cdots, 20$) amino acid in the protein sequence.

The type of amino acids is determined by their side chains, as the 20 types of amino acid side chains differ in shape, size, negativity, hydrophobicity, and acid-base properties. The distinct characteristics of the 20 amino acid side chains result in various combinations of amino acid sequences that exhibit different structures and functions. Therefore, algorithms based on the physicochemical properties of amino acids are another major category of feature extraction methods. Pse-AAC, originally proposed by Chou, is a feature extraction algorithm, that is, based on the physical and chemical properties of amino acids (Chou, 2005). By using Pse-AAC, a protein sample can be represented as follows:

$$V_{PAAC} = [x_1, \ldots, x_{20}, x_{20+1}, \ldots, x_{20+\lambda}]^T \tag{2}$$

where the first 20 numbers in Eq. 2 are the classic AAC features, and the next λ discrete numbers represent the position information of residues in amino acid sequences. For different problems, the optimal value of λ may vary. In this study, we selected the optimal value of λ that yielded the highest sensitivity through the jackknife test.

2.3 Support vector machine

SVM is a powerful supervised machine learning classification method based on statistical learning theory (Manavalan et al., 2019). It was originally designed based on the idea of the generalized linear classifier. First, features were mapped to high-dimensional space. Next, a separating hyperplane is constructed to separate the two categories in the high-dimensional feature space (Vapnik and Control, 2019). To avoid expensive computations, the mapping function only involves the relatively low-dimensional vector in the input space and the dot product

in the feature space. The global optimization approach and avoidance of overfitting in SVM have made it a successful tool for addressing various bioinformatics problems (Zhang H. et al., 2022). In this paper, the support vector machine (SVM) was implemented using the widely used software LIBSVM (http://www.csie.ntu.edu.tw/~cjlin/libsvm) (Chang and Lin, 2011). The radial basis function which is defined as $K(x_i, x_j) = exp(-\gamma\|x_i - x_j\|^2)$ was chosen as the kernel function. The regularization parameter C and the kernel width parameter γ were optimized on the training set using a grid search strategy.

2.4 Evaluation methods

At present, k-fold cross-validation and jackknife cross-validation are widely used for prediction evaluation (Tabaie et al., 2021; Dao et al., 2022a; Xiao et al., 2022; Zhou et al., 2022). The jackknife test is a type of cross-validation that involves leaving one observation out of the dataset at a time and using the remaining observations to train a model. This process is repeated for each observation in the dataset, resulting in n different models, where n is the number of observations in the dataset. In this article, we used the Jackknife test to evaluate the prediction results. The sensitivity (S_n), specificity (S_p), average accuracy (AA), overall prediction accuracy (OA), and Matthew's correlation coefficient (MCC), the area under ROC curve (auROC) were used to evaluate the prediction performance of the algorithm (Yang et al., 2021; Zhang Q. et al., 2022). The evaluation metrics are defined as follows:

$$Sn = \frac{TP}{TP + FN} \tag{3}$$

$$Sp = \frac{TN}{TN + FP} \tag{4}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FN)(TP + FP)(TN + FP)(TN + FN)}} \tag{5}$$

$$OA = \frac{TP + TN}{TP + TN + FN + FP} \tag{6}$$

$$AA = \frac{1}{2} \times \left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right) \tag{7}$$

where TP represents the number of the positive sample correctly identified, FN represents the positive sample wrongly identified as a negative sample, FP represents the negative sample wrongly identified as a positive sample, and TN represents the negative sample correctly identified. AuROC is an indicator that relates to the receiver operating characteristic (ROC) curve, which is a plot of a series of continuous (1-Sp) values on the horizontal axis against their corresponding Sn values on the vertical axis. The ROC curve is a useful tool for evaluating the sensitivity and specificity of a model (Hasan et al., 2022; Jeon et al., 2022). AuROC is calculated in this study as an indicator of classification ability and performance. A larger auROC value indicates better performance and classification ability of the model.

3 Results and discussion

3.1 Model performance

In this study, the proteins were first obtained in FASTA format and then the PseAAC program (Shen and Chou, 2008) was used to extract the feature vectors of pseudo amino acid components. To achieve relatively optimal prediction results, different parameters were selected to extract pseudo amino acid component feature vectors of protein sequences. Specifically, feature vectors were extracted using different values of ω (the weight factor) including 0.1, 0.2, 0.3, 0.4, 0.5, and 0.6, and λ was taken as either 3 or 5. The extracted feature vectors were then used for prediction using different values of γ including 0.04, 0.05, 0.06, 0.07, 0.08, and 0.09. The SVM model was trained using svm-train in LIBSVM, and the optimal parameter array and optimal feature subset were searched from the prediction results. Only the γ value that achieved the optimal prediction result was selected and listed in Table 1.

TABLE 1 The performance comparison of prediction models under different parameter conditions.

ω, λ, γ	Sn(%)	*Sp*(%)	*MCC*(%)	*OA* (%)	*AA* (%)	AuROC
0.1,3,0.05	78.37	96.59	77.16	93.10	87.48	0.962
0.2,3,0.08	78.85	96.59	77.49	93.19	87.72	0.966
0.3,3,0.09	75.96	96.59	75.46	92.64	86.27	0.965
0.4,3,0.08	79.33	95.79	75.97	92.64	87.56	0.962
0.5,3,0.09	79.33	95.79	75.97	92.64	87.56	0.961
0.6,3,0.09	79.81	95.90	76.57	92.82	87.86	0.958
0.1,5,0.07	79.81	96.59	78.17	93.38	88.20	0.965
0.2,5,0.09	79.81	95.56	75.79	92.55	87.69	0.968
0.3,5,0.09	80.77	95.45	76.22	92.64	88.11	0.966
0.4,5,0.08	81.73	95.11	76.15	92.55	88.42	0.964
0.5,5,0.07	84.61	95.11	78.19	93.10	89.86	0.963
0.6,5,0.07	81.25	94.43	74.34	91.90	87.84	0.956

In this study, the benchmark dataset consisted of 208 OMPs and 879 non-OMPs. Due to this imbalanced dataset, using average accuracy as the sole evaluation criterion may lead to skewed results toward the negative sets. Thus, the paper used overall accuracy as the main criterion for model evaluation. By analyzing the data in Table 1, it was observed that high prediction sensitivity was achieved using Jackknife cross-validation with different parameters. And, with the best prediction result obtained with a weight factor of 0.5, the parameter λ taking 5, γ taking 0.07, resulting in an overall accuracy of 93.10%.

3.2 Model comparison

Various methods have been proposed by different researchers to predict and distinguish OMPs from other types of membrane proteins. Wu et al. (2007) proposed a prediction method that uses information differences to compare the distribution of subsequences and residual sequences, resulting in a prediction accuracy of 99.20%. Yan et al. (2008) proposed a method based on the K-nearest neighbor (KNN) method, which predicted the weighted Euclidean distance calculated by residual synthesis and achieved a recognition accuracy of 96.1%, sensitivity of 87.5%,

specificity of 98.2% with 0.873 MCC. Gromiha et al. (2006) discriminate of OMPs and non-OMPs using different machine learning approaches, the best performance achieved sensitivity of 84.6%, specificity of 95.8% and accuracy of 93.7%. And the SVM-based model achieved sensitivity of 72.6%, specificity of 98.2% and accuracy of 93.3%. Park et al. (2005) proposed an SVM method that considers both amino acid composition and residue pair information, achieving sensitivity of 90.9 %, specificity of 94.7%, MCC 0.816 of and accuracy of 93.9%. Gao et al. (2010) developed a method that combined the structural and physicochemical characteristics of sequence-derived proteins with amino acid composition to distinguish OMPs and non-OMPs using SVM, with an overall accuracy of 97.8%, sensitivity of 91.8 %, specificity of 99.2% and MCC 0.928.

In this paper, the model constructed using the SVM algorithm achieved an overall accuracy of 93.10% and auROC of 0.963 under Jackknife cross-validation, respectively. Besides, the sensitivity, specificity, MCC, and average accuracy were found to be 84.61%, 95.11%, 78.19%, and 89.86%, respectively. Compared to previous SVM-based models, some progress has been made.

4 Conclusion

This article focused on the prediction and recognition of OMPs using the method of combining Pse-AAC with SVM. The study achieved good results with the Pse-AAC method, which not only considers the content of 20 natural amino acids in each protein sequence but also includes the correlation between various amino acids, such as physical and chemical properties. This approach is more advanced than traditional methods that only consider amino acid composition, leading to more accurate prediction results. SVM is a widely used algorithm in bioinformatics (Hasan et al., 2020; Shoombuatong et al., 2022a; Bupi et al., 2023), and applying it to the prediction of OMPs is an inevitable trend in current research. The constructed model using the SVM algorithm achieved high performance with an overall accuracy of 93.10% and auROC of 0.963 under Jackknife cross-validation. The sensitivity, specificity, Matthew correlation coefficient, and average accuracy achieved 84.61%, 95.11%, 78.19%, and 89.86%, respectively. However, while feature extraction algorithms have been widely used in prediction methods and have achieved good performance, the relationship between the extracted information and protein structure and function needs to be further explored. This challenge will undoubtedly be the focus of our future research efforts aimed at identifying OMPs. The development of accurate prediction models for OMPs has the potential to significantly impact fields ranging from antibiotic discovery and vaccine development to biotechnology and bacterial diagnostics.

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 authors.

Author contributions

Conceptualization, HD and ZZ; data curation, WS and XQ; formal analysis, WS, XQ, and KY; funding acquisition, WS and ZZ; supervision, CH; writing—original draft, WS; writing—review and editing, CH and ZZ. All authors contributed to the article and approved the submitted version.

Funding

This research was funded by grants from the National Natural Science Foundation of China (Grant Nos. 62201299, 62102067), Natural Science Foundation of the Inner Mongolia of China (Grant No. 2021BS06003), Science and Technology Research Project of Colleges and Universities in Inner Mongolia of China (Grant No. NJZY21473), and Basic Scientific Research Foundation of Colleges and Universities directly under Inner Mongolia of China (Grant No. BR220505).

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.

References

Andreeva, A., Kulesha, E., Gough, J., and Murzin, A. G. J. N. A. R. (2020). The SCOP database in 2020: Expanded classification of representative family and superfamily domains of known protein structures. *Nucleic Acids Res.* 48, D376–D382. doi:10.1093/nar/gkz1064

Awais, M., Hussain, W., Rasool, N., and Khan, Y. D. J. C. B. (2021). iTSP-PseAAC: identifying tumor suppressor proteins by using fully connected neural network and PseAAC. *Curr. Bioinform.* 16, 700–709. doi:10.2174/15748936mtezfmteby

Basith, S., Manavalan, B., Hwan Shin, T., and Lee, G. (2020). Machine intelligence in peptide therapeutics: A next-generation tool for rapid disease screening. *Med. Res. Rev.* 40, 1276–1314. doi:10.1002/med.21658

Budiardjo, S. J., Ikujuni, A. P., Firlar, E., Cordova, A., Kaelber, J. T., and Slusky, J. S. J. T. J. O. M. B. (2021). High-yield preparation of outer membrane protein efflux pumps by *in vitro* refolding is concentration dependent. *J. Membr. Biol.* 254, 41–50. doi:10.1007/s00232-020-00161-y

Bupi, N., Sangaraju, V. K., Phan, L. T., Lal, A., Vo, T. T. B., Ho, P. T., et al. (2023). An effective integrated machine learning framework for identifying severity of tomato yellow leaf curl virus and their experimental validation. *Research* 6, 0016. doi:10.34133/research.0016

Chang, C. C., and Lin, C. J. (2011). Libsvm: A library for support vector machines. *ACM Trans. Intelligent Syst. Technol.* 2, 1–27.

Cheng, L., Qi, C., Yang, H., Lu, M., Cai, Y., Fu, T., et al. (2022). gutMGene: a comprehensive database for target genes of gut microbes and microbial metabolites. *Nucleic Acids Res.* 50, D795–D800. doi:10.1093/nar/gkab786

Chou, K.-C. J. B. (2005). Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes. *Bioinformatics* 21, 10–19. doi:10.1093/bioinformatics/bth466

Dao, F.-Y., Lv, H., Fullwood, M. J., and Lin, H. J. R. (2022a). Accurate identification of DNA replication origin by fusing epigenomics and chromatin interaction information. *Res. (Wash D C)* 2022, 9780293. doi:10.34133/2022/9780293

Dao, F.-Y., Lv, H., Zhang, Z.-Y., and Lin, H. J. C. B. (2022b). BDselect: A package for k-mer selection based on the binomial distribution. *Curr. Bioinforma.* 17, 238–244. doi:10.2174/1574893616666211007102747

Fahie, M. A., Yang, B., Chisholm, C. M., Chen, M. J. N. T. M., and Protocols (2021). Protein analyte sensing with an outer membrane protein G (OmpG) nanopore. *Methods Mol. Biol.* 186, 77–94. doi:10.1007/978-1-0716-0806-7_7

Gao, Q.-B., Ye, X.-F., Jin, Z.-C., and He, J. J. A. B. (2010). Improving discrimination of outer membrane proteins by fusing different forms of pseudo amino acid composition. *Anal. Biochem.* 398, 52–59. doi:10.1016/j.ab.2009.10.040

Gardy, J. L., Spencer, C., Wang, K., Ester, M., Tusnady, G. E., Simon, I., et al. (2003). PSORT-B: Improving protein subcellular localization prediction for Gram-negative bacteria. *Nucleic Acids Res.* 31, 3613–3617. doi:10.1093/nar/gkg602

Gromiha, M. M., Ahmad, S., and Suwa, M. (2005). Application of residue distribution along the sequence for discriminating outer membrane proteins. *Comput. Biol. Chem.* 29, 135–142. doi:10.1016/j.compbiolchem.2005.02.006

Gromiha, M. M., and Suwa, M. J. B. (2005). A simple statistical method for discriminating outer membrane proteins with better accuracy. *Bioinformatics* 21, 961–968. doi:10.1093/bioinformatics/bti126

Gromiha, M. M., Suwa, M. J. B. E. B. A.-P., and Proteomics (2006). Influence of amino acid properties for discriminating outer membrane proteins at better accuracy. *Biochim. Biophys. Acta* 1764, 1493–1497. doi:10.1016/j.bbapap.2006.07.005

Gromiha, M. M., and Suwa, M. J. I. J. O. B. M. (2003). Variation of amino acid properties in all-β globular and outer membrane protein structures. *Int. J. Biol. Macromol.* 32, 93–98. doi:10.1016/s0141-8130(03)00042-4

Gromiha, M. M., and Suwa, M. J. P. S. (2006). Discrimination of outer membrane proteins using machine learning algorithms. *Funct. Bioinforma. Proteins* 63, 1031–1037. doi:10.1002/prot.20929

Hasan, M. M., Schaduangrat, N., Basith, S., Lee, G., Shoombuatong, W., and Manavalan, B. (2020). HLPpred-fuse: Improved and robust prediction of hemolytic peptide and its activity by fusing multiple feature representation. *Bioinformatics* 36, 3350–3356. doi:10.1093/bioinformatics/btaa160

Hasan, M. M., Tsukiyama, S., Cho, J. Y., Kurata, H., Alam, M. A., Liu, X., et al. (2022). Deepm5C: A deep-learning-based hybrid framework for identifying human rna N5-methylcytosine sites using a stacking strategy. *Mol. Ther.* 30, 2856–2867. doi:10.1016/j.ymthe.2022.05.001

Hu, Y., Zhou, M., Shi, H., Ju, H., Jiang, Q., and Cheng, L. (2017). Measuring disease similarity and predicting disease-related ncRNAs by a novel method. *Bmc Med. Genomics* 10, 71. doi:10.1186/s12920-017-0315-9

Hunt, C., Montgomery, S., Berkenpas, J. W., Sigafoos, N., Oakley, J. C., Espinosa, J., et al. (2022). Recent progress of machine learning in gene therapy. *Curr. Gene Ther.* 22, 132–143. doi:10.2174/1566523221666210622164133

Jeon, Y. J., Hasan, M. M., Park, H. W., Lee, K. W., and Manavalan, B. (2022). Tacos: A novel approach for accurate prediction of cell-specific long noncoding RNAs subcellular localization. *Brief. Bioinform* 23, bbac243. doi:10.1093/bib/bbac243

Karuna Nidhi, M. B., Ganapathy, R., Subbiah, P., Suvaiyarasan, S., and Karuppasamy, M. P. J. C. B. (2022). GenNBPSeq: Online web server to generate never born protein sequences using toeplitz matrix approach with structure analysis. *Curr. Bioinform.* 17, 565–577. doi:10.2174/1574893617666220519110154

Lin, H. J. J. O. T. B. (2008). The modified Mahalanobis discriminant for predicting outer membrane proteins by using Chou's pseudo amino acid composition. *J. Theor. Biol.* 252, 350–356. doi:10.1016/j.jtbi.2008.02.004

Manavalan, B., Basith, S., Shin, T. H., Wei, L., and Lee, G. J. B. (2019). mAHTPred: a sequence-based meta-predictor for improving the prediction of anti-hypertensive peptides using effective feature representation. *Bioinformatics* 35, 2757–2765. doi:10.1093/bioinformatics/bty1047

Manavalan, B., and Patra, M. C. J. J. O. M. B. (2022). Mlcpp 2.0: An updated cell-penetrating peptides and their uptake efficiency predictor. *J. Mol. Biol.* 434, 167604. doi:10.1016/j.jmb.2022.167604

Park, K.-J., Gromiha, M. M., Horton, P., and Suwa, M. J. B. (2005). Discrimination of outer membrane proteins using support vector machines. *Bioinformatics* 21, 4223–4229. doi:10.1093/bioinformatics/bti697

Qi, C., Cai, Y., Qian, K., Li, X., Ren, J., Wang, P., et al. (2022). SCovid: Single-cell atlases for exposing molecular characteristics of COVID-19 across 10 human tissues. *Nucleic acids Res.* 50, D867–D874. doi:10.1093/nar/gkab881

Rollauer, S. E., Sooreshjani, M. A., Noinaj, N., and Buchanan, S. K. J. P. T. O. T. R. S. B. B. S. (2015). Outer membrane protein biogenesis in Gram-negative bacteria. *Philos. Trans. R. Soc. Lond B Biol. Sci.* 370, 20150023.

Rout, R. K., Hassan, S. S., Sheikh, S., Umer, S., Sahoo, K. S., and Gandomi, A. H. (2022). Feature-extraction and analysis based on spatial distribution of amino acids for SARS-CoV-2 Protein sequences. *Comput. Biol. Med.* 141, 105024. doi:10.1016/j.compbiomed.2021.105024

Shen, H.-B., and Chou, K.-C. J. a. B. (2008). PseAAC: A flexible web server for generating various kinds of protein pseudo amino acid composition. *Anal. Biochem.* 373, 386–388. doi:10.1016/j.ab.2007.10.012

Shoombuatong, W., Basith, S., Pitti, T., Lee, G., and Manavalan, B. J. J. O. M. B. (2022b). Throne: A new approach for accurate prediction of human RNA N7-methylguanosine sites. *J. Mol. Biol.* 434, 167549. doi:10.1016/j.jmb.2022.167549

Shoombuatong, W., Basith, S., Pitti, T., Lee, G., and Manavalan, B. (2022a). Throne: A new approach for accurate prediction of human rna N7-methylguanosine sites. *J. Mol. Biol.* 434, 167549. doi:10.1016/j.jmb.2022.167549

Su, W., Liu, M.-L., Yang, Y.-H., Wang, J.-S., Li, S.-H., Lv, H., et al. (2021). Ppd: A manually curated database for experimentally verified prokaryotic promoters. *J. Mol. Biol.* 433, 166860. doi:10.1016/j.jmb.2021.166860

Sun, Z., Huang, Q., Yang, Y., Li, S., Lv, H., Zhang, Y., et al. (2022). PSnoD: Identifying potential snoRNA-disease associations based on bounded nuclear norm regularization. *Brief. Bioinform.* 23, bbac240. doi:10.1093/bib/bbac240

Tabaie, A., Orenstein, E. W., Nemati, S., Basu, R. K., Kandaswamy, S., Clifford, G. D., et al. (2021). Predicting presumed serious infection among hospitalized children on central venous lines with machine learning. *Comput. Biol. Med.* 132, 104289. doi:10.1016/j.compbiomed.2021.104289

Tran, H. V., and Nguyen, Q. H. J. C. B. (2022). iAnt: combination of convolutional neural network and random Forest models using PSSM and BERT features to identify antioxidant proteins. *Curr. Bioinform.* 17, 184–195. doi:10.2174/1574893616666210820095144

Vapnik, V. N. J. A., and Control, R. (2019). Complete statistical theory of learning. *Inf. Fusion* 80, 1949–1975.

Wang, P., Zhang, S., He, G., Du, M., Qi, C., Liu, R., et al. (2022). microbioTA: an atlas of the microbiome in multiple disease tissues of *Homo sapiens* and *Mus musculus*. *Nucleic acids Res.* 51, D1345–D1352. doi:10.1093/nar/gkac851

Wu, Z., Feng, E., Wang, Y., Chen, L. J. P., and Letters, P. (2007). Discrimination of outer membrane proteins by a new measure of information discrepancy. *Protein Pept. Lett.* 14, 37–44. doi:10.2174/092986607779117254

Xiao, J., Liu, M., Huang, Q., Sun, Z., Ning, L., Duan, J., et al. (2022). Analysis and modeling of myopia-related factors based on questionnaire survey. *Comput. Biol. Med.* 150, 106162. doi:10.1016/j.compbiomed.2022.106162

Yan, C., Hu, J., and Wang, Y. J. a. A. (2008). Discrimination of outer membrane proteins using a K-nearest neighbor method. *Amino Acids* 35, 65–73. doi:10.1007/s00726-007-0628-7

Yang, H., Luo, Y., Ren, X., Wu, M., He, X., Peng, B., et al. (2021). Risk prediction of diabetes: Big data mining with fusion of multifarious physical examination indicators. *Inf. Fusion* 75, 140–149. doi:10.1016/j.inffus.2021.02.015

Yang, Y., Gao, D., Xie, X., Qin, J., Li, J., Lin, H., et al. (2022). DeepIDC: A prediction framework of injectable drug combination based on heterogeneous information and deep learning. *Clin. Pharmacokinet.* 61, 1–11. doi:10.1007/s40262-022-01180-9

Yu, H., Shen, Z.-A., Zhou, Y.-K., and Du, P.-F. (2022). Recent advances in predicting protein-lncRNA interactions using machine learning methods. *Curr. Gene Ther.* 22, 228–244. doi:10.2174/1566523221666210712190718

Zhang, H., Wang, S., and Huang, T. (2021). Identification of chronic hypersensitivity pneumonitis biomarkers with machine learning and differential Co-expression analysis. *Curr. Gene Ther.* 21, 299–303. doi:10.2174/1566523220666201208093325

Zhang, H., Zou, Q., Ju, Y., Song, C., and Chen, D. J. C. B. (2022a). Distance-based support vector machine to predict DNA N6-methyladenine modification. *Curr. Bioinform.* 17, 473–482. doi:10.2174/1574893617666220404145517

Zhang, Q., Li, H., Liu, Y., Li, J., Wu, C., and Tang, H. J. C. O. (2022b). Exosomal non-coding RNAs: New insights into the biology of hepatocellular carcinoma. *Curr. Oncol.* 29, 5383–5406. doi:10.3390/curroncol29080427

Zhang, Z.-Y., Ning, L., Ye, X., Yang, Y.-H., Futamura, Y., Sakurai, T., et al. (2022c). iLoc-miRNA: extracellular/intracellular miRNA prediction using deep BiLSTM with attention mechanism. *Brief. Bioinform.* 23, bbac395. doi:10.1093/bib/bbac395

Zhou, H., Wang, H., Ding, Y., and Tang, J. J. C. B. (2022). Multivariate information fusion for identifying antifungal peptides with Hilbert-Schmidt Independence Criterion. *Curr. Bioinform.* 17, 89–100. doi:10.2174/1574893616666210727161003

Zhu, Z., Han, X., and Cheng, L. (2022). Identification of gene signature associated with type 2 diabetes mellitus by integrating mutation and expression data. *Curr. Gene Ther.* 22, 51–58. doi:10.2174/1566523221666210707140839

frontiers | Frontiers in Genetics

TYPE Original Research
PUBLISHED 24 July 2023
DOI 10.3389/fgene.2023.1232117

OPEN ACCESS

EDITED BY
Nathan Olson,
National Institute of Standards and Technology (NIST), United States

REVIEWED BY
Sandip Debnath,
Visva-Bharati University, India
Jinlun Zhang,
Fudan University, China

*CORRESPONDENCE
Yu Tian,
tianyu21@mails.tsinghua.edu.cn
Xiaoxiao Liu,
liuxiaoxiao@gdph.org.cn

RECEIVED 31 May 2023
ACCEPTED 11 July 2023
PUBLISHED 24 July 2023

CITATION
Wang Y, Wang L, Li C, Pei Y, Liu X and Tian Y (2023), AMP-EBiLSTM: employing novel deep learning strategies for the accurate prediction of antimicrobial peptides.
Front. Genet. 14:1232117.
doi: 10.3389/fgene.2023.1232117

AMP-EBiLSTM: employing novel deep learning strategies for the accurate prediction of antimicrobial peptides

Yuanda Wang[1], Liyang Wang[2], Chengquan Li[2], Yilin Pei[2], Xiaoxiao Liu[3]* and Yu Tian[4]*

[1]School of Modern Post (School of Automation), Beijing University of Posts and Telecommunications, Beijing, China, [2]School of Clinical Medicine, Tsinghua University, Beijing, China, [3]Laboratory Medicine, Guangdong Provincial People's Hospital (Guangdong Academy of Medical Sciences), Southern Medical University, Guangzhou, China, [4]Vascular Surgery Department, Shanxi Bethune Hospital, Shanxi Academy of Medical Sciences, Tongji Shanxi Hospital, Third Hospital of Shanxi Medical University, Taiyuan, China

Antimicrobial peptides are present ubiquitously in intra- and extra-biological environments and display considerable antibacterial and antifungal activities. Clinically, it has shown good antibacterial effect in the treatment of diabetic foot and its complications. However, the discovery and screening of antimicrobial peptides primarily rely on wet lab experiments, which are inefficient. This study endeavors to create a precise and efficient method of predicting antimicrobial peptides by incorporating novel machine learning technologies. We proposed a deep learning strategy named AMP-EBiLSTM to accurately predict them, and compared its performance with ensemble learning and baseline models. We utilized Binary Profile Feature (BPF) and Pseudo Amino Acid Composition (PSEAAC) for effective local sequence capture and amino acid information extraction, respectively, in deep learning and ensemble learning. Each model was cross-validated and externally tested independently. The results demonstrate that the Enhanced Bi-directional Long Short-Term Memory (EBiLSTM) deep learning model outperformed others with an accuracy of 92.39% and AUC value of 0.9771 on the test set. On the other hand, the ensemble learning models demonstrated cost-effectiveness in terms of training time on a T4 server equipped with 16 GB of GPU memory and 8 vCPUs, with training durations varying from 0 to 30 s. Therefore, the strategy we propose is expected to predict antimicrobial peptides more accurately in the future.

KEYWORDS

antimicrobial peptides, diabetic foot, deep learning, ensemble learning, accurate screening

1 Introduction

Antimicrobial peptides are a class of small peptide molecules widely present both inside and outside of organisms, possessing strong antibacterial and antifungal properties (Zasloff, 2002). Their mechanism of action primarily involves disrupting microbial cell membranes, leading to cell death (Brogden, 2005). The biological structure of antimicrobial peptides usually encompasses various amino acids, offering a broader antimicrobial spectrum and lower resistance than traditional antibiotics (Hancock and Sahl, 2006; Mahlapuu et al., 2016). This makes them promising candidates for applications in biomedical, food preservation,

cosmetic antimicrobial, and environmental protection (Axel et al., 2016; Kumar et al., 2018; Yoon et al., 2018) fields. For instance, in the medical domain, antimicrobial peptides serve as topical anti-infective drugs, treating skin and soft tissue infections and preventing and treating hospital-acquired infections (Omardien et al., 2016; Nakatsuji et al., 2017; Lázár et al., 2018). They are also extensively employed in medical device applications, such as coatings on pacemakers, artificial joints, and dental implants, to prevent the formation of bacterial biofilms and reduce device-related infection risks (Melo et al., 2009). In clinical therapy, antimicrobial peptides have gradually attracted attention as potential alternative antibiotic treatments, demonstrating promising potential in wound healing, infectious disease treatment, and antitumor therapy (Costa et al., 2011; Hilchie et al., 2013; Mansour et al., 2014). Additionally, antimicrobial peptides have gradually attracted people's attention as a potential alternative to antibiotic therapy and to promote the formation of new blood vessels. For example, in the clinical practice of vascular surgery, some antimicrobial peptides have been successfully used in the treatment of diabetic feet, such as LL -37 and hBDs, both of which exhibit good anti-bacterial and wound healing effects (Lázár et al., 2018; Da et al., 2021). At the same time, some studies have found that antimicrobial peptides can regulate the function of endothelial cells, pro-mote the formation of new blood vessels, and improve blood flow in the feet, thereby positively affecting the vascular lesions of diabetic feet (Xing et al., 2023).

Traditional screening methods for antimicrobial peptides include biochemical methods and molecular dynamics simulation techniques. Biochemical methods typically involve extracting peptide segments from biological samples and screening them through antimicrobial activity tests, such as the agar diffusion test and minimum inhibitory concentration determination (Hancock and Diamond, 2000; Wimley, 2010). Molecular dynamics simulations, as a bioinformatics approach, offer a new perspective for antimicrobial peptide screening. By simulating the interactions between antimicrobial peptides and bacterial target molecules, researchers can gain deeper insights into the mechanism of action of antimicrobial peptides, thereby optimizing their design and screening. Molecular dynamics simulation technology can assist researchers in screening peptide segments with higher antimicrobial activity, thereby enhancing the efficiency of antimicrobial peptide re-search and applications (Haney et al., 2017; Ulmschneider and Ulmschneider, 2018).

However, the development and screening of antimicrobial peptides currently face a series of challenges. Firstly, traditional biochemical methods are costly and have lengthy development cycles. These methods require laboratory screening of numerous peptide segments, which can consume substantial time and resources. Furthermore, due to experimental condition constraints, false-positive or false-negative results may be generated, thereby reducing the accuracy of the screening. Although molecular dynamics simulations, as a bioinformatics approach, have somewhat improved screening efficiency, they still present shortcomings. The simulation process might be con-strained by computational resources, resulting in less accurate results. Moreover, the variety of antimicrobial peptides screened might be limited, and their stability may not be sufficient to meet practical application requirements. Consequently, the development of an efficient, precise, and convenient screening strategy is crucial. Such a strategy should overcome the limitations of current screening methods, enhance screening efficiency and accuracy, and reduce research and development costs.

With the rapid advancement of AI technology and computational power, an in-creasing number of researchers have begun to focus on the identification of small functional peptides. These small peptides have shorter amino acid sequences, typically containing between 5 and 50 amino acid residues (Al-Khdhairawi et al., 2023). These short peptides play various crucial functions in biological systems, including antimicrobial, antiviral, immunoregulatory, and cellular signal transduction roles (Hancock et al., 2016). Optimized machine learning algorithms can enhance the accuracy and efficiency of identifying and predicting functional peptides, deepening our understanding of their roles in biological systems and providing robust support for related field research. Over the past few years, significant progress has been made in peptide recognition work. Meher et al. improved the accuracy of antimicrobial peptide prediction by integrating compositional, physicochemical, and structural features into the Pseudo Amino Acid Composition (PSEAAC) (Chou, 2001; Meher et al., 2017). Veltri et al. improved antimicrobial peptide identification in their research using deep learning methods (Veltri et al., 2018). Manavalan et al. enhanced prediction accuracy by using ma-chine learning and ensemble learning methods to predict cell-penetrating peptides and their engulfment efficiency (Manavalan et al., 2018). Hasan et al. proposed an improved and robust method for predicting hemolytic peptides and their activity—HLPpred-Fuse. They enhanced prediction performance by fusing various feature representations, such as amino acid composition, dihedral angles, amino acid sequence, and PSEAAC, and used a random forest (RF) for model training (Hasan et al., 2020). Although existing research has made some break-throughs in identifying antimicrobial peptides, the precision of prediction and the efficiency of screening still need improvement. These methods might encounter low computational efficiency and high time costs when handling large-scale datasets. While existing methods have contributed significantly to the identification of these peptides, there's a need for more versatile approaches that can rapidly adapt to diverse identification requirements. Furthermore, some models' generalization capability on new datasets needs to be strengthened. Hence, our work presents a new approach that addresses this gap, by developing a prediction model that offers flexibility and efficiency in identifying antimicrobial peptides under diverse conditions.

The aim of this study is to develop an accurate and efficient antimicrobial peptide screening strategy using novel deep learning models. We constructed two datasets: the first for training and five-fold cross-validation, and the second for external independent testing. We proposed the Enhanced Bi-directional Long Short-Term Memory (EBiLSTM) deep learning model and compared it with mainstream ensemble learning and baseline models. In particular, our model incorporates feature fusion strategies to combine different feature types and extract comprehensive characteristics from the peptide sequences. Additionally, a multi-scale convolutional layer is used to capture peptide sequence features at various scales. These modifications aim to improve the model's

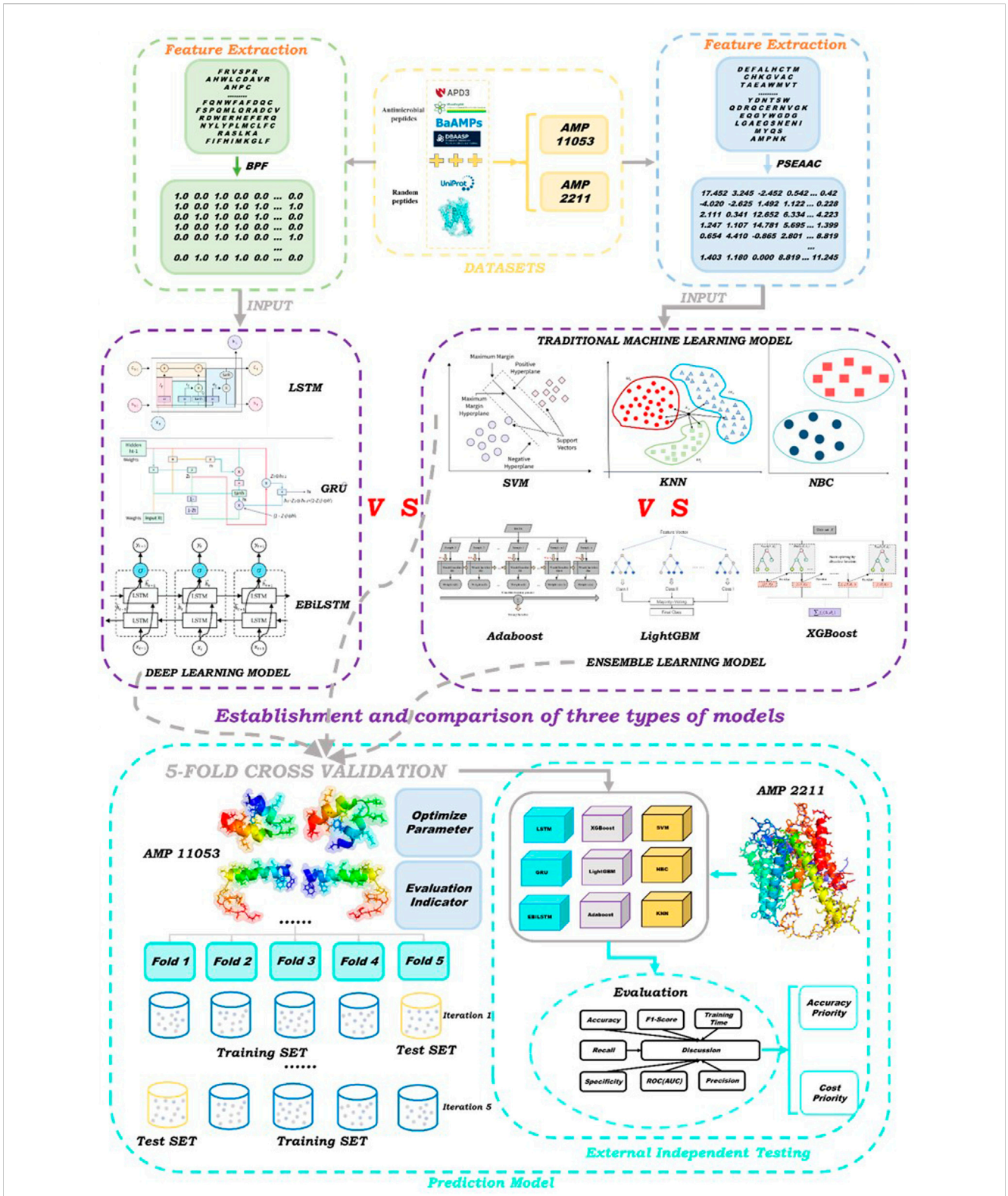


FIGURE 1
Schematic showing experimental workflow.

ability to recognize various features within peptide sequences, thereby enhancing its predictive performance for identifying antimicrobial peptides. For ensemble learning, we utilized Adaptive Boosting (AdaBoost), Light Gradient Boosting Machine (LightGBM), and Extreme Gradient Boosting (XGBoost). In terms of deep learning, in addition to EBiLSTM, two classic deep learning

models were also selected to participate in the work. Additionally, we employed three baseline machine learning models for comparison with the aforementioned six types. These models were tested on an external dataset to evaluate their performance. The specific workflow of this study is illustrated in Figure 1.

In summary, the main contributions of this study are as follows.

- This study is the first to propose EBiLSTM models for antimicrobial peptides prediction. In detail, we made suitable modifications based on the BiLSTM network structure to enhance prediction performance.
- Regarding the dataset, we independently constructed two antimicrobial peptide datasets: one for cross-validation and another for independent verification. This provides a reliable foundation for evaluating and comparing the performance of different models.
- Considering the characteristics of different models, we separately adopted two feature extraction methods: PSEAAC and Binary Profile Feature of k-spaced Amino Acid Pairs (BPF) (Chen et al., 2016). These methods are designed to maximize the potential of each model in antimicrobial peptide prediction tasks.

2 Materials and methods

2.1 Data collection

The antimicrobial peptide data used in this study are all sourced from multiple public databases, including: APD3 (https://aps.unmc.edu/about), PlantPepDB (http://14.139.61.8/PlantPepDB/index.php), BaAMPs (https://www.baamps.it/), Bio-PepDB (https://bis.zju.edu.cn/biopepdbr/), CAMP (https://webs.iiitd.edu.in/raghava/satpdb/catalogs/camp/), DBAASP (https://dbaasp.org/home), DRAMP (https://dramp.cpu-bioinfor.org/), LAMP (https://ngdc.cncb.ac.cn/databasecommons/database/id/4562). After screening, we obtained 5605 and 1119 antimicrobial peptide samples from these databases, respectively. Simultaneously, to construct a comparable proportion of negative samples to the antimicrobial peptide samples, we referred to previous studies (Tyagi et al., 2013; Kumar et al., 2015) and randomly selected the corresponding number of peptide sequences from the UniProt database. These negative samples primarily included peptides that are non-antimicrobial. We aimed to ensure a balance with our positive samples, and therefore, additional criteria were considered in their selection. We ensured that these samples were similar to our positive samples in terms of length, to prevent the length from becoming a distinguishing feature. We also took into account the amino acid composition, ensuring that the negative samples did not exhibit any uncommon composition that could introduce bias. Peptide sequences were added to two datasets, which we named AMP-11053 and AMP-2211. The AMP-11053 dataset was used for model training and internal validation (i.e., five-fold cross-validation), while the AMP-2211 dataset was used for external independent testing to evaluate the model's generalization performance. After the construction of the datasets, we ensured that there were no duplicate peptide sequences within or between the two datasets through careful verification. This procedure helps to ensure the reliability of model training and evaluation.

2.2 Peptide sequence feature representation

To fully tap into the potential of different models for antimicrobial peptide identification tasks, we adopted a variety of model types in this study. Considering the characteristics of each type of model, we chose different feature extraction methods to match their respective applicability. Specifically, for ensemble learning and traditional machine learning models, we utilized the PSEAAC feature extraction method, which has demonstrated commendable performance in many bioinformatics problems. For deep learning models, we selected the BPF feature extraction method. This method effectively captures the local features of sequences, thereby enhancing the performance of the models.

2.2.1 Binary profile feature of k-spaced amino acid pairs

BPF is a feature extraction method used to characterize protein sequences. It con-siders the binary representation of amino acid pairs with k intervals in the amino acid sequence, thereby capturing the relationship between locally adjacent amino acids. After determining the value of k, the BPF algorithm constructs a binary matrix with 20 × 20 rows and columns equivalent to the sequence length minus k. The matrix is populated based on the occurrence of amino acid pairs in the sequence. If a specific pair appears in the sequence, the corresponding position in the matrix is filled with 1; otherwise, it is filled with 0. The binary matrix is then flattened into a feature vector for subsequent analysis.

To determine the appropriate value of k, we extracted 15% of the data from the AMP-11053 dataset as a pre-experimental dataset and conducted pre-experiments with k set to 0, 1, 2, 3, 4, and 5, respectively. The average AUC value was calculated through five-fold cross-validation, and the AUC curve was plotted. The results showed that the AUC value was highest when k = 3, so we selected k = 3 as the parameter for the BPF method in this study. Subsequently, the AMP-11053 and AMP-2211 datasets processed using the BPF method were used as inputs for the deep learning models.

2.2.2 Pseudo amino acid composition

PSEAAC is a feature extraction method widely applied in the field of bioinformatics, primarily used to represent protein sequences. This method integrates both local and global features of amino acid sequences to generate a feature vector of fixed length. Any peptide sequence can be represented as shown in Equation 1, with the specific calculation formula x_μ for different subscripts as given in Equation 2. Here, the integer λ represents the highest order of sequence correlation, and ω is a weight coefficient between 0 and 1. $f_i\,(i = 1, 2, ...20)$ represents the frequency of occurrence of the 20 natural amino acids in the peptide, and $\theta_j\,(j = 1, 2, ..., \lambda)$ denotes the correlation factor of order j, which is defined as shown in Equation 3. The correlation function is calculated according to Equation 4, where $X_1\,(R_i)$,

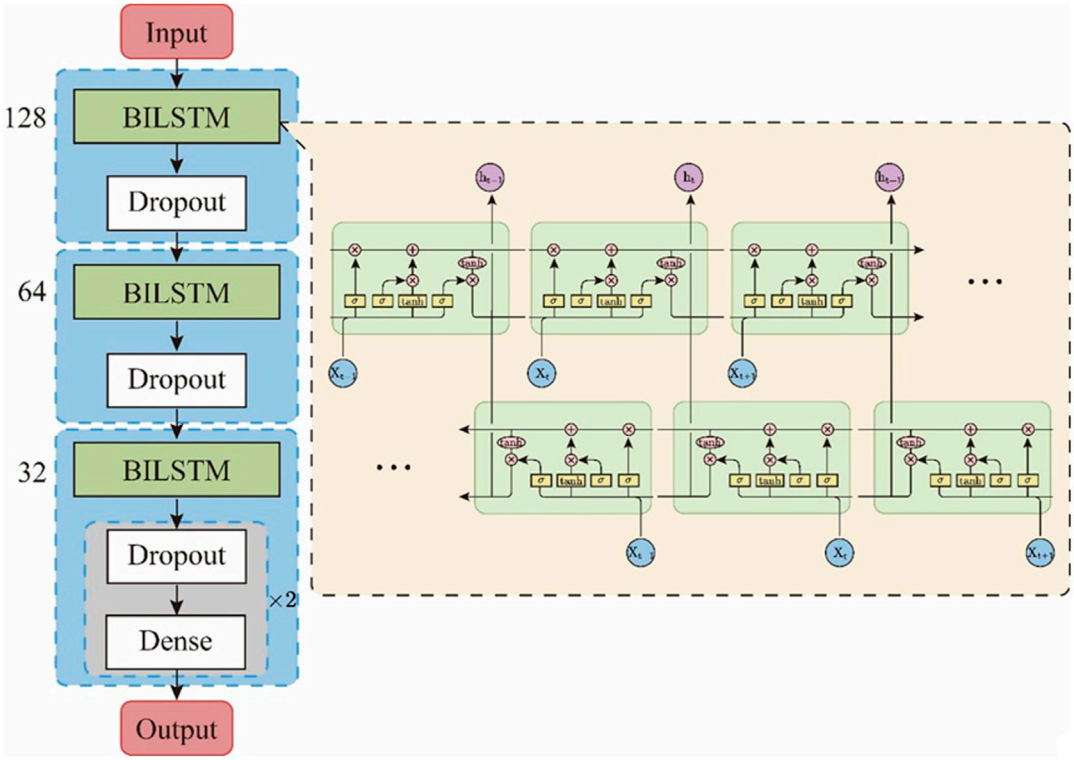


FIGURE 2
Structure of the EBiLSTM model.

$X_2(R_i), \ldots X_n(R_i)$ represent the physicochemical properties of R_i (Ge et al., 2020).

$$P = [x_1, x_2, \ldots x_{20}, x_{20+1}, \ldots x_{20+\lambda}] \tag{1}$$

$$x_\mu = \begin{cases} \dfrac{f_\mu}{\sum_{i=1}^{20} f_i + \omega \sum_{j=1}^{\lambda} \theta_j} \; (1 \le \mu \le 20) \\ \dfrac{\omega \theta_{\mu-20}}{\sum_{i=1}^{20} f_i + \omega \sum_{j=1}^{\lambda} \theta_j} \; (20+1 \le \mu \le 20+\lambda) \end{cases} \tag{2}$$

$$\theta_j = \frac{1}{L-j} \sum_{i=1}^{L-j} \Theta\left(R_i, R_{i+j}\right) (1 \le j \le \lambda) \tag{3}$$

$$\Theta\left(R_i, R_j\right) = \frac{1}{n}\left\{\left[X_1(R_i) - X_1\left(R_j\right)\right]^2 + \left[X_2(R_i) - X_2\left(R_j\right)\right]^2 + \ldots + \left[X_n(R_i) - X_n\left(R_j\right)\right]^2\right\} \tag{4}$$

PSEAAC has two types: Type 1 and Type 2. In this study, we employed the Type 2 PSEAAC approach and selected six physicochemical properties, namely, 'Hydrophobicity,' 'Hydrophilicity,' 'Mass,' 'pK1' (acid dissociation constant), 'pK2' (base dissociation constant), and 'pI' (isoelectric point). In the experiment, the weight was set to 0.05, and the sequence interval (lambda) was set to 2. The processed features were then in-putted into ensemble learning models and baseline machine learning models for further analysis.

2.3 Deep learning model construction

2.3.1 Enhancing bidirectional long short-term memory

BiLSTM is a unique variant of LSTM networks designed to consider both forward and backward information in an input sequence (Schuster and Paliwal, 1997). Traditional LSTM networks process sequence data in a forward manner, unable to capture information from future elements. However, BiLSTM enhances this by adding a parallel LSTM layer to the original, which processes the input sequence in reverse order. This bidirectional characteristic empowers the model to grasp the context of both preceding and subsequent sequences at any given point, furnishing a more comprehensive apprehension of the sequence context. Such a feature makes BiLSTM superior to traditional LSTM in tasks with bidirectional dependencies, such as part-of-speech tagging, named entity recognition, semantic role labeling, and more, offering significant advantages in the identification of antimicrobial peptides.

In this study, we designed an EBiLSTM model composed of three BiLSTM net-works, as illustrated in Figure 2. Our model accepts input of size (100, 20), corresponding to sequence data with a length of 100 and feature dimension of 20. The model begins with a bidirectional LSTM layer containing 128 units and a dropout ratio of 0.5, followed by a dropout layer of 0.3. Subsequent bidirectional LSTM layers with 64 and 32 units, each followed by dropout layers,

form a structure that reduces lay-er-by-layer and helps prevent overfitting. Finally, the model ends with two fully connected layers. The first layer contains 32 nodes and uses the 'relu' activation function, while the latter has 2 nodes and uses the 'sigmoid' activation function to predict the probability for each category. A dropout layer is also placed between these two layers. The model has proven to perform exceptionally, boasting high classification accuracy and robust performance.

2.3.2 Long short term memory

Long Short-Term Memory (LSTM) is a unique form of recurrent neural network proposed by Hochreiter and Schmidhuber. It exhibits a remarkable memory capacity and is particularly adept at handling long sequence data, effectively sidestepping is-sues of "gradient vanishing" or "gradient explosion" (Hochreiter and Schmidhuber, 1997). In antimicrobial peptide recognition, LSTM has been proven to be a potent tool. For instance, Wang et al. utilized a parallel combination of Convolutional Neural Networks (CNN) and LSTM to identify anticancer peptides (Wang H. et al., 2021), and Christina Wang and colleagues employed LSTM to design short novel AMP sequences with potential antimicrobial activity (Wang C. et al., 2021). In this study, our network comprises multiple LSTM layers, which transmit outputs layer by layer to take full advantage of the depth of the model. To prevent overfitting, we introduced a dropout layer after each LSTM layer with a dropout rate set at 0.3. The network finally employs a fully connected layer with a sigmoid activation function to output the prediction results. Adam was chosen as the optimizer, with a learning rate set at 0.01, and a fixed random seed value of 50 was used to ensure consistency.

2.3.3 Gate recurrent unit

The Gated Recurrent Unit (GRU) is an improved variant of the Recurrent Neural Network (RNN) proposed by Cho et al., in 2014 (Cho et al., 2014). By introducing two novel gating mechanisms - the update and reset gates, GRU effectively retains long-term dependency information and enhances model performance. Compared to LSTM's four types of gates, the GRU's structure is more concise, with fewer parameters and higher computational efficiency, yet its performance in various tasks is not inferior to LSTM's. For instance, a model developed by Choi et al., which is based on GRU, successfully predicted patient diagnoses, drug prescriptions, and future disease risks (Choi et al., 2016). In designing the GRU deep learning network for this study, we followed design principles similar to those used with LSTM, ensuring that the model maintains high computational efficiency and robust performance when handling complex tasks. In our preliminary trials, these three models, each showcasing distinct strengths in managing sequence data, emerged as the superior performers in predicting antimicrobial peptides. Consequently, we selected them for our research.

2.4 Ensemble learning model construction

2.4.1 Adaptive boosting

AdaBoost is a powerful ensemble learning technique, central to which is the concept of integrating multiple weak classifiers to enhance model performance (Freund and Schapire, 1997). In bioinformatics, as shown in research by Haoyi Fu et al., AdaBoost has been successfully applied to identify the structure and physicochemical properties of antimicrobial pep-tides (Fu et al., 2020). We chose Adaboost for its remarkable capability to concentrate on challenging-to-classify instances by progressively emphasizing the data misclassified by the preceding classifier. In this study, we used a decision tree as the base classifier, set an iteration limit of 200 to avoid overfitting, controlled the step size of the training process with a learning rate of 0.05, and selected 'SAMME.R' as the algorithm scheme to achieve genuine boosting effects. To ensure the consistency of the experimental results, we set a fixed random seed value of 50. These settings allowed our AdaBoost classifier to achieve a good balance in terms of robustness and stability.

2.4.2 Light gradient boosting machine

LightGBM, developed by Microsoft Research (Ke et al., 2017), is an efficient and accurate gradient boosting decision tree algorithm characterized by its rapid training speed and low memory usage. It employs a histogram-based gradient boosting technique and a leaf-wise growth strategy, effectively enhancing training speed and optimizing the handling of imbalanced data, it is also recognized for its superior accuracy, a critical attribute essential for our study. In our study, key parameters were set as follows: 'num_leaves' was set to 20 to control model complexity and prevent overfitting; 'min_data_in_leaf' was also set to 20 to further guard against overfitting; the depth of the decision tree was unrestricted; the learning rate was set at 0.3 to ensure a balance between training speed and performance; 100 trees were used for fitting; a binary loss function was selected; the traditional gradient boosting decision tree method was employed; and the random seed was set to 40 to ensure the reproducibility of the experiment.

2.4.3 Extreme gradient boosting

XGBoost is an advanced algorithm centered around gradient boosting decision trees, developed by Chen et al. (Chen and Guestrin, 2016). It is highly acclaimed for its superior predictive power and efficient computational speed. By using the second-order derivative information of the objective function and a regularization term, XGBoost optimizes predictive accuracy. Furthermore, by introducing column block data storage and performing parallel and distributed optimizations, it greatly enhances computational efficiency. By utilizing a more regularized model formulation to curb overfitting, it demonstrates superior performance over other models across a range of datasets. XGBoost has also been applied in the medical field; for instance, Junjie Huang et al. utilized it in their machine learning pipeline to identify potent antimicrobial peptides across the entire peptide sequence space (Huang et al., 2023). In our study, the CART tree was chosen as the base learner, and the maximum depth of weak learners and the maximum number of trees were set to 6 and 10, respectively, to prevent overfitting. The learning rate was set to 0.1 to control the step size of iterative updates, and the subsample ratio was set to 0.2 to enhance the model's generalization ability. Additionally, the random seed value was set to 50 to enhance model stability. These settings enabled the XGBoost model to achieve excellent results in terms of predictive performance and stability.

2.5 Baseline model

To comprehensively evaluate the performance of our models, we chose to com-pare them against traditional machine learning models often used in small peptide screening, such as the Support Vector Machine (SVM), Naive Bayes Classifier (NBC), and K-Nearest Neighbors (KNN) (Manavalan et al., 2017; Khabbaz et al., 2021; Wani et al., 2021; Jiang et al., 2022). For SVM, we employed the Gaussian radial basis function kernel to address non-linear classification problems, set the C parameter to 1.0 to balance misclassification penalties, enabled the probability option to output prediction probabilities, and allowed the model to optimize the gamma parameter automatically. We used Gaussian Naive Bayes as it assumes that the continuous features follow a Gaussian distribution. For KNN, we set the number of neighbors, k, to 5 to balance bias and variance and used Euclidean distance as the metric. By comparing these traditional models, we further validated the performance and robustness of our deep learning and ensemble learning models.

2.6 Experiment

In this study, we adopted the widely accepted method of five-fold cross-validation for model training on the AMP-11053 dataset. This approach divides the dataset into five portions, with four of them being used for training and the remaining one for validation. By alternating the training and validation sets, five rounds of training and validation were conducted, with the final model performance evaluation result being the average of the five validation results. Throughout the model training process, we performed parameter optimization on all models to achieve optimal performance. On the AMP-2211 dataset, we carried out independent testing to further validate the models' generalization capability. The experimental environment was configured as follows: we used a T4 server with 16 GB of GPU memory and 8 vCPUs, equipped with 32 GB of RAM, running on a Linux operating system. We utilized the Python 3.8 programming language for model writing and training, relying on machine learning libraries such as Tensorflow 2.2.0 and Scikit-learn 1.2.2 for the construction of deep learning models and implementation of traditional machine learning models. This setup strikes a balance between abundant computational resources and the use of common, easily accessible hardware devices, aiming to ensure the replicability of our study's results.

2.7 Model evaluation

To comprehensively assess model performance, we adopted metrics such as Ac-curacy, Recall, Specificity, Precision, F1-Score, and AUC value, and also plotted ROC curves (Bradley, 1997; Sokolova and Lapalme, 2009; Powers, 2020). TP, TN, FP, FN in the confusion matrix are the primary evaluation parameters, representing true positives, true negatives, false positives, and false negatives. Accuracy calculates the proportion of samples that the model correctly predicts, Precision measures the proportion of true positive samples in those predicted as positive, while Specificity reflects the proportion of true negative samples that were correctly predicted. The F1-score is the harmonic mean of precision and recall. Through the ROC curve, we can see the classifier's performance under all possible classification thresholds, and the area under the curve (AUC) quantifies the overall performance of the classifier. The closer the AUC value is to 1, the better the model performance. In detail:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Specificity = \frac{TN}{TN + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{9}$$

3 Results

3.1 Statistical results of amino acids in the dataset

The two research datasets, AMP-11053 and AMP-2211, encompass amino acid sequences of AMPs and non-antimicrobial peptides, incorporating 20 common natural amino acids. Figure 3A depicts the distribution of amino acid frequencies in AMPs and non-AMPs across both datasets. Upon close inspection, we can observe a degree of similarity in the distribution of amino acid frequencies between AMPs and non-AMPs, which not only reflects the complexity of the classification task but also underscores the challenges and value of this research. Furthermore, the analysis of sequence length distribution between AMPs and non-AMPs is shown in Figure 3B, with most AMP sequence lengths falling between 5 and 50 amino acids. Similarly, non-AMP sequences also have a rich distribution within this length range.

3.2 Deep learning model results

In the experiments conducted on the AMP-11053 dataset, we utilized LSTM, GRU, and EBiLSTM as models for training. During the training process, each model was trained 500 times. We employed multi-class logarithmic loss as the loss function, and the accuracy served as the evaluation standard. The training strategy involved five-fold cross-validation to ensure more stable and reliable model evaluations. We implemented an early termination criterion which stops the training process if there's no improvement in the validation set performance over a defined number of epochs. This strategy not only conserves computational resources, but also aids in preventing the model from assimilating noise present in the training data. Simultaneously, we also computed the evaluation metrics mentioned in Section 3.1. The training results, which include the values of each evaluation metric, the AUC curve, and the ROC values, are shown in Table 1 and Figure 4A. Similarly, we tested the models' generalization

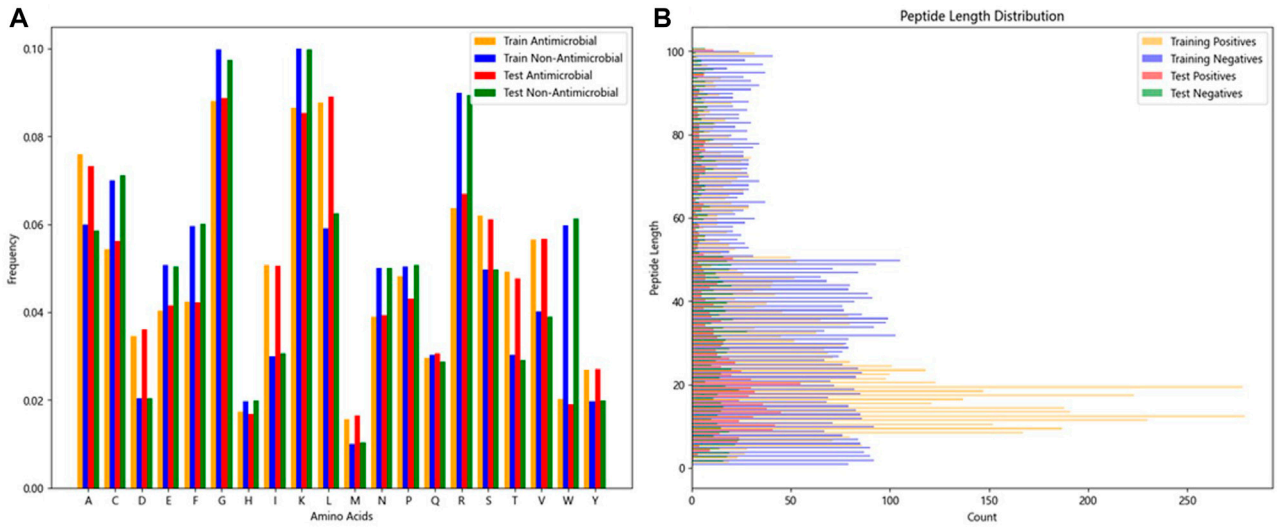


FIGURE 3
Features of peptide chains from two datasets.**(A)** Frequency distribution ratio of various amino acids in peptides.**(B)** Distribution of peptide lengths.

TABLE 1 Performance of the deep learning models on AMP-11053.

AMP-11053	Accuracy	Recall	Specificity	Precision	F1-score
EBILSTM	0.9685 ± 0.0408	0.9619 ± 0.0426	0.9654 ± 0.0388	0.9663 ± 0.0376	0.9699 ± 0.0401
LSTM	0.9383 ± 0.0329	0.9139 ± 0.0472	0.9558 ± 0.0207	0.95640 ± 0.0213	0.9375 ± 0.0334
GRU	0.927 ± 0.0512	0.9489 ± 0.0097	0.9043 ± 0.1004	0.912 ± 0.0768	0.9265 ± 0.0440

capabilities on an additional external dataset, AMP-2211. The testing results are presented in Table 2 and Figure 4B. It is not difficult to find that the proposed EBiLSTM has the most excellent performance both in the training set and the external test set.

While our models achieved excellent performance on the AMP-11053 dataset, as evidenced by the evaluation metrics, AUC curve, and ROC values in Table 1 and Figure 4A, the performance on the external dataset AMP-2211, depicted in Table 2 and Figure 4B, was marginally lower. It is crucial to note that this dip in performance, while important to acknowledge, is not entirely unexpected. When applying a model trained on one dataset (AMP-11053) to a different dataset (AMP-2211), it is common to see some decrease in performance. This is due to the inherent differences between the datasets, which might include variations in complexity, distribution of data, or the amount and type of noise present. Essentially, the AMP-2211 dataset presents previously unseen scenarios for the model, and it is natural that the model will not perform as effectively on this new data as on the data it was trained on. However, this difference in performance can actually be seen as a positive. If our model performed identically on both datasets, it would raise concerns about overfitting. Overfitting occurs when a model learns the training data too well, to the point where it is too specialized to the training data and performs poorly on new, unseen data. The fact that our model's performance decreases slightly on the external AMP-2211 dataset suggests that our model is not overfitted and is capable of generalizing to new data.

3.3 Ensemble learning model results

In the case of the AMP-11053 dataset, we trained using Adaboost, LightGBM, and XGBoost, employing a five-fold cross-validation method. We calculated five main evaluation metrics: accuracy, recall, specificity, precision, and F1-Score. During the process of evaluating model performance, to accurately assess model capabilities, we also calculated the 95% confidence interval for these metrics. Specific details are shown in Table 3, while the AUC curves derived from the three types of models are depicted in Figure 5A. To further validate the models' generalization capabilities, we employed an additional external dataset, AMP-2211, to test the models. In the testing process, we calculated the aforementioned five evaluation metrics and drew the AUC curve. Test results are displayed in Table 4 and Figure 5B. These results provide us with a comprehensive and in-depth understanding, allowing us to assess and compare the performance of different models on multiple levels. The results above indicate that while ensemble learning demonstrates considerable accuracy in identifying antimicrobial peptides, its performance is still not on par with that of deep learning, especially EBiLSTM.

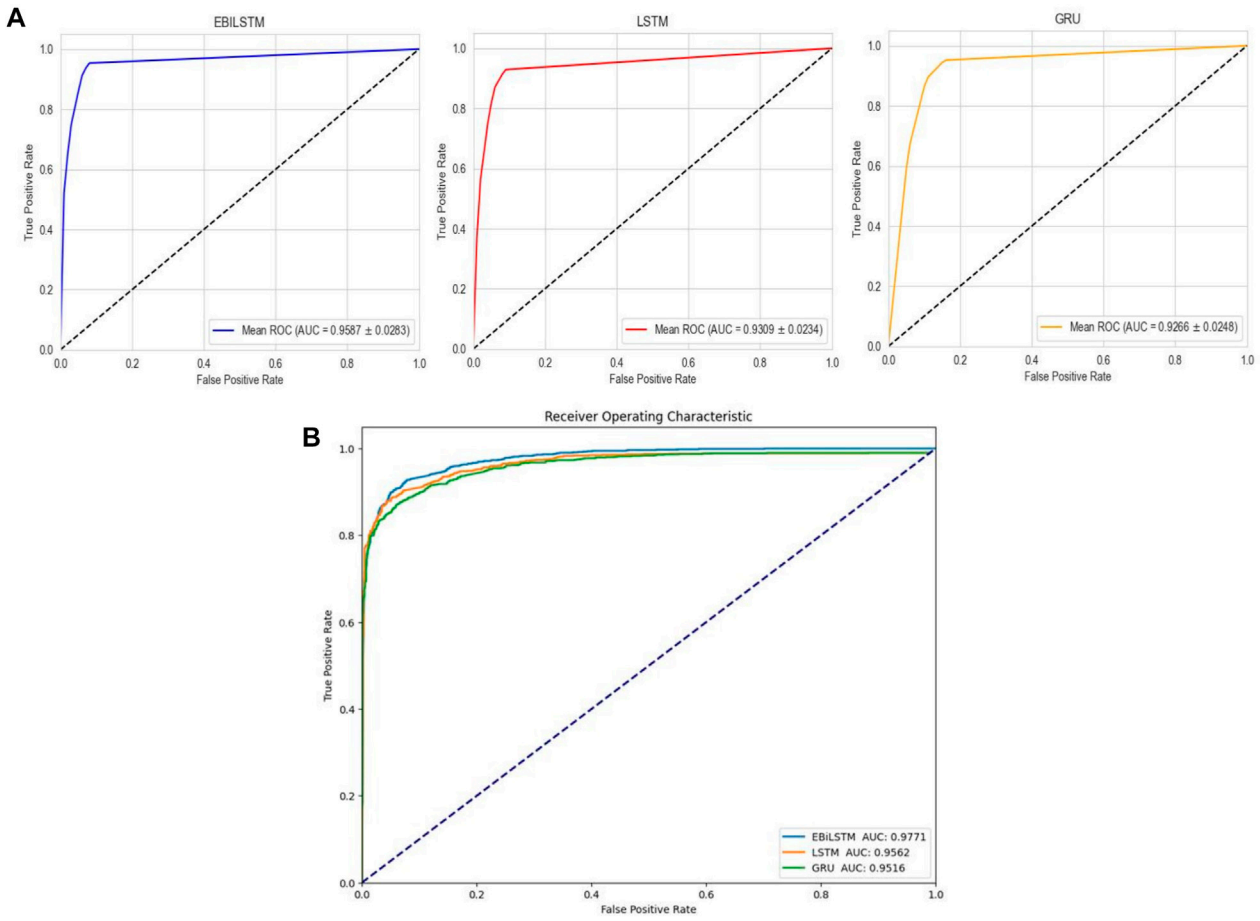


FIGURE 4
Performance of the deep learning models on datasets. **(A)** ROC curves and AUC values of EBiLSTM, LSTM, and GRU after 5-fold cross-validation on the AMP-11053 dataset. **(B)** ROC curves and AUC values of EBiLSTM, LSTM, and GRU on the validation set AMP-2211.

TABLE 2 Results derived from the independent external validation set, AMP-2211.

AMP-2211	Accuracy	Recall	Specificity	Precision	F1-score
EBILSTM	0.9239	0.9186	0.9294	0.9303	0.9244
LSTM	0.9099	0.8971	0.9132	0.9217	0.9189
GRU	0.9018	0.9045	0.8846	0.8907	0.9044

TABLE 3 Performance of the ensemble learning models on AMP-11053.

AMP-11053	Accuracy	Recall	Specificity	Precision	F1-score
AdaBoost	0.8432 ± 0.0095	0.8493 ± 0.0118	0.8366 ± 0.0173	0.8425 ± 0.0145	0.8459 ± 0.0108
LightGBM	0.8912 ± 0.0062	0.9058 ± 0.0089	0.8759 ± 0.0153	0.8826 ± 0.0118	0.8940 ± 0.0061
Xgboost	0.8932 ± 0.0086	0.9033 ± 0.0081	0.8824 ± 0.0123	0.8879 ± 0.0064	0.8955 ± 0.0070

Notes: The above results show the average value of each indicator and the corresponding 95% confidence interval.

3.4 Baseline model results

To comprehensively validate the performance of our models, we used traditional machine learning models, SVM, NBC, and KNN, as benchmarks for comparison with the two categories of models mentioned earlier. On the AMP-11053 dataset, the results from the five-fold cross-validation of the traditional models are shown in Table 5, with the specific AUC curves and ROC values illustrated in

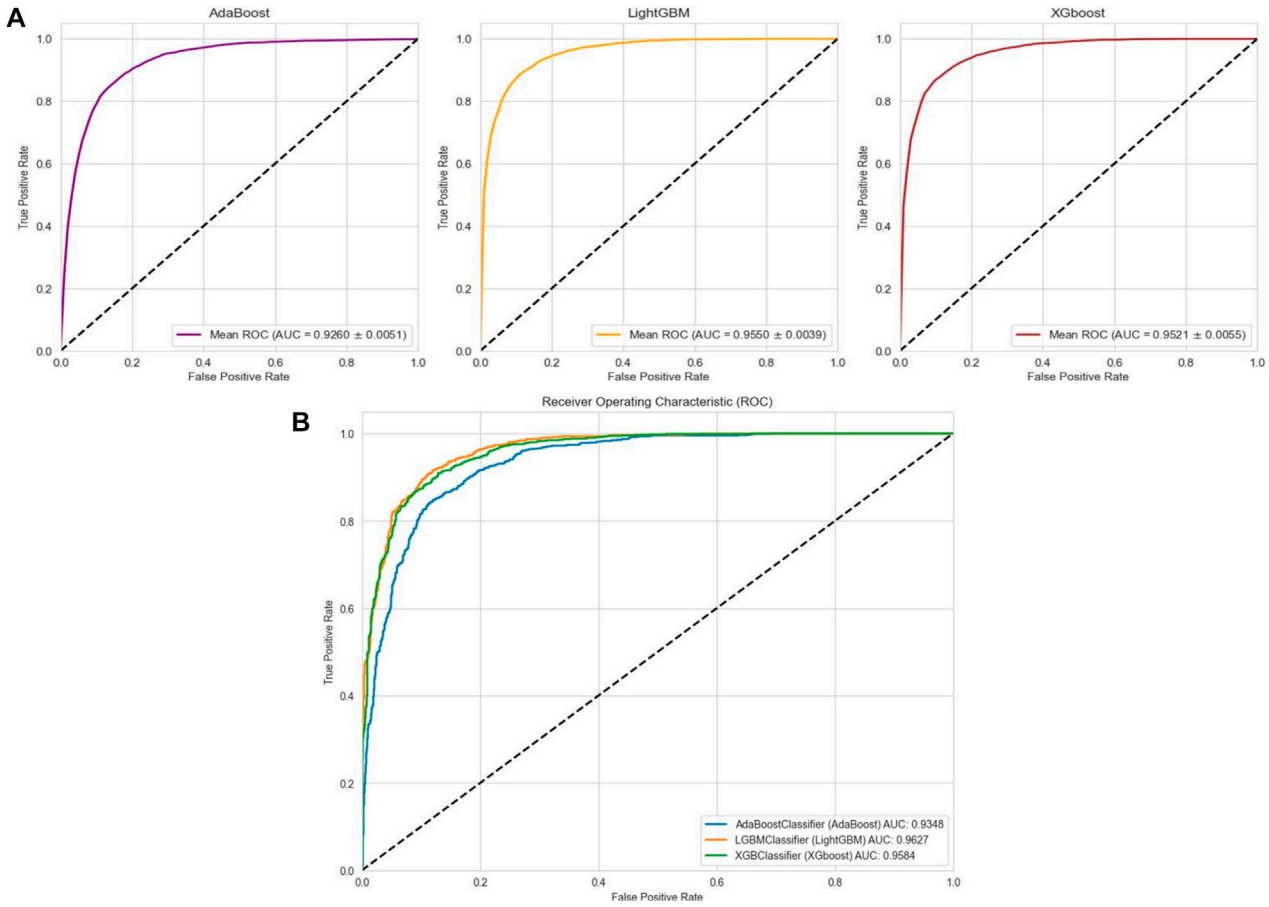


FIGURE 5
Performance of the ensemble learning models on datasets. **(A)** ROC curves and AUC values of Adaboost, LightGBM, and XGBoost after 5-fold cross-validation on the AMP-11053 dataset. **(B)** ROC curves and AUC values of Adaboost, LightGBM, and XGBoost on the validation set AMP-2211.

TABLE 4 Performance of the ensemble learning models on AMP-11053.

AMP-2211	Accuracy	Recall	Specificity	Precision	F1-score
AdaBoost	0.8408	0.8365	0.8452	0.8471	0.8417
LightGBM	0.8996	0.9097	0.8892	0.8938	0.9017
Xgboost	0.9032	0.9133	0.8929	0.8973	0.9052

TABLE 5 Performance of the traditional machine learning models on AMP-11053.

AMP-11053	Accuracy	Recall	Specificity	Precision	F1-score
SVM	0.7663 ± 0.0071	0.7309 ± 0.0057	0.8030 ± 0.0139	0.7919 ± 0.0277	0.7601 ± 0.0140
NBC	0.7167 ± 0.0160	0.5617 ± 0.0173	0.8763 ± 0.0088	0.8234 ± 0.0211	0.6678 ± 0.0169
KNN	0.8749 ± 0.0066	0.8997 ± 0.0068	0.8494 ± 0.0066	0.8597 ± 0.0154	0.8792 ± 0.0097

Figure 6A. We also evaluated the generalization capabilities of each model on an external dataset, AMP-2211. The results of these external validations are listed in Table 6 and depicted in Figure 6B. Among them, K-NN performs the best, but its performance is still not as good as the prediction strategy proposed above.

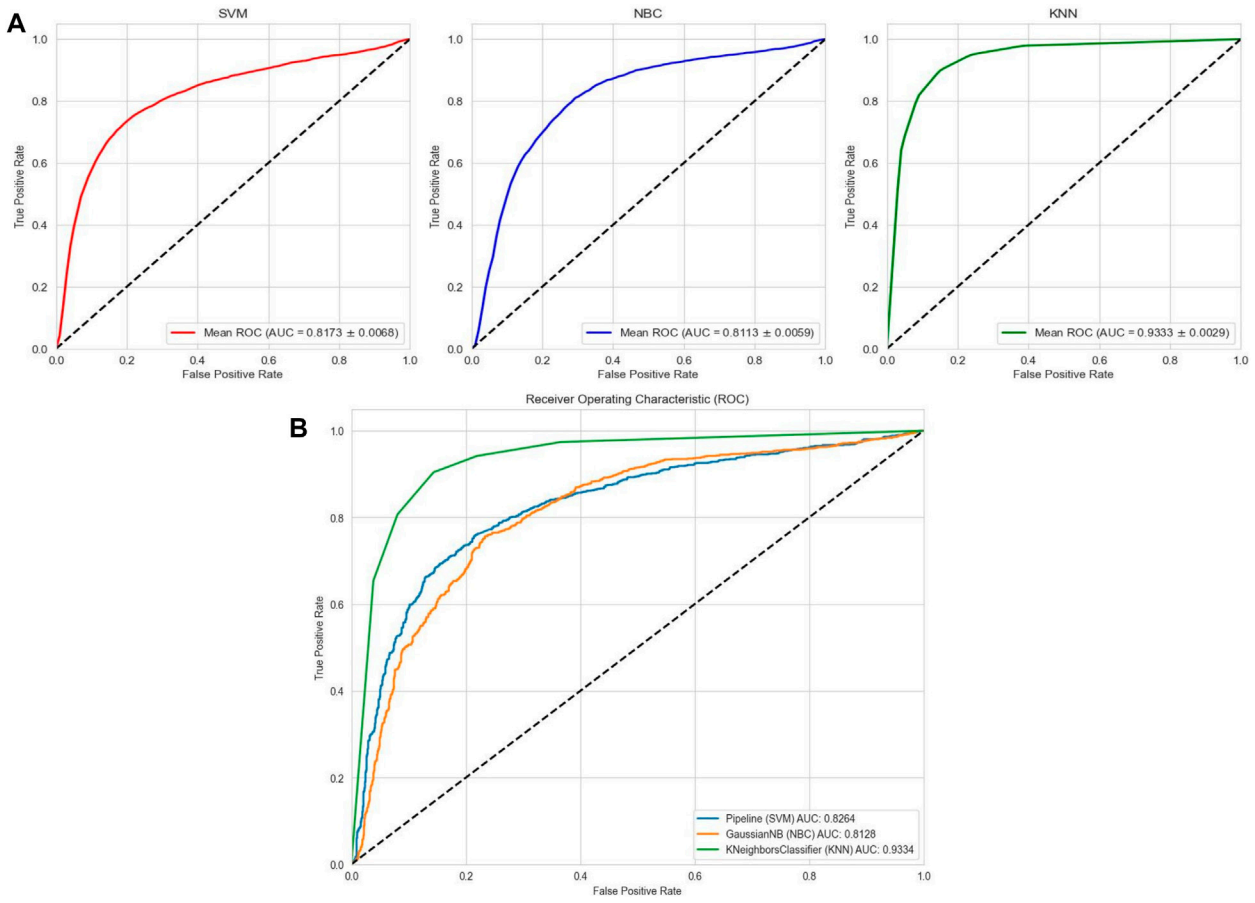


FIGURE 6
Performance of traditional machine learning models on datasets. **(A)** ROC curves and AUC values of SVM, KNN, and NBC after 5-fold cross-validation on the AMP-11053 dataset. **(B)** ROC curves and AUC values of SVM, KNN, and NBC on the validation set AMP-2211.

TABLE 6 Evaluation outcomes from the external standalone validation dataset, AMP-2211.

AMP-2211	Accuracy	Recall	Specificity	Precision	F1-score
SVM	0.7698	0.7623	0.7775	0.7783	0.7702
NBC	0.7092	0.5416	0.881	0.8234	0.6534
KNN	0.8806	0.9044	0.8562	0.8657	0.8846

4 Discussion

In this study, we have focused on finding accurate strategies for antimicrobial peptide screening. With an emphasis on ensemble learning and deep learning methods, we constructed two datasets which were applied for model training, cross-validation, and independent external testing. This ensured rigorous and impartial model evaluation. By comparing various evaluation metrics, we analyzed the performance of the ensemble learning and deep learning models in prediction tasks. The results revealed that our custom-built EBiLSTM model had the highest accuracy, nearly 98% on the test set, demonstrating its significant predictive power in this prediction tasks. In further analyses, our EBiLSTM model was not only effective in peptide screening, but also efficient, significantly reducing the time and resources needed for conventional experimental methods. These results illustrate the potential utility of ensemble learning and deep learning methods in biomolecular studies. The success of the EBiLSTM model underscores the power of these algorithms in handling complex biological data and has promising implications for accelerating antimicrobial peptide discovery. Going forward, we plan to improve this model by integrating additional features and refining hyperparameters to further enhance its predictive capacity. Our ultimate aim is to contribute to effective solutions against antibiotic resistance.

Deep learning outperforms both traditional machine learning and ensemble learning models in terms of accuracy. To understand

why the EBiLSTM model exhibits optimal performance, it is necessary to analyze the network architecture. Firstly, from a design perspective, the model incorporates three BiLSTM layers and four dropout layers. The multi-layer network structure equips the model with sufficient capacity to learn complex patterns in the sequence. An appropriate dropout rate (the optimal value of 0.3, chosen after numerous experiments) plays a key role in preventing over-fitting and enhancing the model's generalization capability. In the final fully connect-ed layer, the network adopts a ReLU activation function. ReLU alleviates the vanishing gradient problem, thereby enhancing the model's learning ability. Concurrently, the number of neurons in each BiLSTM layer is judiciously halved, maintaining sufficient model complexity while avoiding the issue of low computational efficiency. Examining model specifics, most operations in the EBiLSTM are point-wise, such as the activation functions of various gates and the update of cell states. The advantage of these point-wise operations is their high degree of parallelism, enabling the model to effectively utilize the parallel computational capabilities of modern hardware, thus achieving high efficiency in practical applications.

Ensemble learning models and traditional machine learning models have been used extensively in various applications due to their simplicity and interpretability. However, when it comes to predicting AMPs, these models have several limitations. Firstly, they typically operate on a feature-engineering basis, where appropriate features need to be manually extracted from the peptide sequences. This can often be a time-consuming process and may overlook complex patterns or dependencies in the data that could be critical to accurate prediction. Secondly, these models usually treat sequences as fixed-length inputs and lose valuable information when sequences are of variable lengths. This is a significant challenge as peptides can have different lengths, and disregarding this variation can lead to sub-optimal predictions. Finally, these models lack the capacity to automatically learn and improve from data in the same way that deep learning models can. They do not adapt their structure and parameters based on the complexity of the task at hand, which can lead to lower prediction accuracy. In contrast, deep learning models, like our proposed AMP-EBiLSTM, can automatically extract features, accommodate variable-length sequences, and improve over time by learning intricate data patterns. As such, they can often outperform ensemble and traditional machine learning models in complex predictive tasks such as AMP prediction.

Training cost is a key consideration in the application of machine learning and deep learning. Compared to deep learning models, baseline machine learning and ensemble learning models have lower training costs. Ensemble learning and baseline models exhibit low costs in terms of training time, with the training time ranging from 0 to 30 s on our equipment. From the perspective of the number of model parameters, baseline machine learning and ensemble learning models usually have significantly fewer parameters than deep learning models. Secondly, in terms of the training process, baseline machine learning models are typically more concise and efficient. Specifically, SVM is based on the solution of convex optimization problems, NBC is grounded in statistical theory of conditional probability, KNN is based on distance measurement, while Adaboost, LightGBM, and XGboost are implemented through the iterative optimization of a series of weak learners. These processes are typically more efficient than complex training procedures in deep learning, such as backpropagation and gradient descent. Accurate prediction ability and low training cost may provide strong support for the early promotion of antimicrobial peptides to clinical practice. For example, the challenges faced by the application of antimicrobial peptides in clinical diseases such as diabetic foot are high production cost, poor stability, and toxicity problems. Peptides are widely used in clinical departments such as vascular surgery to provide support.

While our study presents promising outcomes, certain limitations need to be acknowledged, and potential avenues for future research should be highlighted. Firstly, despite the comprehensive dataset employed for model training and validation in this study, future research would benefit from the expansion of these datasets. To further ascertain the robustness and generalizability of our approach, it would be beneficial to accumulate more data pertaining to antimicrobial peptides and validate our models on datasets of larger scale and diversity. Secondly, although deep learning models demonstrated superior predictive performance in our study, their substantial training costs pose a challenge. Future efforts should be concentrated on refining these models to lessen training costs whilst sustaining their high predictive accuracy. This might necessitate intensive research and exploration into model architecture, training strategies, and optimization algorithms, among other aspects. Lastly, the current study has primarily focused on the theoretical screening of antimicrobial peptides. An exciting direction for future research would involve integrating our approach with wet lab experiments to provide a more precise validation of the screening results. Such empirical validation could not only further substantiate the effectiveness of our screening strategy but also assist us in comprehending and enhancing our model's predictive out-comes, thereby bolstering the precision and efficiency of antimicrobial peptide screening. In conclusion, these identified avenues for future research will facilitate a deeper understanding and application of machine learning and deep learning in antimicrobial peptide screening. These advancements will undoubtedly contribute to bolstering the research and development of antimicrobial peptides.

5 Conclusion

In this study, we explored the application of deep learning techniques in con-structing models for the identification of antimicrobial peptides, aiming to strike an effective balance between wet lab experimental methods and computational predictions. We proposed a novel deep learning model-EBiLSTM, and conducted meticulous parameter tuning and comprehensive performance evaluations. The results demonstrated that although this model bore a relatively high training cost, it achieved an ac-curacy of 92.39% on the test set, with an AUC value nearing 0.98, showcasing its superior predictive performance. Our study offers fresh perspectives and possibilities for antimicrobial peptide prediction and screening. It showcases the advantages of deep learning and ensemble learning in addressing practical needs and resource conditions with flexibility, providing new research directions and tools for future studies on antimicrobial peptides.

Data availability statement

Publicly available datasets were analyzed in this study. This data can be found here: The antimicrobial peptide data used in this study are all sourced from multiple public databases, including: APD3 (https://aps.unmc.edu/about), PlantPepDB (http://14.139.61.8/PlantPepDB/index.php), BaAMPs (https://www.baamps.it/), Bio-PepDB (https://bis.zju.edu.cn/biopepdbr/), CAMP (https://webs.iiitd.edu.in/raghava/satpdb/catalogs/camp/), DBAASP (https://dbaasp.org/home), DRAMP (https://dramp.cpu-bioinfor.org/), LAMP (https://ngdc.cncb.ac.cn/databasecommons/database/id/4562).

Author contributions

YW: study concept, image preprocessing, experimental design, data analysis, writing of manuscript. LW: experimental design, editing the manuscript, and data collection. CL: data analysis and data collection. YP: data collection and experimental design. XL and YT: study concept and funding. All authors contributed to the article and approved the submitted version.

Funding

This study was funded by Shanxi Province "136 Revitalization Medical Project Construction Funds" and the National Natural Science Foundation of China (32170160).

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.

References

Al-Khdhairawi, A., Sanuri, D., Akbar, R., Lam, S. D., Sugumar, S., Ibrahim, N., et al. (2023). Machine learning and molecular simulation ascertain antimicrobial peptide against *Klebsiella pneumoniae* from public database. *Comput. Biol. Chem.* 102, 107800. doi:10.1016/j.compbiolchem.2022.107800

Axel, C., Brosnan, B., Zannini, E., Peyer, L. C., Furey, A., Coffey, A., et al. (2016). Antifungal activities of three different Lactobacillus species and their production of antifungal carboxylic acids in wheat sourdough. *Appl. Microbiol. Biotechnol.* 100, 1701–1711. doi:10.1007/s00253-015-7051-x

Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognit.* 30 (7), 1145–1159. doi:10.1016/s0031-3203(96)00142-2

Brogden, K. A. (2005). Antimicrobial peptides: Pore formers or metabolic inhibitors in bacteria?[J]. *Nat. Rev. Microbiol.* 3 (3), 238–250. doi:10.1038/nrmicro1098

Chen, T., and Guestrin, C. (2016). "Xgboost: A scalable tree boosting system[C]," in Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, San Francisco, CA, USA, August 13-17, 2016, 785–794.

Chen, W., Ding, H., Feng, P., Lin, H., and Chou, K. C. (2016). iACP: a sequence-based tool for identifying anticancer peptides. *Oncotarget* 7 (13), 16895–16909. doi:10.18632/oncotarget.7815

Cho, K., Van Merriënboer, B., and Gulcehre, C. (2014). *Learning phrase representations using RNN encoder-decoder for statistical machine translation[J]*. arXiv preprint arXiv:1406.1078.

Choi, E., Bahadori, M. T., and Searles, E. (2016). "Multi-layer representation learning for medical concepts[C]," in proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, San Francisco, CA, USA, August 13-17, 2016, 1495–1504.

Chou, K. C. (2001). Prediction of protein cellular attributes using pseudo-amino acid composition. *Proteins Struct. Funct. Bioinforma.* 43 (3), 246–255. doi:10.1002/prot.1035

Costa, F., Carvalho, I. F., Montelaro, R. C., Gomes, P., and Martins, M. C. L. (2011). Covalent immobilization of antimicrobial peptides (AMPs) onto biomaterial surfaces. *Acta biomater.* 7 (4), 1431–1440. doi:10.1016/j.actbio.2010.11.005

Da, S. J., Leal, E. C., and Carvalho, E. (2021). Bioactive antimicrobial peptides as therapeutic agents for infected diabetic foot ulcers. *Bio-molecules.* 11 (12), 1894. doi:10.3390/biom11121894

Freund, Y., and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* 55 (1), 119–139. doi:10.1006/jcss.1997.1504

Fu, H., Cao, Z., Li, M., Zheng, X., and Ge, C. (2020). Effect of simulated microgravity and ionizing radiation on expression profiles of miRNA, lncRNA, and mRNA in human lymphoblastoid cells. *Proc. Fourth Int. Conf. Biol. Inf. Biomed. Eng.* 24, 1–8. doi:10.1016/j.lssr.2019.10.009

Ge, R., Feng, G., Jing, X., Zhang, R., Wang, P., and Wu, Q. (2020). EnACP: An ensemble learning model for identification of anticancer peptides. *Front. Genet.* 11, 760. doi:10.3389/fgene.2020.00760

Hancock, R. E. W., and Diamond, G. (2000). The role of cationic antimicrobial peptides in innate host defences. *Trends Microbiol.* 8 (9), 402–410. doi:10.1016/s0966-842x(00)01823-0

Hancock, R. E. W., Haney, E. F., and Gill, E. E. (2016). The immunology of host defence peptides: Beyond antimicrobial activity. *Nat. Rev. Immunol.* 16 (5), 321–334. doi:10.1038/nri.2016.29

Hancock, R. E. W., and Sahl, H. G. (2006). Antimicrobial and host-defense peptides as new anti-infective therapeutic strategies. *Nat. Biotechnol.* 24 (12), 1551–1557. doi:10.1038/nbt1267

Haney, E. F., Mansour, S. C., and Hancock, R. E. W. (2017). Antimicrobial peptides: An introduction. *Antimicrob. peptides methods Protoc.* 1548, 3–22. doi:10.1007/978-1-4939-6737-7_1

Hasan, M. M., Schaduangrat, N., Basith, S., Lee, G., Shoombuatong, W., and Manavalan, B. (2020). HLPpred-fuse: Improved and robust prediction of hemolytic peptide and its activity by fusing multiple feature representation. *Bioinformatics* 36 (11), 3350–3356. doi:10.1093/bioinformatics/btaa160

Hilchie, A. L., Wuerth, K., and Hancock, R. E. W. (2013). Immune modulation by multifaceted cationic host defense (antimicrobial) peptides. *Nat. Chem. Biol.* 9 (12), 761–768. doi:10.1038/nchembio.1393

Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.* 9 (8), 1735–1780. doi:10.1162/neco.1997.9.8.1735

Huang, J., Xu, Y., and Xue, Y. (2023). Identification of potent antimicrobial peptides via a machine-learning pipeline that mines the entire space of peptide sequences[J]. *Nat. Biomed. Eng.* 2023, 1–14.

Jiang, M., Zhang, R., Xia, Y., Jia, G., Yin, Y., Wang, P., et al. (2022). i2APP: A two-step machine learning framework for antiparasitic peptides identification. *Front. Genet.* 13, 884589. doi:10.3389/fgene.2022.884589

Ke, G., Meng, Q., and Finley, T. (2017). Lightgbm: A highly efficient gradient boosting decision tree[J]. *Adv. neural Inf. Process. Syst.* 30.

Khabbaz, H., Karimi-Jafari, M. H., and Saboury, A. A. (2021). Prediction of antimicrobial peptides toxicity based on their physico-chemical properties using machine learning techniques[J]. *BMC Bioinforma.* 22 (1), 1–11.

Kumar, P., Kizhakkedathu, J. N., and Straus, S. K. (2018). Antimicrobial peptides: Diversity, mechanism of action and strategies to improve the activity and biocompatibility *in vivo*. *Biomolecules* 8 (1), 4. doi:10.3390/biom8010004

Kumar, R., Chaudhary, K., Sharma, M., Nagpal, G., Chauhan, J. S., Singh, S., et al. (2015). Ahtpdb: A comprehensive platform for analysis and presentation of antihypertensive peptides. *Nucleic acids Res.* 43 (D1), D956–D962. doi:10.1093/nar/gku1141

Lázár, V., Martins, A., Spohn, R., Daruka, L., Grézal, G., Fekete, G., et al. (2018). Antibiotic-resistant bacteria show widespread collateral sensitivity to antimicrobial peptides. *Nat. Microbiol.* 3 (6), 718–731. doi:10.1038/s41564-018-0164-0

Mahlapuu, M., Hakansson, J., Ringstad, L., and Bjorn, C. (2016). Antimicrobial peptides: An emerging category of therapeutic agents. *Front. Cell. Infect. Microbiol.* 6, 194. doi:10.3389/fcimb.2016.00194

Manavalan, B., Basith, S., Shin, T. H., Choi, S., Kim, M. O., and Lee, G. (2017). MLACP: Machine-learning-based prediction of anticancer peptides. *Oncotarget* 8 (44), 77121–77136. doi:10.18632/oncotarget.20365

Manavalan, B., Subramaniyam, S., Shin, T. H., Kim, M. O., and Lee, G. (2018). Machine-learning-based prediction of cell-penetrating peptides and their uptake efficiency with improved accuracy. *J. proteome Res.* 17 (8), 2715–2726. doi:10.1021/acs.jproteome.8b00148

Mansour, S. C., Pena, O. M., and Hancock, R. E. W. (2014). Host defense peptides: Front-line immunomodulators. *Trends Immunol.* 35 (9), 443–450. doi:10.1016/j.it.2014.07.004

Meher, P. K., Sahu, T. K., and Saini, V. (2017). Predicting antimicrobial peptides with improved accuracy by incorporating the compositional, physicochemical and structural features into Chou's general PseAAC[J]. *Sci. Rep.* 7 (1), 1–12.

Melo, M. N., Ferre, R., and Castanho, M. A. R. B. (2009). Antimicrobial peptides: Linking partition, activity and high membrane-bound concentrations. *Nat. Rev. Microbiol.* 7 (3), 245–250. doi:10.1038/nrmicro2095

Nakatsuji, T., Chen, T. H., Narala, S., Chun, K. A., Two, A. M., Yun, T., et al. (2017). Antimicrobials from human skin commensal bacteria protect against *Staphylococcus aureus* and are deficient in atopic dermatitis. *Sci. Transl. Med.* 9 (378), 4680. doi:10.1126/scitranslmed.aah4680

Omardien, S., Brul, S., and Zaat, S. A. J. (2016). Antimicrobial activity of cationic antimicrobial peptides against gram-positives: Current progress made in understanding the mode of action and the response of bacteria. *Front. Cell Dev. Biol.* 4, 111. doi:10.3389/fcell.2016.00111

Powers, D. M. W. (2020). *Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation[J]*. arXiv preprint arXiv:2010.16061.

Schuster, M., and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* 45 (11), 2673–2681. doi:10.1109/78.650093

Sokolova, M., and Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Inf. Process. Manag.* 45 (4), 427–437. doi:10.1016/j.ipm.2009.03.002

Tyagi, A., Kapoor, P., and Kumar, R. (2013). *In silico* models for designing and discovering novel anticancer peptides[J]. *Sci. Rep.* 3 (1), 1–8.

Ulmschneider, J. P., and Ulmschneider, M. B. (2018). Molecular dynamics simulations are redefining our view of peptides interacting with biological membranes. *Accounts Chem. Res.* 51 (5), 1106–1116. doi:10.1021/acs.accounts.7b00613

Veltri, D., Kamath, U., and Shehu, A. (2018). Deep learning improves antimicrobial peptide recognition. *Bioinformatics* 34 (16), 2740–2747. doi:10.1093/bioinformatics/bty179

Wang, C., Garlick, S., and Zloh, M. (2021b). Deep learning for novel antimicrobial peptide design. *Biomolecules* 11 (3), 471. doi:10.3390/biom11030471

Wang, H., Zhao, J., and Zhao, H. (2021a). CL-ACP: A parallel combination of CNN and LSTM anticancer peptide recognition model[J]. *BMC Bioinforma.* 22, 1–22.

Wani, M. A., Garg, P., and Roy, K. K. (2021). Machine learning-enabled predictive modeling to precisely identify the antimicrobial peptides. *Med. Biol. Eng. Comput.* 59 (11-12), 2397–2408. doi:10.1007/s11517-021-02443-6

Wimley, W. C. (2010). Describing the mechanism of antimicrobial peptide action with the interfacial activity model. *ACS Chem. Biol.* 5 (10), 905–917. doi:10.1021/cb1001558

Xing, C., Zhu, H., Dou, X., Gao, L., Baddi, S., Zou, Y., et al. (2023). Infected diabetic wound regeneration using peptide-modified chiral dressing to target Re-vascularization. *ACS Nano* 17 (7), 6275–6291. doi:10.1021/acsnano.2c10039

Yoon, B. K., Jackman, J. A., Valle-Gonzalez, E. R., and Cho, N. J. (2018). Antibacterial free fatty acids and monoglycerides: Biological activities, experimental testing, and therapeutic applications. *Int. J. Mol. Sci.* 19 (4), 1114. doi:10.3390/ijms19041114

Zasloff, M. (2002). Antimicrobial peptides of multicellular organisms. *nature* 415 (6870), 389–395. doi:10.1038/415389a

frontiers | Frontiers in Genetics

TYPE Original Research
PUBLISHED 27 July 2023
DOI 10.3389/fgene.2023.1226905

OPEN ACCESS

EDITED BY
Lei Chen,
Shanghai Maritime University, China

REVIEWED BY
Hao Lin,
University of Electronic Science and Technology of China, China
Wang-Ren Qiu,
Jingdezhen Ceramic Institute, China

*CORRESPONDENCE
Cangzhi Jia,
cangzhijia@dlmu.edu.cn
Zhengkui Lin,
dalianjx@163.com

RECEIVED 22 May 2023
ACCEPTED 30 June 2023
PUBLISHED 27 July 2023

CITATION
Liu D, Lin Z and Jia C (2023), NeuroCNN_GNB: an ensemble model to predict neuropeptides based on a convolution neural network and Gaussian naive Bayes. *Front. Genet.* 14:1226905. doi: 10.3389/fgene.2023.1226905

NeuroCNN_GNB: an ensemble model to predict neuropeptides based on a convolution neural network and Gaussian naive Bayes

Di Liu[1], Zhengkui Lin[1]* and Cangzhi Jia[2]*

[1]Information Science and Technology College, Dalian Maritime University, Dalian, China, [2]School of Science, Dalian Maritime University, Dalian, China

Neuropeptides contain more chemical information than other classical neurotransmitters and have multiple receptor recognition sites. These characteristics allow neuropeptides to have a correspondingly higher selectivity for nerve receptors and fewer side effects. Traditional experimental methods, such as mass spectrometry and liquid chromatography technology, still need the support of a complete neuropeptide precursor database and the basic characteristics of neuropeptides. Incomplete neuropeptide precursor and information databases will lead to false-positives or reduce the sensitivity of recognition. In recent years, studies have proven that machine learning methods can rapidly and effectively predict neuropeptides. In this work, we have made a systematic attempt to create an ensemble tool based on four convolution neural network models. These baseline models were separately trained on one-hot encoding, AAIndex, G-gap dipeptide encoding and word2vec and integrated using Gaussian Naive Bayes (NB) to construct our predictor designated NeuroCNN_GNB. Both 5-fold cross-validation tests using benchmark datasets and independent tests showed that NeuroCNN_GNB outperformed other state-of-the-art methods. Furthermore, this novel framework provides essential interpretations that aid the understanding of model success by leveraging the powerful Shapley Additive exPlanation (SHAP) algorithm, thereby highlighting the most important features relevant for predicting neuropeptides.

KEYWORDS

neuropeptides, word2vec, one-hot, stacking strategy, convolution neural network

Introduction

Neuropeptides are bioactive peptides that mainly exist in neurons and play a role in information transmission (Svensson et al., 2003). They are ubiquitous not only in the nervous system but also in various systems of the body, with a low content, high activity, and extensive and complex functions (Hökfelt et al., 2000). According to the specific type, they play role as transmitters, modulators, and hormones. Neuropeptides share the common characteristic that they are produced from a longer neuropeptide precursor (NPP) (Kang et al., 2019). Generally, an NPP contains a signal peptide sequence, one or more neuropeptide sequences and some other sequences that are often homologous among neuropeptides. After the NPP enters the rough endoplasmic reticulum (Rer), the signal peptide is quickly cleaved by signal peptidase and converted into a prohormone, which is

transferred to the Golgi complex for proteolysis and posttranslational processing, which ultimately results in a mature neuropeptide. The neuropeptides modified by various physiological processes are transported to the terminal, stored in larger vesicles and released, and their length ranges from 3 to 100 amino acid residues (Salio et al., 2006; Wang et al., 2015). At present, there is much evidence indicating that neuropeptides play a particularly important role in the regulation of nervous system adaptation to pressure, pain, injury and other stimuli. These characteristics indicate that neuropeptides may represent a new direction in the treatment of nervous system diseases. A popular experimental method for the identification of neuropeptides is LC–MS, whose accuracy has been greatly reduced because it has certain requirements for the quantity and quality of peptides to be extracted (Van Eeckhaut et al., 2011; Van Wanseele et al., 2016).

With the development of high-throughput next-generation sequencing technology and expressed sequence tag databases, machine learning methods have been applied to rapidly and effectively predict neuropeptide peptides. NeuroPID, NeuroPred and NeuroPP are the earliest computational tools for identifying neuropeptide precursors (Southey et al., 2006; Ofer and Linial, 2014; Kang et al., 2019). NeuroPIpred was the first predictor designed for identifying insect neuropeptides based on amino acid composition, dipeptide composition, split composition, binary profile feature extraction and the support vector machine (SVM) classification algorithm (Agrawal et al., 2019). PredNeuroP was designed by building a two-layer stacking model that was trained on nine kinds of hybrid features for animal phyla neuropeptide prediction (Bin et al., 2020). In PredNeuroP, extremely randomized trees (ERT), artificial neural network (ANN), k-nearest neighbor (KNN), logistic regression (LR), and extreme gradient boosting (XGBoost) were employed to develop ML-based models. In terms of feature coding, PredNeuroP uses amino acid composition, dipeptide composition, binary profile-based features, amino acid index features, grouped amino acid composition, grouped dipeptide composition, composition-transition-distribution, and amino acid entropy. In 2021, Hasan *et al.* developed a meta-predictor NeuroPred-FRL on the basis of 11 different encodings and six different classifiers (Hasan et al., 2021). Although the existing models have achieved relatively satisfactory prediction performances, most of them are developed based on traditional machine learning methods, and deep learning predictors have not been fully explored.

In this work, we have made a systematic attempt to create a tool that can predict neuropeptides using a stacking strategy based on four convolution neural network models. These base models were separately trained on one-hot encoding, AAIndex, G-gap dipeptide encoding and word2vec. By comparing five integration strategies, including LR (Perlman et al., 2011), AdaBoost (Freund and Schapire, 1997), GBDT (Lei and Fang, 2019), Gaussian NB and XGBoost, on 5-fold cross-validation tests, we finally selected Gaussian NB to construct our predictor designated NeuroCNN_GNB, with an AUC of 0.963, Acc of 90.77%, Sn of 89.86% and Sp of 91.69% on 5-fold cross-validation test. Moreover, to enhance the interpretability of the 'black-box' machine learning approach used by NeuroCNN_GNB, we employed the Shapley Additive exPlanation (SHAP) method (Lundberg and Lee, 2017) to highlight the most important and contributing features allowing NeuroCNN_GNB to generate the prediction outcomes. The analysis results showed that one-hot encoding and word2vec play key roles in the identification of neuropeptides.

Materials and methods

Overall framework

The construction process of NeuroCNN_GNB is shown in Figure 1. First, we collected the training dataset and the independent test dataset from original work (Bin et al., 2020). Then, we extracted four types of sequence information from different aspects and combined them with convolutional neural networks to construct base classifiers. In the third step, we considered different stacking strategies to build the final optimal model. Next, we evaluated the performance of the model on the training and independent test datasets and compared it with that of other state-of-the art methods. In the final step, the NeuroCNN_GNB webserver and the corresponding source code were developed and publicly released.

Data collection

Building the benchmark datasets is one of the most important and critical steps in building a prediction algorithm. In this work, we applied the dataset that was first constructed by (Bin et al., 2020) and subsequently used by (Hasan et al., 2021; Jiang et al., 2021). This dataset contains 2425 neuropeptides collected from (Wang et al., 2015) and 2425 nonneuropeptides collected from Swiss-Prot (UniProt Consortium, 2021). It should be noted that the samples in this dataset were processed in two steps. The first step was to remove those protein sequences that contained less than 5 and more than 100 amino acids, as neuropeptides are small peptides generally containing less than 100 amino acids (Salio et al., 2006; Wang et al., 2015). The second step was to remove the protein sequences with a high similarity. Using the threshold of 0.9, CD-HIT was applied to delete redundant samples inside positive and negative samples, and CD-HIT-2D was applied to delete redundant samples between positive and negative samples (Huang et al., 2010). To optimize and compare the predictor, the dataset was further divided into training and independent test datasets according to the proportion of 8:2.

Feature extraction

In this study, we use four different encoding schemes to obtain information on neuropeptides and nonneuropeptides, including one-hot encoding, physicochemical-based features, amino-acid frequency-based features, and embedding methods. These encoding schemes consider 20 types of natural amino acid residues ('A', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'K', 'L', 'M', 'N', '*p*', 'Q',

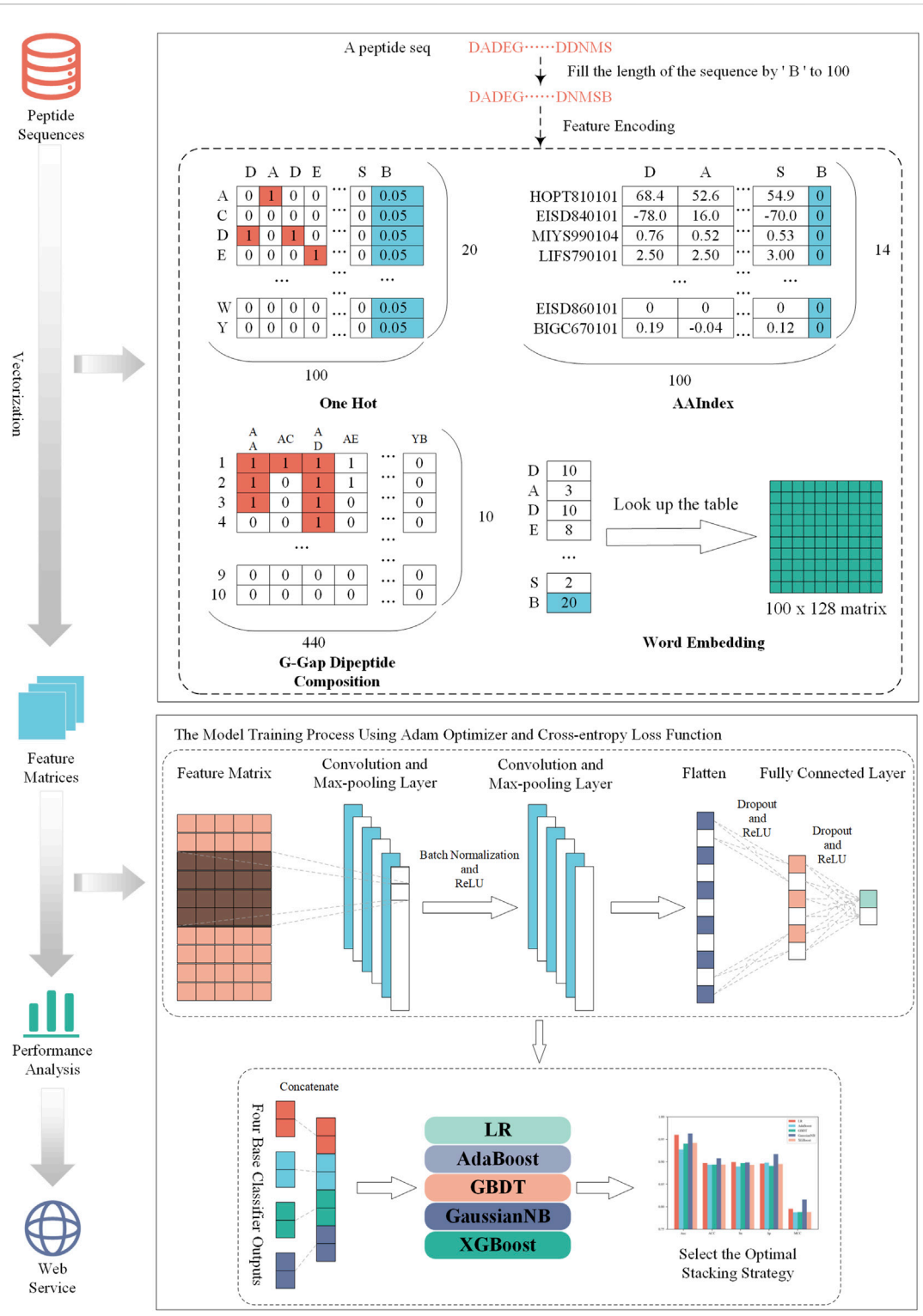


FIGURE 1
The developmental flowchart of NeuroCNN_GNB.

'R', 'S', 'T', 'V', 'W','Y') and add a pseudo character ('B') to obtain the characteristics with the same dimension. Specifically, we fixed the sequence length to 100 and filled the gaps with 'B' if the protein sequence length was less than 100. The details of the feature encodings are described in the following sections.

One-hot encoding

One-hot encoding can reflect the specific amino acid position of a given protein sequence. Each amino acid residue was transformed into a binary vector as follows:

$$\begin{cases} A = (1, 0, 0, \ldots, 0, 0) \\ C = (0, 1, 0, \ldots, 0, 0) \\ \ldots \\ \ldots \\ W = (0, 0, 0, \ldots, 1, 0) \\ V = (0, 0, 0, \ldots, 0, 1) \\ B = (0.05, 0.05, 0.05, \ldots, 0.05, 0.05) \end{cases} \quad (1)$$

The reason that we set each element of B as 0.05 is that we assumed the average frequency of each amino acid is uniformly distributed as the work (Pan et al., 2018; Pan and Shen, 2018; Yang et al., 2021). Thus, one-hot encoding generates a 100 × 20-D feature matrix for a given peptide sequence with a length of 100.

Amino acid index (AAIndex)

AAIndex is a database that includes 566 various physicochemical and biochemical properties of amino acids and amino acid pairs (Kawashima et al., 2007). In this section, we chose 14 properties because they have been verified to be very effective in improving the prediction performance of neuropeptide recognition (Bin et al., 2020; Khatun et al., 2020). Their accession numbers are HOPT810101, EISD840101, MIYS990104, LIFS790101, MAXF760101, CEDJ970104, GRAR740102, KYTJ820101, MITS020101, DAWD720101, BIOV880101, CHAM810101, EISD860101, and BIGC670101. For each physicochemical property, each amino acid was assigned a numerical index, and their values are listed in Supplementary Table S1.

G-gap dipeptide encoding

The G-gap dipeptide encoding scheme incorporates the amino acid frequency information of the peptide sequence, where g' is a parameter that represents a dipeptide with a gap of 'g' amino acids (A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y, B) (Lin et al., 2013; Lin et al., 2015; Xu et al., 2018). In this study, we tried 0, 1, 2,3, and 4-gap dipeptides to encode each protein peptide. For the 21 amino acids (20 natural amino acids and a temporary amino acid B′), there were 441 dipeptide combinations. We discarded the combination BB' and reserved 440 amino acid pairs to effectively capture the component information in protein peptides. Based on the statistical analysis, the highest number of amino acid pairs in the existing training dataset was 10. Therefore, the number of amino acid pairs was encoded into one-hot encoding of 10 dimensions. Finally, we could generate a characteristic matrix of 440*10 for a given peptide sequence.

Word embedding

Word embedding is a strategy to convert words in text into digital vectors for analysis using standard machine learning algorithms (Mikolov et al., 2013). This strategy has been extensively applied in natural language processing and has been introduced to the fields of proteomics and genomics (Lilleberg et al., 2015; Ng, 2017; Jatnika et al., 2019; Wu et al., 2019). Word2vec is an efficient method to create word embedding that includes two algorithms, namely, skip Gram and CBOW (continuous bag-of-words). The difference

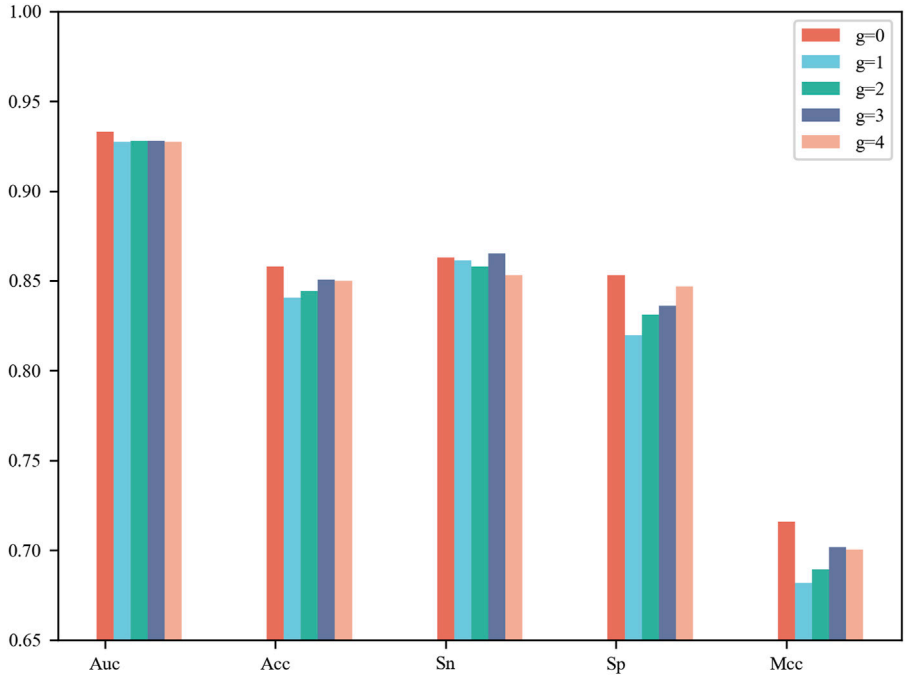


FIGURE 2
Performance comparison of g-Gap Model on 5-fold cross-validation test.

TABLE 1 The Performance of base classifiers on 5-fold cross validation.

Feature	AUC	Acc	Sn	Sp	MCC
One-Hot	0.956	0.887	0.891	0.883	0.775
AAIndex	0.954	0.885	0.872	0.899	0.771
G-Gap	0.933	0.858	0.863	0.853	0.716
Word2vec	0.952	0.882	0.867	0.898	0.765

TABLE 2 Results of 5-fold and 10-fold cross-validation on base classifiers.

Cross-validation	Encoding	AUC	Acc	Sn	Sp	MCC
5-fold	one-hot	**0.956**	**0.887**	**0.891**	0.883	**0.775**
10-fold	one-hot	0.952	0.882	0.879	**0.885**	0.765
5-fold	AAIndex	**0.954**	**0.885**	**0.872**	**0.899**	**0.771**
10-fold	AAIndex	0.948	0.877	0.868	0.885	0.755
5-fold	word2vec	**0.952**	**0.882**	**0.867**	**0.898**	**0.765**
10-fold	word2vec	0.942	0.871	0.865	0.875	0.741

The bold values indicate the higher values of the 5-fold and the 10-fold cross validation results.

between them is that skip Gram predicts the words around the head word through the central word, while CBOW predicts the central word through the surrounding words. According to the preliminary experimental performance, we selected skip Gram to encode each protein peptide in the subsequent experiments.

Model framework

To capture the information contained in multiple feature scenarios, we used a stacking strategy to develop our model to efficiently identify neuropeptides. Stacking is an ensemble learning method that combines predicted information from multiple models to generate a more stable model (Ganaie et al., 2022). The stacking method has two main steps, in which we used the so-called base classifier and meta-classifier. In our work, four base classifiers were constructed based on convolutional neural networks (CNNs). For each type of feature, the corresponding CNN model was trained using grid search to optimize the hyperparameters. All training processes are conducted through the Python package 'pytorch'.

Performance evaluation

To objectively evaluate and compare the predictive performance of the models, five frequently used performance metrics were used, including sensitivity (Sn), specificity (Sp), accuracy (Acc), and MCC. Their formulas are given as follows:

$$Sn = \frac{TP}{TP + FN} \tag{2}$$

$$Sp = \frac{TN}{TN + FP} \tag{3}$$

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \tag{5}$$

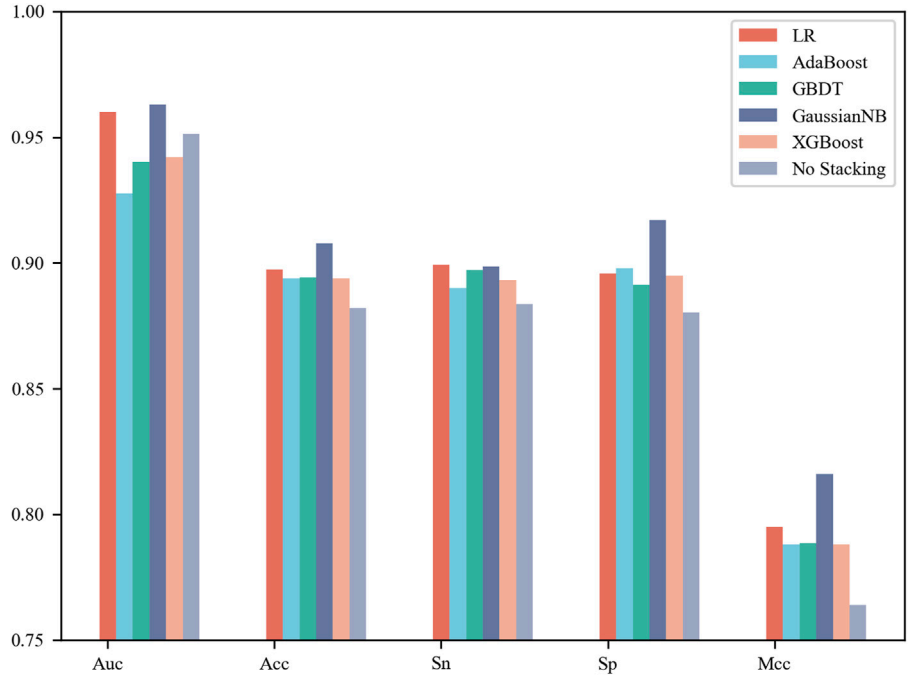


FIGURE 3
Performance comparison of the different stacking algorithms on 5-fold cross-validation test.

TABLE 3 Comparing with other exiting methods on the independent test dataset.

Method	AUC	Acc	Sn	Sp	MCC
NeuroPred-FRL	0.960	0.916	**0.929**	0.903	0.834
NeuroPpred-Fuse	0.958	0.906	0.882	**0.930**	0.813
PredNeuroP	0.954	0.897	0.886	0.907	0.794
Our model	**0.962**	**0.918**	0.919	0.917	**0.836**

where TP, TN, FP and FN denote the numbers of true positives, true negatives, false-positives and false-negatives, respectively. Furthermore, we used the area under the ROC curve (AUC) as one of the main metrics to evaluate model performance.

Results and discussion

Performance analysis of base classifiers

CNN contains a number of tunable hyperparameters, which can affect the validity and robustness of the model. We used a grid search to tune the hyperparameters and explore their optimal combination using 5-fold cross-validation. The average AUCs were designed as the criterion for selecting the parameter combinations. For the G-gap-model (g = 0, 1, 2, 3, 4), we compared their performance on 5-fold cross-validation and show their results in Figure 2. The model based on g = 0 reached the best AUC of 0.933, Acc of 0.858, Sp of 0.853 and MCC of 0.716, while the model based on g = 3 achieved the best Sn of 0.865. Upon comprehensive consideration, an appropriate selection of g = 0 was adopted to build one of the base classifiers. The details of the G-gap based model are summarized in Supplementary Table S3.

Supplementary Table S2 summarizes the optimal combination of parameters for each base classifier, and Table 1 lists their 5-fold cross-validation results. It was observed that the one-hot-based model achieved the best AUC of 0.956, which was slightly superior to the AAIndex and word2vec models. In total, the AUC values of the four base classifiers were greater than 0.93, showing satisfactory prediction results.

In addition, we also performed 10-fold cross-validation test to evaluate the generalization ability of our model. As shown in Table 2, there is almost no difference in the prediction results between 5-fold and 10-fold cross-validation results. Specifically, the AUC of 10-fold cross-validation results based on one-hot is 0.004 lower, based on AAIndex is 0.006 lower, based on word2vec is 0.01 lower than that of 5-fold, respectively.

Stacking models providing robust and reliable prediction results

In this section, each base classifier was considered a weak classifier and then integrated into a strong classifier. LR, AdaBoost, GBDT, Gaussian NB and XGBoost were used as stacking algorithms to construct the meta model. The specific process is that we concatenate

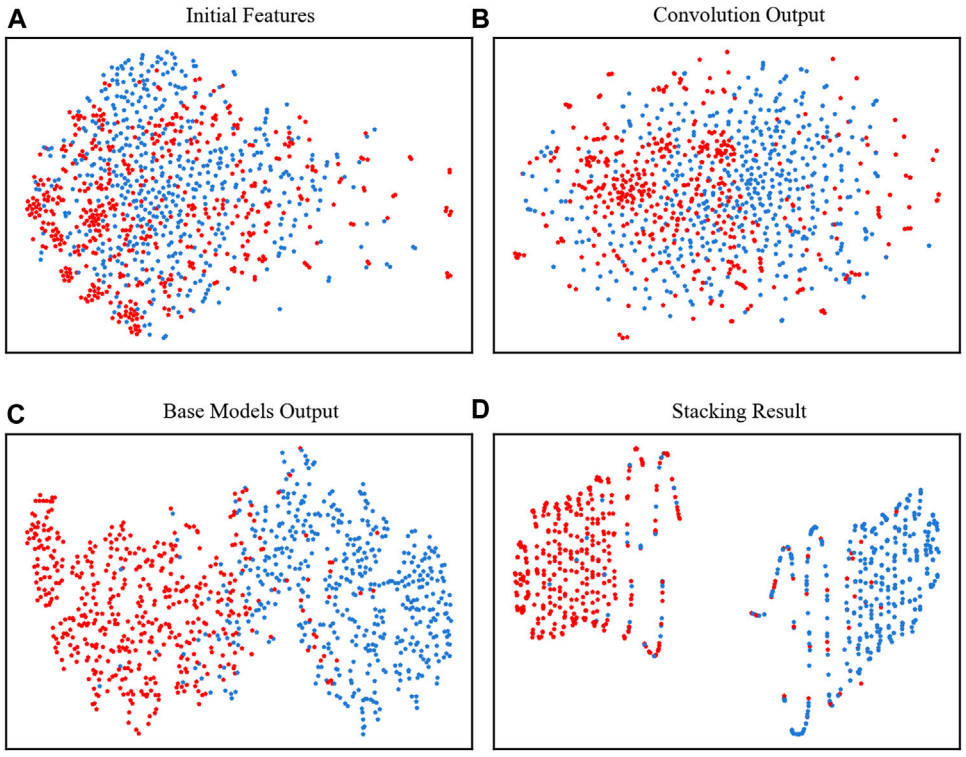


FIGURE 4
t-SNE plots of the positive and negative samples. **(A)** The initial features, **(B)** the features extracted by convolutional layer, **(C)** outputs of the four base classifiers and **(D)** the final output of the model.

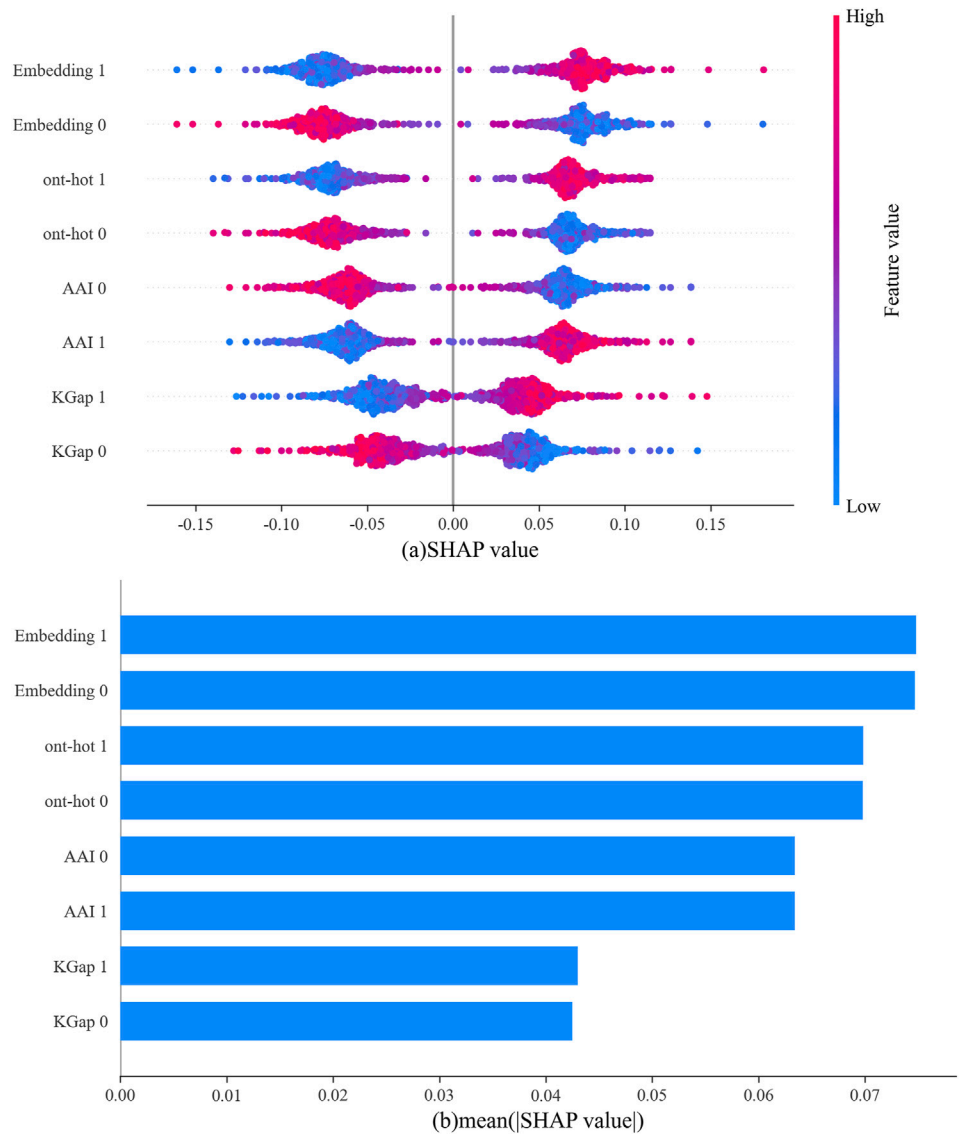


FIGURE 5
The outputs from four base classifiers according to SHAP values for the neuropeptides, **(A)** SHAP value for each sample; **(B)** the average of the absolute values of SHAP for all samples.

the prediction results of four base classifiers for the same sample as the input to the stacking algorithm to obtain the final classification label (Rokach, 2010; Lalmuanawma et al., 2020; Aishwarya and Ravi Kumar, 2021; Ganaie et al., 2022). It can be observed from Figure 3 that Gaussian NB achieved the best performance with an AUC of 0.963, Acc of 90.77%, Sn of 89.86% and Sp of 91.69% on the 5-fold cross-validation test. Moreover, this set of results achieved by the stacking strategy was better than those obtained by the four base classifiers. However, not all integration results were superior to a single model. The stacking results of AdaBoost were inferior to those of the four base classifiers, whose AUC was only 0.928. Taken together, the results showed that selection of a stacking strategy is necessary for different biological sequences. How to find the relationship between the data distribution and classification algorithm is a problem worth studying in the future.

Performance comparison with existing methods on the independent test datasets

We then used the independent test dataset to verify the robustness of NeuroCNN_GNB and compared the prediction results with those of NeuroPpred-Fuse, NeuroPred-FRL and PredNeuroP. These predictors were developed based on the same training dataset as our model, which guarantees the fairness and objectivity of the independent test. The comparison results in Table 3 show that our model obtained the best AUC of 0.962, Acc of 0.918 and MCC of 0.836, which implied a similar effect of predicting positive and negative samples. NeuroPred-FRL achieved the second best AUC of 0.960 and the best Sn of 0.929, and NeuroPred-Fuse showed the best Sp of 0.930. Thus, each of the three models has its own advantages in prediction performance based on four types of features and four base classifiers, whose complexity was

lower than that of the other four models. In particular, this work not only establishes an efficient prediction model but also provides a freely convenient web server for researchers.

Visualization of features

To clearly show how the model performs at each stage, we used t-SNE to visually observe the classification results of the two types of data (Van der Maaten and Hinton, 2008). In Figure 4A, the points were mixed in disorder by using the initial features to concatenate all 4 kinds of encodings, which were almost impossible to divide. However, after the four base classifiers, the neuropeptides and nonneuropeptides were almost separated except for the middle part, which occasionally overlaps, as shown in Figures 4B, C. Finally, after the stacking strategy, our model clearly identified the neuropeptides and nonneuropeptides, as shown in Figure 4D. This figure shows that our model can effectively acquire the intrinsic information of the neuropeptides.

Model interpretation: the effect of feature encoding on model prediction

In this study, four different feature-encoding schemes were used to generate the feature vectors. The performance of each type of feature is listed in Table 1. To display the influence of various features on the model more intuitively, the SHAP (SHapley Additive exPplanation) algorithm was applied to evaluate feature behavior in our datasets (Lundberg and Lee, 2017).

In Figure 5A, the abscissa represents the SHAP value, the ordinate represents each type of feature for the positive sample (abbreviated as 1) and negative sample (abbreviated as 0), and each point is the SHAP value of an instance. Redder sample points indicate that the value of the feature is larger, and bluer sample points indicate that the value of the feature is smaller. If the SHAP value is positive, this indicates that the feature drives the predictions toward neuropeptides and has a positive effect; if negative, the feature drives the predictions toward nonneuropeptides and has a negative effect. For a more intuitive display, the average absolute values for each type of feature are shown in Figure 5B. It can be clearly observed that among the output of the four base classifiers, the one-hot and word embedding-based models were the primary contributors to the final output of the model.

Conclusion

In this study, we introduced a robust predictor based on a stacking strategy to accurately predict neuropeptides. The predictor extracted four types of protein sequence information, employed CNN to train base classifiers, and then selected Gaussian NB to build an ensemble model. The validity of our model was assessed using 5-fold cross-validation and an independent test dataset. In addition, t-SNE was used to visually observe the clustering of samples at each stage, and SHAP was also used to interpret what role each type of feature plays in the classification process. A user-friendly webserver and the source code for our model are freely available at http://47.92.65.100/neuropeptide/. Our model showed satisfactory results when evaluated from different aspects, but there is still room for optimization of the model as a predictor with the increase in experimental neuropeptide data.

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.

Author contributions

DL and CJ designed the study. DL and ZL carried out all data collection and drafted the manuscript. CJ and ZL revised 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

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fgene.2023.1226905/full#supplementary-material

References

Agrawal, P., Kumar, S., Singh, A., Raghava, G. P. S., and Singh, I. K. (2019). NeuroPIpred: A tool to predict, design and scan insect neuropeptides. *Sci. Rep.* 9, 5129. doi:10.1038/s41598-019-41538-x

Aishwarya, T., and Ravi Kumar, V. (2021). Machine learning and deep learning approaches to analyze and detect COVID-19: A review. *SN Comput. Sci.* 2, 226. doi:10.1007/s42979-021-00605-9

Bin, Y., Zhang, W., Tang, W., Dai, R., Li, M., Zhu, Q., et al. (2020). Prediction of neuropeptides from sequence information using ensemble classifier and hybrid features. *J. proteome Res.* 19, 3732–3740. doi:10.1021/acs.jproteome.0c00276

Freund, Y., and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* 55, 119–139. doi:10.1006/jcss.1997.1504

Ganaie, M. A., Hu, M., Malik, A., Tanveer, M., and Suganthan, P. (2022). Ensemble deep learning: A review. *Eng. Appl. Artif. Intell.* 115, 105151. doi:10.1016/j.engappai.2022.105151

Hasan, M. M., Alam, M. A., Shoombuatong, W., Deng, H. W., Manavalan, B., and Kurata, H. (2021). NeuroPred-FRL: An interpretable prediction model for identifying neuropeptide using feature representation learning. *Briefings Bioinforma.* 22, bbab167. doi:10.1093/bib/bbab167

Hökfelt, T., Broberger, C., Xu, Z-Q. D., Sergeyev, V., Ubink, R., and Diez, M. (2000). Neuropeptides—An overview. *Neuropharmacology* 39, 1337–1356. doi:10.1016/s0028-3908(00)00010-1

Huang, Y., Niu, B., Gao, Y., Fu, L., and Li, W. (2010). CD-HIT suite: A web server for clustering and comparing biological sequences. *Bioinformatics* 26, 680–682. doi:10.1093/bioinformatics/btq003

Jatnika, D., Bijaksana, M. A., and Suryani, A. A. (2019). Word2vec model analysis for semantic similarities in English words. *Procedia Comput. Sci.* 157, 160–167. doi:10.1016/j.procs.2019.08.153

Jiang, M., Zhao, B., Luo, S., Wang, Q., Chu, Y., Chen, T., et al. (2021). NeuroPpred-fuse: An interpretable stacking model for prediction of neuropeptides by fusing sequence information and feature selection methods. *Briefings Bioinforma.* 22, bbab310. doi:10.1093/bib/bbab310

Kang, J., Fang, Y., Yao, P., Tang, Q., and Huang, J. (2019). NeuroPP: A tool for the prediction of neuropeptide precursors based on optimal sequence composition. *Interdiscip. Sci. Comput. Life Sci.* 11, 108–114. doi:10.1007/s12539-018-0287-2

Kawashima, S., Pokarowski, P., Pokarowska, M., Kolinski, A., Katayama, T., and Kanehisa, M. (2007). AAindex: Amino acid index database, progress report 2008. *Nucleic acids Res.* 36, D202–D205. doi:10.1093/nar/gkm998

Khatun, M. S., Hasan, M. M., Shoombuatong, W., and Kurata, H. (2020). ProIn-fuse: Improved and robust prediction of proinflammatory peptides by fusing of multiple feature representations. *J. Computer-Aided Mol. Des.* 34, 1229–1236. doi:10.1007/s10822-020-00343-9

Lalmuanawma, S., Hussain, J., and Chhakchhuak, L. (2020). Applications of machine learning and artificial intelligence for covid-19 (SARS-CoV-2) pandemic: A review. *Solit. Fractals* 139, 110059. doi:10.1016/j.chaos.2020.110059

Lei, X., and Fang, Z. (2019). Gbdtcda: Predicting circRNA-disease associations based on gradient boosting decision tree with multiple biological data fusion. *Int. J. Biol. Sci.* 15, 2911–2924. doi:10.7150/ijbs.33806

Lilleberg, J., Zhu, Y., and Zhang, Y. (2015). "Support vector machines and word2vec for text classification with semantic features," in 2015 IEEE 14th International Conference on Cognitive Informatics and Cognitive Computing (ICCI* CC), Beijing, China, 06-08 July 2015 (IEEE), 136–140.

Lin, H., Chen, W., and Ding, H. (2013). AcalPred: A sequence-based tool for discriminating between acidic and alkaline enzymes. *PloS one* 8, e75726. doi:10.1371/journal.pone.0075726

Lin, H., Liu, W-X., He, J., Liu, X. H., Ding, H., and Chen, W. (2015). Predicting cancerlectins by the optimal g-gap dipeptides. *Sci. Rep.* 5, 16964–16969. doi:10.1038/srep16964

Lundberg, S. M., and Lee, S-I. (2017). A unified approach to interpreting model predictions. *Adv. neural Inf. Process. Syst.* 30. doi:10.48550/arXiv.1705.07874

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

Ng, P. (2017). *dna2vec: Consistent vector representations of variable-length k-mers*. arXiv preprint arXiv:1701.06279.

Ofer, D., and Linial, M. (2014). NeuroPID: A predictor for identifying neuropeptide precursors from metazoan proteomes. *Bioinformatics* 30, 931–940. doi:10.1093/bioinformatics/btt725

Pan, X., Rijnbeek, P., Yan, J., and Shen, H. B. (2018). Prediction of RNA-protein sequence and structure binding preferences using deep convolutional and recurrent neural networks. *BMC genomics* 19, 511–11. doi:10.1186/s12864-018-4889-1

Pan, X., and Shen, H-B. (2018). Predicting RNA–protein binding sites and motifs through combining local and global deep convolutional neural networks. *Bioinformatics* 34, 3427–3436. doi:10.1093/bioinformatics/bty364

Perlman, L., Gottlieb, A., Atias, N., Ruppin, E., and Sharan, R. (2011). Combining drug and gene similarity measures for drug-target elucidation. *J. Comput. Biol.* 18, 133–145. doi:10.1089/cmb.2010.0213

Rokach, L. (2010). Ensemble-based classifiers. *Artif. Intell. Rev.* 33, 1–39. doi:10.1007/s10462-009-9124-7

Salio, C., Lossi, L., Ferrini, F., and Merighi, A. (2006). Neuropeptides as synaptic transmitters. *Cell tissue Res.* 326, 583–598. doi:10.1007/s00441-006-0268-3

Southey, B. R., Amare, A., Zimmerman, T. A., Rodriguez-Zas, S. L., and Sweedler, J. V. (2006). NeuroPred: A tool to predict cleavage sites in neuropeptide precursors and provide the masses of the resulting peptides. *Nucleic acids Res.* 34, W267–W272. doi:10.1093/nar/gkl161

Svensson, M., Sköld, K., Svenningsson, P., and Andren, P. E. (2003). Peptidomics-based discovery of novel neuropeptides. *J. proteome Res.* 2, 213–219. doi:10.1021/pr020010u

UniProt Consortium (2021). UniProt: The universal protein knowledgeable in 2021. *Nucleic acids Res.* 49, D480–D489. doi:10.1093/nar/gkaa1100

Van der Maaten, L., and Hinton, G. (2008). Visualizing data using t-SNE. *J. Mach. Learn. Res.* 9. https://www.jmlr.org/papers/v9/vandermaaten08a.html

Van Eeckhaut, A., Maes, K., Aourz, N., Smolders, I., and Michotte, Y. (2011). The absolute quantification of endogenous levels of brain neuropeptides *in vivo* using LC–MS/MS. *Bioanalysis* 3, 1271–1285. doi:10.4155/bio.11.91

Van Wanseele, Y., De Prins, A., De Bundel, D., Smolders, I., and Van Eeckhaut, A. (2016). Challenges for the *in vivo* quantification of brain neuropeptides using microdialysis sampling and LC–MS. *Bioanalysis* 8, 1965–1985. doi:10.4155/bio-2016-0119

Wang, Y., Wang, M., Yin, S., Jang, R., Wang, J., Xue, Z., et al. (2015). NeuroPep: A comprehensive resource of neuropeptides. *Database* 2015, bav038. doi:10.1093/database/bav038

Wu, C., Gao, R., Zhang, Y., and De Marinis, Y. (2019). Ptpd: Predicting therapeutic peptides by deep learning and word2vec. *BMC Bioinforma.* 20, 456–458. doi:10.1186/s12859-019-3006-z

Xu, L., Liang, G., Wang, L., and Liao, C. (2018). A novel hybrid sequence-based model for identifying anticancer peptides. *Genes* 9, 158. doi:10.3390/genes9030158

Yang, H., Deng, Z., Pan, X., Shen, H. B., Choi, K. S., Wang, L., et al. (2021). RNA-binding protein recognition based on multi-view deep feature and multi-label learning. *Briefings Bioinforma.* 22, bbaa174. doi:10.1093/bib/bbaa174

frontiers | Frontiers in Genetics

TYPE Original Research
PUBLISHED 04 August 2023
DOI 10.3389/fgene.2023.1219894

OPEN ACCESS

EDITED BY
Lei Chen,
Shanghai Maritime University, China

REVIEWED BY
Rajeev Aurora,
Saint Louis University, United States
Baohong Liu,
Chinese Academy of Agricultural Sciences, China

*CORRESPONDENCE
Bin Yuan,
yuanbin8210@163.com

RECEIVED 08 June 2023
ACCEPTED 26 July 2023
PUBLISHED 04 August 2023

CITATION
Deng Y-J, Li Z, Wang B, Li J, Ma J, Xue X, Tian X, Liu Q-C, Zhang Y and Yuan B (2023), Immune-related gene *IL17RA* as a diagnostic marker in osteoporosis. *Front. Genet.* 14:1219894. doi: 10.3389/fgene.2023.1219894

Immune-related gene *IL17RA* as a diagnostic marker in osteoporosis

Ya-Jun Deng, Zhi Li, Bo Wang, Jie Li, Jun Ma, Xiong Xue, Xin Tian, Quan-Cheng Liu, Ying Zhang and Bin Yuan*

Department of Spine Surgery, Xi'an Daxing Hospital, Yanan University, Xi'an, China

Objectives: Bone immune disorders are major contributors to osteoporosis development. This study aims to identify potential diagnostic markers and molecular targets for osteoporosis treatment from an immunological perspective.

Method: We downloaded dataset GSE56116 from the Gene Expression Omnibus database, and identified differentially expressed genes (DEGs) between normal and osteoporosis groups. Subsequently, differentially expressed immune-related genes (DEIRGs) were identified, and a functional enrichment analysis was performed. A protein-protein interaction network was also constructed based on data from STRING database to identify hub genes. Following external validation using an additional dataset (GSE35959), effective biomarkers were confirmed using RT-qPCR and immunohistochemical (IHC) staining. ROC curves were constructed to validate the diagnostic values of the identified biomarkers. Finally, a ceRNA and a transcription factor network was constructed, and a Gene Ontology and Kyoto Encyclopedia of Genes and Genomes enrichment analysis was performed to explore the biological functions of these diagnostic markers.

Results: In total, 307 and 31 DEGs and DEIRGs were identified, respectively. The enrichment analysis revealed that the DEIRGs are mainly associated with Gene Ontology terms of positive regulation of MAPK cascade, granulocyte chemotaxis, and cytokine receptor. protein–protein interaction network analysis revealed 10 hub genes: *FGF8*, *KL*, *CCL3*, *FGF4*, *IL9*, *FGF9*, *BMP7*, *IL17RA*, *IL12RB2*, *CD40LG*. The expression level of *IL17RA* was also found to be significantly high. RT-qPCR and immunohistochemical results showed that the expression of *IL17RA* was significantly higher in osteoporosis patients compared to the normal group, as evidenced by the area under the curve Area Under Curve of 0.802. Then, we constructed *NEAT1*-hsa-miR-128-3p-*IL17RA*, and *SNHG1*-hsa-miR-128-3p-*IL17RA* ceRNA networks in addition to *ERF*-*IL17RA*, *IRF8*-*IL17RA*, *POLR2A*-*IL17RA* and *ERG*-*IL17RA* transcriptional networks. Finally, functional enrichment analysis revealed that *IL17RA* was involved in the development and progression of osteoporosis by regulating local immune and inflammatory processes in bone tissue.

Conclusion: This study identifies the immune-related gene *IL17RA* as a diagnostic marker of osteoporosis from an immunological perspective, and provides insight into its biological function.

KEYWORDS

IL17RA, osteoporosis, gene expression omnibus, DEIRGs, diagnostic markers

1 Introduction

Osteoporosis is the most common metabolic bone disease, characterized by reduced bone density and deterioration of bone tissue microarchitecture, increases bone fragility and the risk of fractures, leading to significant mortality (Chandra and Rajawat, 2021). Osteoporosis primarily affects postmenopausal women and men over the age of 50 (Borgström et al., 2020). The pathogenesis of osteoporosis exhibits noteworthy disparities between genders. In women, age-related bone loss and decreased estrogen secretion after menopause are the primary contributors to this condition (Shih et al., 2019). Estrogen plays a pivotal role in augmenting bone cell activity, inhibiting bone resorption, and averting calcium loss from bones. Moreover, estrogen restrains osteoclast formation and induces apoptosis in these cells, thereby curtailing bone resorption. In the absence of sufficient estrogen levels, osteoclast function heightens, leading to accelerated bone loss and the eventual onset of osteoporosis (Zhou et al., 2001). In men, the principal causes of osteoporosis include advancing age, prolonged glucocorticoid use, and declining testosterone levels (Vilaca et al., 2022). With age, inadequate testosterone levels impede the proliferation and differentiation of osteoblasts while intensifying osteoclast activity. Consequently, bone resorption escalates, resulting in subsequent loss of bone mass (Diab and Watts, 2021). It is evident that testosterone plays a pivotal role in the development of osteoporosis among elderly men.

Dual-energy x-ray absorptiometry (DXA) is considered the gold standard for diagnosing osteoporosis (Carey et al., 2022). Nonetheless, it has limitations when it comes to detecting early-stage bone loss. Early diagnosis and timely intervention are beneficial for preventing the development of osteoporosis (Sakka, 2022). In recent years, numerous studies have demonstrated that *CUL1*, *PTEN*, *STAT1*, *MAPKAPK2*, *RARRES2*, *FLNA*, *STXBP2*, miR-340-5p, and miR-506-3p could potentially serve as biomarkers for osteoporosis (Wang et al., 2022; Zhao Y. et al., 2022; Lu et al., 2023; Yuxuan et al., 2023). However, the majority of these molecular markers have not yet been validated using clinical samples. Thus, their potential for clinical applications remain limited. Therefore, there is still a need to find effective biomarkers for osteoporosis.

Osteoclasts, originating from hematopoietic cells of the myeloid lineage, play a crucial role in bone resorption (Thiolat et al., 2014). These cells undergo differentiation from osteoclast precursors when stimulated by M-CSF and RANKL (Zheng et al., 2014). Osteoblasts play a fundamental role in the synthesis of mineralized bone and are derived from a mesenchymal progenitor cell (Debnath et al., 2018). Multiple immune cells are involved in the regulation of osteoclast and osteoblast homeostasis. Th17 cells induce osteoclastogenesis by IL-17, Th1 cells activate osteoclast function through TNF-α (Adamopoulos and Bowman, 2008; Liu et al., 2011). Conversely, Sato et al. (2006) demonstrated that Th2 cells can impede osteoclast formation via IL-4. DCs enhance osteoclast activity by interacting with T cells through the RANK-RANKL signaling pathway (Santiago-Schwarz et al., 2001). Rivollier et al. (2004) revealed that DCs can transdifferentiate into osteoclasts *in vitro* in the presence of M-CSF and RANKL. Furthermore, B cells secrete RANKL, promoting osteoclast function (Kanematsu et al., 2000). Conversely, ILC2 cells suppress osteoclast formation through the release of IL-4 and IL-13 (Omata et al., 2020). Treg cells can inhibit monocyte differentiation into osteoclasts (Luo et al., 2011). Neutrophils hinder bone formation by affecting osteoblast function (Brunetti et al., 2013). M2 macrophages promote osteoblast differentiation (Vi et al., 2015). Conversely, M1 macrophages leads to bone resorption by increasing osteoclast activity and suppressing osteoblast-mediated bone formation (Bastian et al., 2011). *In vitro* studies have demonstrated the direct enhancement of osteoblast function by Treg cells (Lei et al., 2015). Additionally, B cells activate NF-κB signaling pathways to inhibit the differentiation of mesenchymal precursor cells into osteoblasts (Sun et al., 2018). Therefore, exploring the molecular mechanisms of osteoporosis from an immune perspective and developing new targets for immunotherapy is of great relevance for osteoporosis treatment.

Here, we performed a differential gene expression analysis on an osteoporosis microarray dataset downloaded from the Gene Expression Omnibus (GEO) database, and identified the intersection of differentially expressed genes (DEGs) with immune-related genes (IRGs) to determined differentially expressed immune-related genes (DEIRGs). Then, we constructed a protein–protein interaction (PPI) network to identify hub genes, and finally determined the immune-related gene *IL17RA* as a potential biomarker for osteoporosis after validating it in another dataset (GSE35959). RT-qPCR and immunohistochemical (IHC) staining were performed, and then ROC curves were constructed to verify its diagnostic value. In addition, we also explored the biological function of IL17RA by constructing ceRNA and transcription factor (TF) networks in addition to a Gene Ontology (GO) and Kyoto Encyclopedia of Genes and Genomes (KEGG) enrichment analysis to further elucidate the molecular mechanisms of osteoporosis in which *IL17RA* is involved.

2 Materials and methods

2.1 Microarray data

mRNA [GSE56116, https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE56116, GSE35959 (Benisch et al., 2012)], and miRNA microarray datasets [GSE201543 (Zhao S.-L. et al., 2022)] were downloaded from GEO (http://www.ncbi.nlm.nih.gov/geo/) by GEOquery package (Davis and Meltzer, 2007). GSE56116 was obtained from a GPL4133 Agilent-014850 Whole Human Genome Microarray 4 × 44 K G4112F. GSE35959 was obtained from a GPL570 (HG-U133_Plus_2) Affymetrix Human Genome U133 Plus 2.0 Array and GSE201543 was obtained from a GPL20712 Agilent-070156 Human miRNA (miRNA version). The probes were labeled with gene symbols, then multiple probes corresponding to the same gene were randomly selected to remove duplicates, and finally the gene expression matrix was obtained. GSE56116 contained 3 normal (healthy control) and 10 osteoporosis samples (4 kidney Yin deficiency, 3 kidney Yang deficiency, 3 non-kidney deficiency). GSE35959 contained 14 normal and 5 osteoporosis samples. GSE201543 contained 4 normal and 6 osteoporosis samples. GSE56116 (3 non-kidney deficiency osteoporosis samples and 3 normal samples), GSE35959, and GSE201543 were used as the training, validation, and miRNA validation datasets, respectively. Data of IRGs was downloaded from ImmPort database (https://www.immport.org/shared/), and finally 2,499 immune-related genes were obtained (Supplementary Table S1). The flow chart followed this study is shown in Figure 1.

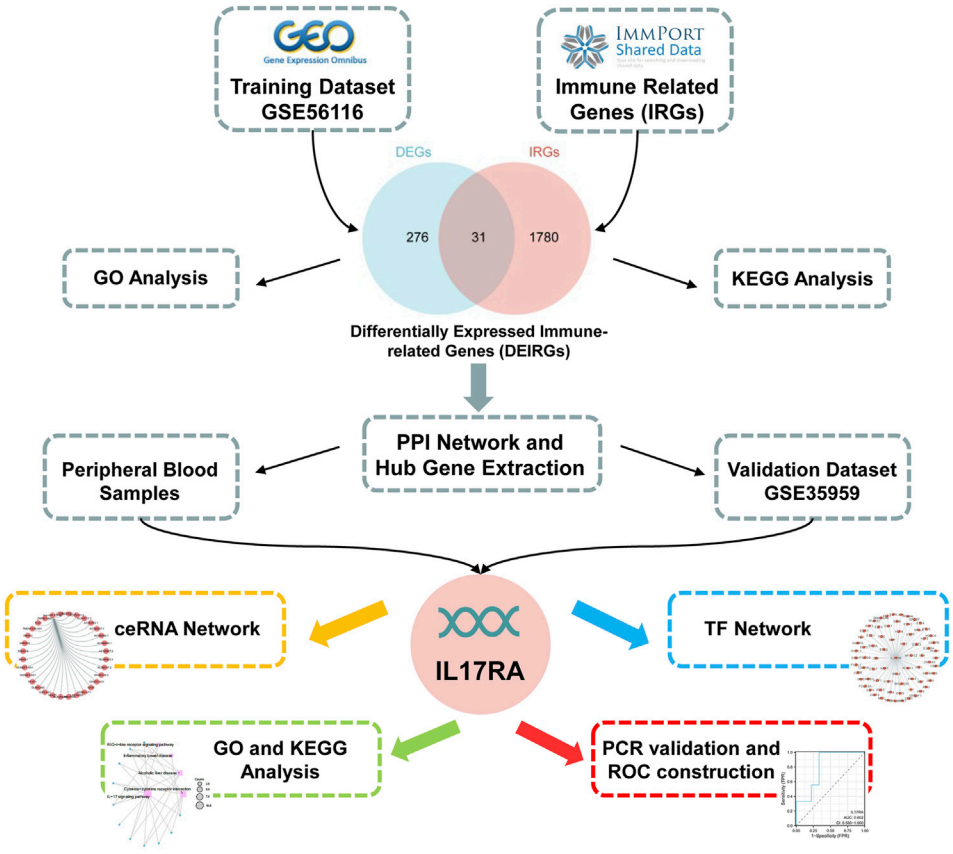


FIGURE 1
Flow chart of the study.

2.2 Identification of DEIRGs

The data set was normalized using the normalizeBetweenArrays function of the limma package (Ritchie et al., 2015). The sample correction is visualized by box plots, and the clustering between sample groups was visualized using PCA plots. Screen for DEGs between patients and controls using the limma package, and *p* value of lower than 0.05 and [log2 fold change (FC)] equal to or higher than 1 were set as the threshold values for DEG identification. After that, the intersection of DEGs and IRGs was determined to obtain list of DEIRGs. The results were visualized using the ggplot2 (Wickham, 2009) and the VennDiagram packages (Chen and Boutros, 2011). The results of DEGs and DEIRGs were visualized using the ggplot2 package for volcano plots and the Complex Heatmap package (Gu et al., 2016) for heat maps.

2.3 GO and KEGG enrichment analyses of DEIRGs

GO and KEGG functional enrichment analysis of DEIRGs were conducted using the cluster Profiler package (Yu et al., 2012). The results were visualized using the ggplot2 package. The human genome was used as a background reference, and a p.adj of lower than 0.05 was set as cut-off.

2.4 Construction of PPI network and selection of hub genes

The PPI network of DEIRGs was constructed using the STRING database (https://string-db.org/) (Szklarczyk et al., 2019). Interaction scores higher than 0.4 were considered significant. The results were visualized using Cytoscape software (Version: 3.9.1). Top 10 genes were determined as hub genes using the cytoHubba plugin and on the maximum correlation criterion algorithm.

2.5 External validation of hub genes

To identify effective biomarkers of osteoporosis, differences in hub gene expression levels between the osteoporosis and the normal groups were validated using another dataset (GSE35959).

2.6 Construction of ceRNA network

MiRNAs were predicted using four different databases [miRDB (https://mirdb.org/mirdb/index.html), TargetScanHuman (https://www.targetscan.org/vert_80/), TarBase (https://dianalab.e-ce.uth.gr/html/diana/web/index.php?r=tarbasev8) and miRWalk (http://mirwalk.umm.uni-heidelberg.de/)]. Furthermore, lncRNA-miRNA relationships

TABLE 1 Study subject demographics of peripheral blood.

Characteristics	Normal	Osteoporosis	*p* Value
n	9	9	
Gender, n (%)			1.000
Female	5 (27.8%)	5 (27.8%)	
Male	4 (22.2%)	4 (22.2%)	
Age (year)	56.11 ± 9.35	63 ± 11.12	0.174
BMI (kg/m^2)	24.63 ± 3.33	23.65 ± 2.37	0.482
BMD (g/cm^2)	0.98 ± 0.13	0.73 ± 0.08	< 0.001
T-score	−0.87 ± 1.13	−3.07 ± 0.58	< 0.001

of predicted hub genes-associated miRNAs were obtained by overlapping results from starBase (http://starbase.sysu.edu.cn/) and DIANA-LncBase v3 (https://diana.e-ce.uth.gr/lncbasev3). The overlap was visualized by ggplot2 and VennDiagram packages. Finally, a competitive endogenous RNA (ceRNA) network regulating hub genes was constructed by using Cytoscape software (Version: 3.9.1).

2.7 Construction of TF network

Prediction of hub genes and their TFs were performed by TF-Marker (http://bio.liclab.net/TF-Marker/) and GRNdb (http://www.grndb.com/), the overlap was visualized by ggplot2 and VennDiagram packages. A TF network regulating hub genes was constructed by using igraph package (Csardi and Nepusz, 2006).

2.8 Study subjects

Peripheral blood samples were obtained from nine patients with osteoporosis and nine healthy adults who were hospitalized in the Department of Spine Surgery at Xi'an Daxing Hospital, affiliated with Yan'an University, and underwent BMD testing between January 2023 and April 2023 (Table 1). Those with a history of long-term use of drugs affecting bone metabolism, endocrine system disorders, spinal tumors or spinal tuberculosis were not included in this study. Bone tissue samples were obtained from twelve patients with osteoporotic compression fractures who were hospitalized for vertebroplasty surgery, with mild osteoporosis (−3. 5 < T-score ≤ −2. 5) and severe osteoporosis (T-score ≤ −3. 5), six in each group (Table 2). All patients underwent MRI and DXA of the spine, which confirmed the presence of fresh fractures and osteoporosis. Inclusion criteria were as follows: 1) age ≥50 years, BMD T-score ≤ −2.5; 2) vertebral fragility fracture, biopsy routinely performed during vertebroplasty. Exclusion criteria were as follows: previous long-term use of drugs affecting bone metabolism, presence of endocrine system diseases, spinal tumors and spinal tuberculosis. The diagnosis of osteoporosis was confirmed based on the classifcation criteria of the World Health Organization (WHO) based on T-score of BMD testing (Yoshimura et al., 2022): T-score ≥ −1.0 was considered normal bone mass, −2.5 < T-score < −1.0 was considered decreased bone mass, and T-score ≤ −2.5 was considered osteoporosis. All included subjects were informed of the medical record review and study design and signed consent documents before data collection. The Ethics Committee of the Xi'an Daxing Hospital, affiliated with Yan'an University approved and reviewed the study protocol.

2.9 Peripheral blood collection

Five milliliters peripheral blood samples were collected the morning following an overnight fast. The serum was obtained following centrifugation (3,000 r/min, 5 min) of blood samples and submitted for bone metabolism marker detection. Cell sediment was collected for RNA extraction. All samples were frozen at −80°C until analysis.

2.10 Bone tissue sample acquisition

The patient was placed in the prone position, and a 0.5–1 cm piece of cancellous bone tissue was drilled using a 14G bone biopsy ring perched on the fracture area within the vertebral body under local anesthesia via the arch root approach. The bone tissue was fixed in 10% neutral-buffered formalin for 1 week, followed by routine decalcification, dehydration, and paraffin embedding for subsequent studies.

2.11 BMD measurements

BMD measurements of the lumbar spine were performed using DXA (QDR X-Ray Bone Densitometer, Hologic, United States). All data were measured by the same group of imaging physicians in strict accordance with the specifications for measuring BMD by DXA.

2.12 RT-qPCR analysis

Total RNA was extracted using RNA Extraction Solution (G3013, Servicebio, Wuhan, China), and RNA concentration and purity were measured by Nanodrop 2000 spectrophotometer (Thermo Scientific, Waltham, United States). The RNA samples were reverse transcribed into cDNA using a reverse transcription kit (G3337, Servicebio, Wuhan, china), and the cDNA was used as a template to amplify the IL17RA gene. The reaction was performed via 40 amplification cycles using the following protocol: Denaturation at 95°C for 30 s, annealing at 60°C for 30 s, extension at 72°C for 60 s. Samples were analyzed in triplicate, the mRNA expression levels of IL17RA was calculated by the $2^{-\Delta\Delta CT}$ method, and GAPDH was used as internal reference. The sequences of the primers are listed in Table3.

2.13 HE and IHC staining

The wax blocks were placed in a paraffin slicer for continuous sectioning, with each section having 4 μm thickness. HE staining was performed using HE staining solution (G1003, Servicebio, Wuhan, China). For IHC, paraffin tissue sections were deparaffinized with xylene and rehydrated with an alcohol gradient and water. Sections were incubated with primary

TABLE 2 Study subject demographics of bone tissue.

Characteristics	Mild osteoporosis	Severe osteoporosis	*p* Value
n	6	6	
Gender, n (%)			1.000
Female	3 (25%)	4 (33.3%)	
Male	3 (25%)	2 (16.7%)	
Age (year)	65.33 ± 10.03	66.83 ± 12.62	0.824
BMI (kg/m^2)	22.39 ± 1.45	20.79 ± 1.28	0.070
BMD (g/cm^2)	0.73 ± 0.05	0.58 ± 0.08	0.002
T-score	−2.78 ± 0.15	−3.95 ± 0.53	0.002

TABLE 3 Primer sequences used for RT-qPCR.

Gene	Forward primer (5′–3′)	Reverse primer (5′–3′)
IL17RA	CCAACATCACCGTGGAGACC	GTGGCGACAGCACCCTTTAA
GAPDH	GGAAGCTTGTCATCAATGGAAATC	TGATGACCCTTTTGGCTCCC

antibodies IL17RA (Catalogue number: DF3602, diluted 1:100, Affinity), at room temperature for 1 h and biotin-labelled secondary antibodies for 30 min, and then stained with DAB peroxidase substrate kit (G1212, Servicebio, Wuhan, China). Finally, washed with water and counterstained with haematoxylin. The results were observed and photographed by an microscope (Eclipse C1, Nikon, Japan).

2.14 Statistical analysis

All data processing and analysis were conducted using R software (version 4.2.1). RT-qPCR were repeated three times, and data were represented as the mean ± SD. Normality was tested using the Shapiro-Wilk normality test and chi-squaredness was tested using Levene's test. Student's t-test or Wilcoxon rank sum test was used to determine the significance of difference between two groups. Correlation coefficients were calculated using Spearman correlation analysis. ROCs were used to evaluate AUCs and predictive abilities. A *p* value lower than 0.05 was considered statistically significant.

3 Results

3.1 Identification of DEIRGs

The median, upper and lower quartiles, maximum and minimum values of each sample gene were significantly close to each other upon normalization of GSE56116 data (Supplementary Figure S1A, B). However, PCA revealed that the centers of the osteoporosis group were farther apart than those of the control group, indicating significant differences in gene expression between the two groups (Supplementary Figure S1C, D). Using a *p* value of lower than 0.05 and a [log2 fold change (FC)] equal to or higher than 1 as the threshold levels, we identified 307 DEGs, including 94 and 213 significantly up- and down-regulated genes, respectively (Supplementary Table S2). Figures 2A, B show results in the form of volcano plots and heat maps (Figures 2A, B). The intersection of DEGs and IRGs included 31 genes (DEIRGs) (Figure 2C; Supplementary Table S3), including 11 and 20 up- and down-regulated genes, respectively (Figure 2D).

3.2 Functional enrichment analyses of DEIRGs

We performed GO and KEGG enrichment analysis to investigate the functions of DEIRGs. In the GO analysis, biological processes (BPs), cell components (CCs), and molecular functions (MFs) were distinguished. The BPs included regulation of chemotaxis, positive regulation of MAPK cascade, granulocyte chemotaxis, cell chemotaxis and granulocyte migration. CCs included clathrin−coated endocytic vesicle membrane, clathrin−coated endocytic vesicle, clathrin−coated vesicle membrane, serine−type peptidase complex and semaphorin receptor complex. Finally, MFs included signaling receptor activator activity, receptor ligand activity, growth factor activity, fibroblast growth factor receptor binding and growth factor receptor binding. KEGG analysis showed that DEIRGs were mainly associated with Cytokine−cytokine receptor interaction, Viral protein interaction with cytokine and cytokine receptor. Figures 3A–D show the top five enrichment items of BP, CC, MF in GO and KEGG analyses.

3.3 Construction of the PPI network and identification of hub genes

The STRING database was used to construct a PPI network of 31 DEIRGs in order to investigate protein-protein interactions. A

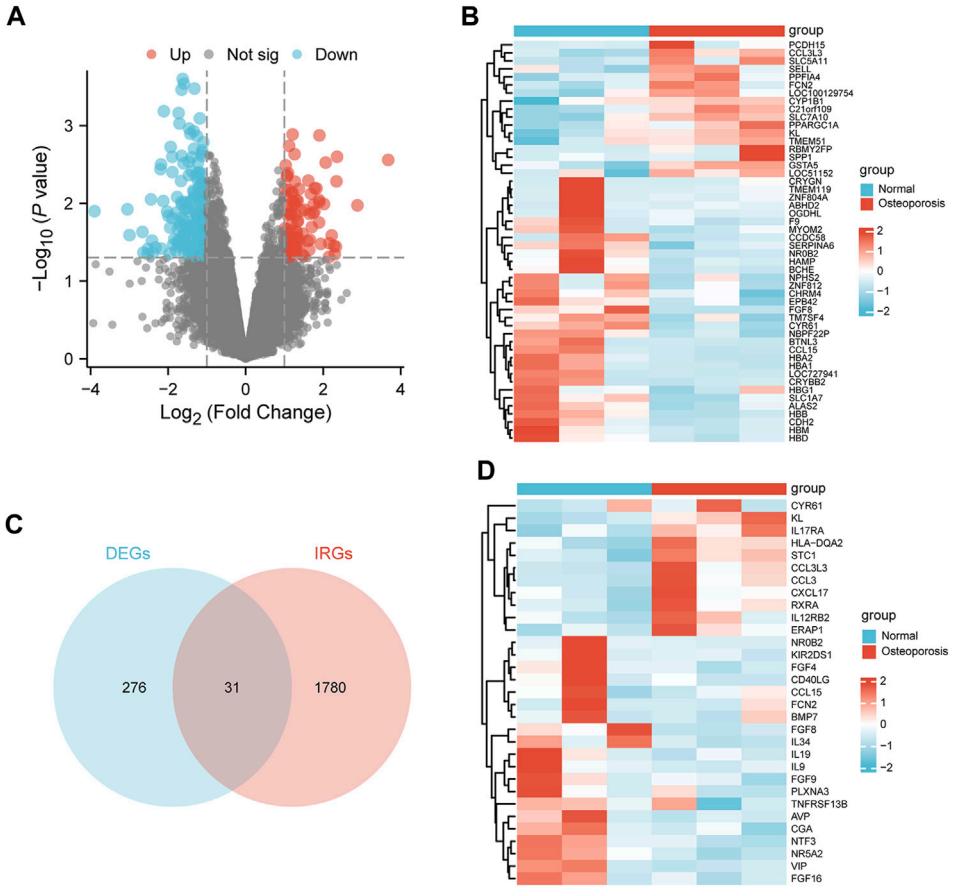


FIGURE 2
Identification of DEIRGs. **(A)** Volcano plot, **(B)** heatmap of DEGs between the osteoporosis and normal samples. **(C)** Venn diagram of overlapping genes between the DEGs and IRGs. **(D)** Heatmap of DEIRGs.

total of 30 nodes and 31 edges were identified in the PPI network (Figure 4A). The cytohubba plug-in of Cytoscape software was then used to select the top 10 hub genes based on their degree of connectivity (Figure 4B; Table 4).

3.4 Validation of diagnostic biomarkers

In the GSE35959 dataset, osteoporotic patients had significantly higher *IL17RA* expression levels than those of patients in the control group ($p < 0.05$) (Figure 5A; Supplementary Figure S2). To confirm the higher expression level of *IL17RA* in osteoporotic patients and its diagnostic performance, we validated this finding using clinical peripheral blood and bone tissue. RT-qPCR results showed that mRNA expression levels of *IL17RA* were significantly higher in peripheral blood of patients with osteoporosis compared to those of patients in the control group ($p < 0.05$) (Figure 5B). The IHC results showed that *IL17RA* expression was higher in the severe osteoporosis group compared to the mild osteoporosis group (Figure 5C). The horizontal and vertical coordinates of the ROC curve indicate sensitivity and specificity, respectively. A larger AUC indicates a more accurate diagnostic model. Accordingly, the AUC was 0.802 (Figure 5D), indicating significant differences between OP and control groups. Hence, *IL17RA* expression level could serve as a good diagnostic biomarker.

3.5 Construction of ceRNA network

We analyzed upstream regulation of *IL17RA*, and screened for miRNAs or lncRNAs targeting *IL17RA*. We identified 142, 1,423, 13, and 2,113 miRNAs possibly targeting *IL17RA* from miRDB, TargetScanHuman, TarBase, and miRWalk databases, respectively (Figure 6A). Consequently, we determined hsa-miR-128-3p to be the most important miRNA regulator by comparing predictions based on each database. Complementary sequences between *IL17RA* and hsa-miR-128-3p are displayed in Figure 6B. We validated hsa-miR-128-3p expression in the GSE201543 dataset, and found that expression level of hsa-miR-128-3p was significantly low in osteoporotic patients (Figure 6C). Next, 30 lncRNAs that could bind to hsa-miR-128-3p were obtained from the overlapping results of DIANA-LncBase v3 and starBase databases (Figure 6D). A lncRNA-miRNA-mRNA network regulating *IL17RA* was constructed, in which lncRNAs competitively bind to miRNAs and attenuate the inhibition

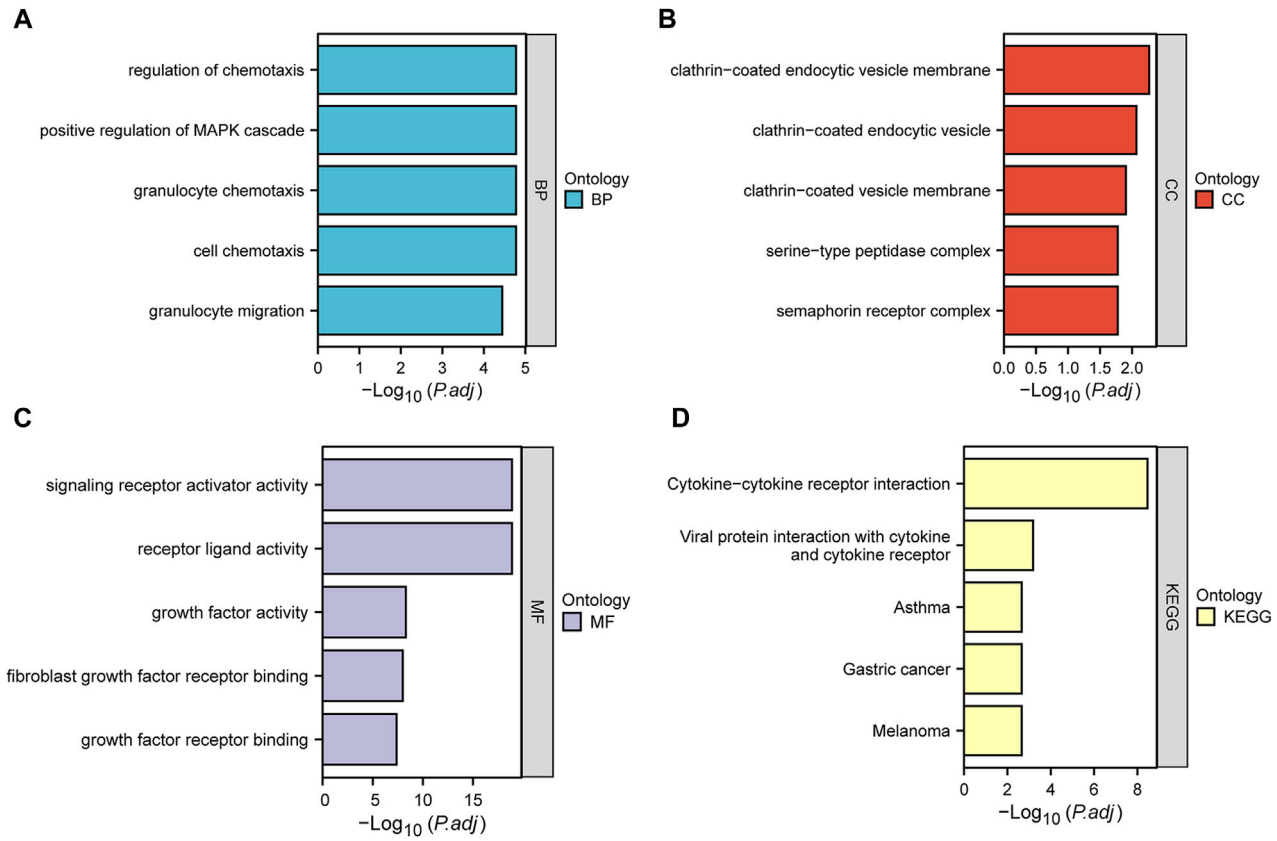


FIGURE 3
Bar plots of 31 DEIRGs-enriched GO terms and KEGG pathways. **(A–D)** represent BP, CC, MF, and KEGG, respectively.

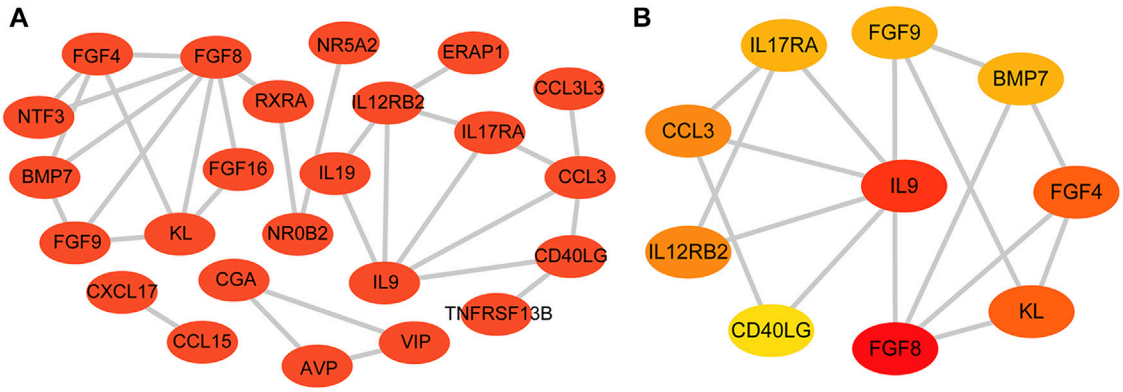


FIGURE 4
PPI network and hub genes. **(A)** PPI network constructed with the DEIRGs; **(B)** Top 10 hub genes.

of *IL17RA* by miRNAs (Figure 6E). A review of the literature was used to determine these 30 lncRNAs. Our findings indicated that expression levels of *NEAT1* and *SNHG1* were significantly high in osteoporotic patients. Since a significantly high level of *IL17RA* expression and a significantly low level of hsa-miR-128-3p were found in osteoporotic patients, the interactions predicted by the above database (Figure 6F) further led us to hypothesize that *NEAT1* and *SNHG1* bind to hsa-miR-128-3p, and impair the inhibitory effect of hsa-miR-128-3p on *IL17RA* in osteoporosis (Figure 6G).

TABLE 4 Top 10 hub genes.

Gene symbol	Entrez id	Full name	logFC	*p*-value
FGF8	2253	Fibroblast growth factor 8	−2.4426	0.008896
KL	9365	Klotho	2.327417	0.036147
CCL3	6348	C-C motif chemokine ligand 3	2.064109	0.002997
FGF4	2249	Fibroblast growth factor 4	−1.60047	0.003061
IL9	3578	Interleukin 9	−1.49079	0.037584
FGF9	2254	Fibroblast growth factor 9	−1.23376	0.024612
BMP7	655	Bone morphogenetic protein 7	−1.23075	0.028784
IL17RA	23765	Interleukin 17 receptor A	1.162541	0.022572
IL12RB2	3595	Interleukin 12 receptor subunit beta 2	1.093632	0.042559
CD40LG	959	CD40 ligand	−1.07323	0.015333

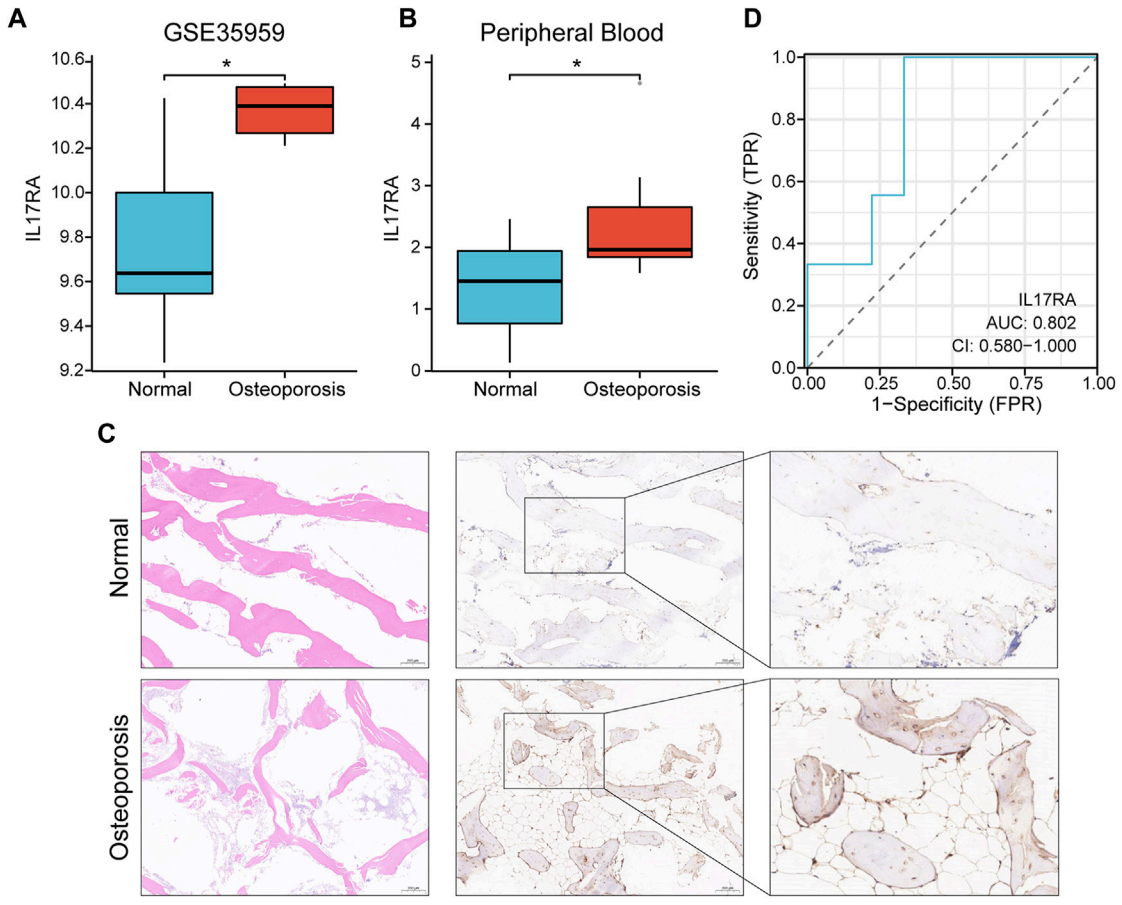


FIGURE 5
Verification of the diagnostic effectiveness of *IL17RA*. **(A)** *IL17RA* expression in the GSE35959 dataset. **(B)** *IL17RA* expression in peripheral blood samples. **(C)** *IL17RA* expression in bone tissue samples. **(D)** ROC curve.

3.6 Transcriptome analysis

To better understand gene expression upstream of *IL17RA*, we performed a transcriptome analysis. First, we obtained 607 and 166 TFs regulating *IL17RA* from the TF-Marker and GRNdb databases, respectively. A total of 75 TFs were found in both databases (Figure 7A). Using these, an *IL17RA* transcriptional regulatory network was constructed (Figure 7B). We selected

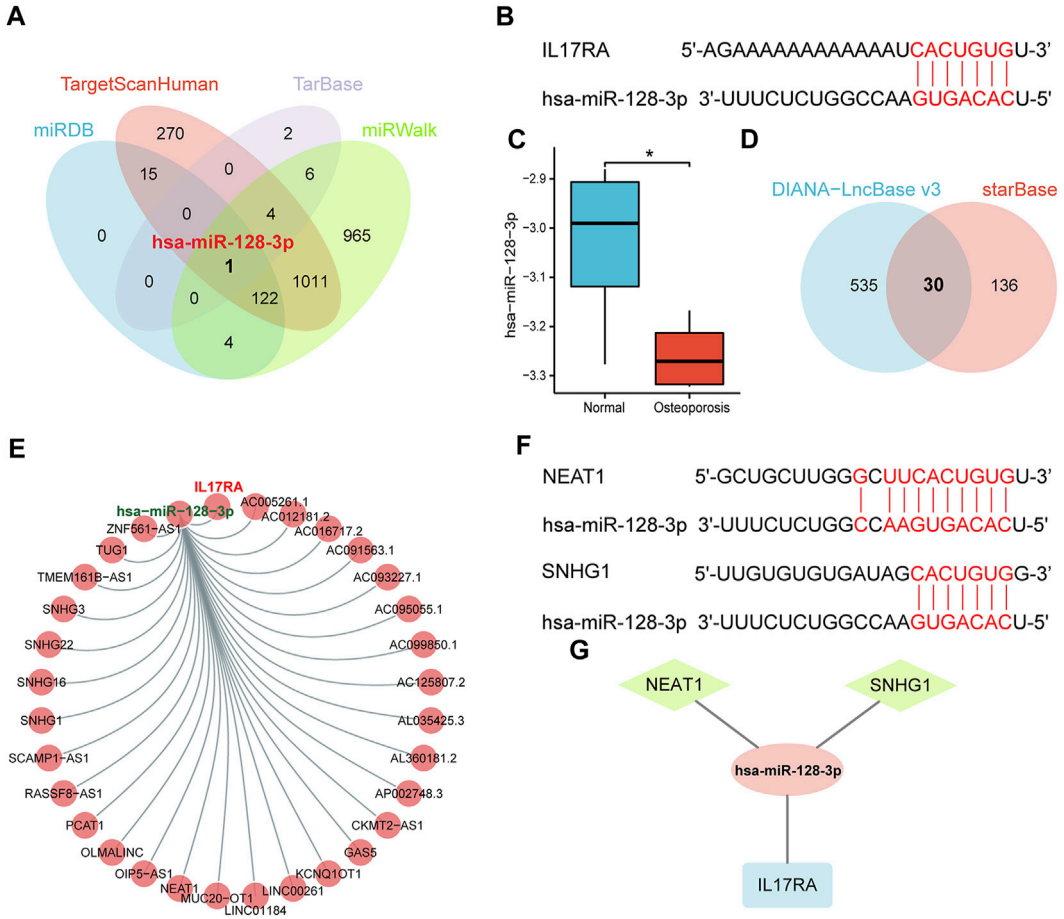


FIGURE 6
The ceRNA regulatory network of *IL17RA*. **(A)** Prediction of miRNAs targeting *IL17RA* using four different databases. **(B)** Predicted interaction between hsa-miR-128-3p and *IL17RA*. **(C)** Expression of hsa-miR-128-3p in the GSE201543 dataset. **(D)** Prediction of lncRNAs targeting hsa-miR-128-3p using two different databases. **(E)** lncRNA-miRNA-mRNA network of IL17RA. **(F)** Predicted interactions between *NEAT1*, *SNHG1* and hsa-miR-128-3p. **(G)** A ceRNA network consisting of *IL17RA*, hsa-miR-128-3p, *NEAT1* and *SNHG1* in osteoporosis.

nine TFs that showed significant differential expression between the osteoporosis and normal groups (Figure 7C). Among these TFs, *ERF* ($R = 0.661$, $p = 0.044$), *IRF8* ($R = 0.709$, $p = 0.028$), *POLR2A* ($R = 0.867$, $p = 0.003$) and *ERG* ($R = -0.867$, $p = 0.003$) were found to be correlated with *IL17RA* (Figure 7D; Supplementary Figure S3). Based on this, we constructed the osteoporosis *ERF-IL17RA*, *IRF8-IL17RA*, *POLR2A-IL17RA* and *ERG-IL17RA* transcriptional networks (Figure 7E).

3.7 GO and KEGG pathway enrichment analysis of diagnostic biomarkers

To investigate the downstream regulatory roles of *IL17RA*, we used the STRING database to predict 10 *IL17RA*-interacting genes (using a confidence score of equal to or higher than 0.4), and constructed a PPI network using Cytoscape (Figure 8A). A total of five KEGG pathways were highlighted by KEGG analysis of *IL17RA* and *IL17RA*-interacting genes: IL−17 signaling pathway, Cytokine−cytokine receptor interaction, alcoholic liver disease, inflammatory bowel disease, and RIG−I−like receptor signaling pathway (Figure 8B). The GO enrichment analysis results indicated that cytokine receptor binding, cytokine activity, immune receptor activity, cytokine receptor activity, and thioesterase binding were the top 5 MF terms (Figure 8C). Cytokine−mediated signaling pathway, interleukin−17−mediated signaling pathway, cellular response to interleukin−17, and response to interleukin−17, positive regulation of interleukin−6 production were the top 5 BP terms (Figure 8D). Finally, plasma membrane signaling receptor complex, cytoplasmic side of membrane, cytoplasmic side of plasma membrane, CD40 receptor complex, and lipid droplet were the top 5 CC terms (Figure 8E).

3.8 Gene Set Enrichment Analysis (GSEA)

To investigate the functions of IL17RA in osteoporosis, we conducted Gene Set Enrichment Analysis (GSEA) by stratifying samples based on IL17RA expression. The results enriched several important pathways, including "INTERFERON_ALPHA_RESPONSE," "INTERFERON_GAMMA_RESPONSE," "IL6_

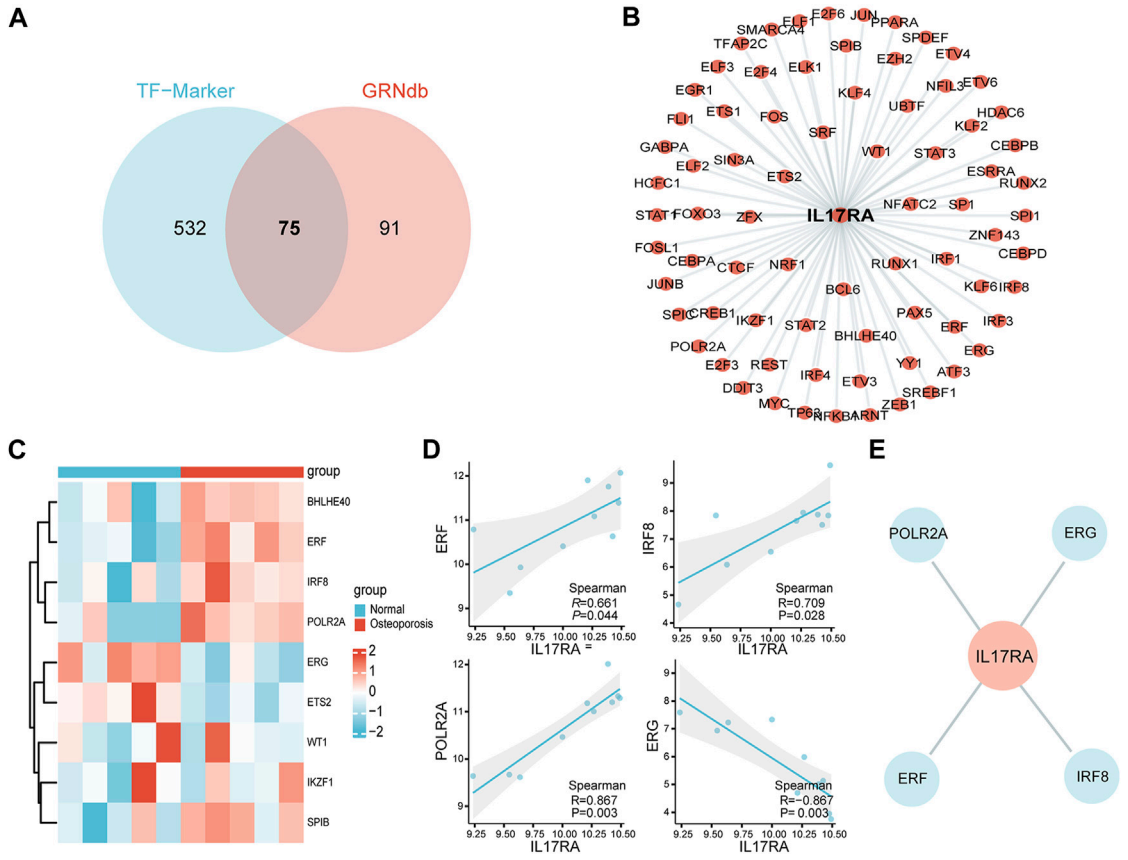


FIGURE 7
Transcriptional network of *IL17RA*. **(A)** Predicted TFs associated with *IL17RA* based on TF-Marker and GRNdb databases. **(B)** *IL17RA* transcriptional regulatory network. **(C)** Heat map of TFs expression between osteoporosis and normal groups. **(D)** Spearman correlation between *ERF*, *IRF8*, *POLR2A* and *ERG* and *IL17RA*. **(E)** Transcriptional network between *ERF*, *IRF8*, *POLR2A*, *ERG* and *IL17RA* in osteoporosis.

JAK_STAT3_SIGNALING", "INFLAMMATORY_RESPONSE", "REACTIVE_OXYGEN_SPECIES_PATHWAY", and "TNFA_SIGNALING_VIA_NFKB" (Supplementary Figure S4). These pathways play a critical role in immune response, pro-inflammatory reactions, cytokine signaling, and other vital biological processes. The identification and enrichment of these pathways shed light on the intricate connections between IL17RA and multiple molecular mechanisms involved in maintaining bone health and homeostasis.

4 Discussion

Osteoporosis is characterized by reduced bone strength and an increased risk of fracture. It is estimated that more than 200 million people worldwide suffer from osteoporosis, with 30%–50% of women experiencing fractures due to osteoporosis during their lifetime (Rachner et al., 2011). Since osteoporosis patients typically exhibit no obvious clinical symptoms before their first fracture, early diagnosis is crucial for timely intervention and pain relief. Hence, there is an urgent need for effective molecular diagnostic markers. Previous studies suggested that the immune system may play a significant role in osteoporosis development (Sapra et al., 2022; Wang et al., 2022), yet the specific immune targets and molecular mechanisms of osteoporosis remain unknown. Microarray technology has enabled the exploration of genetic alterations in osteoporosis, and has proven effective in identifying novel biomarkers for other diseases. In this study, we used bioinformatics methods to identify diagnostic markers for osteoporosis, and validated their diagnostic value using peripheral blood from osteoporosis patients.

An analysis of transcriptome data from peripheral blood samples of osteoporosis patients and healthy individuals yielded a total of 307 DEGs, including 94 up- and 213 down-regulated genes, respectively. The intersection of DEGs and IRGs yielded a total of 31 DEIRGs, including 11 and 20 up- and down-regulated genes, respectively. GO enrichment analysis of DEIRGs showed that the GO terms were associated with positive regulation of MAPK cascade, granulocyte chemotaxis, growth factor activity, and semaphorin receptor complex. KEGG analysis showed that DEIRGs were mainly associated with Cytokine–cytokine receptor interaction, Viral protein interaction with cytokine and cytokine receptor. These findings suggest that immunomodulation plays a significantly role in the development of osteoporosis. MAPK and innate

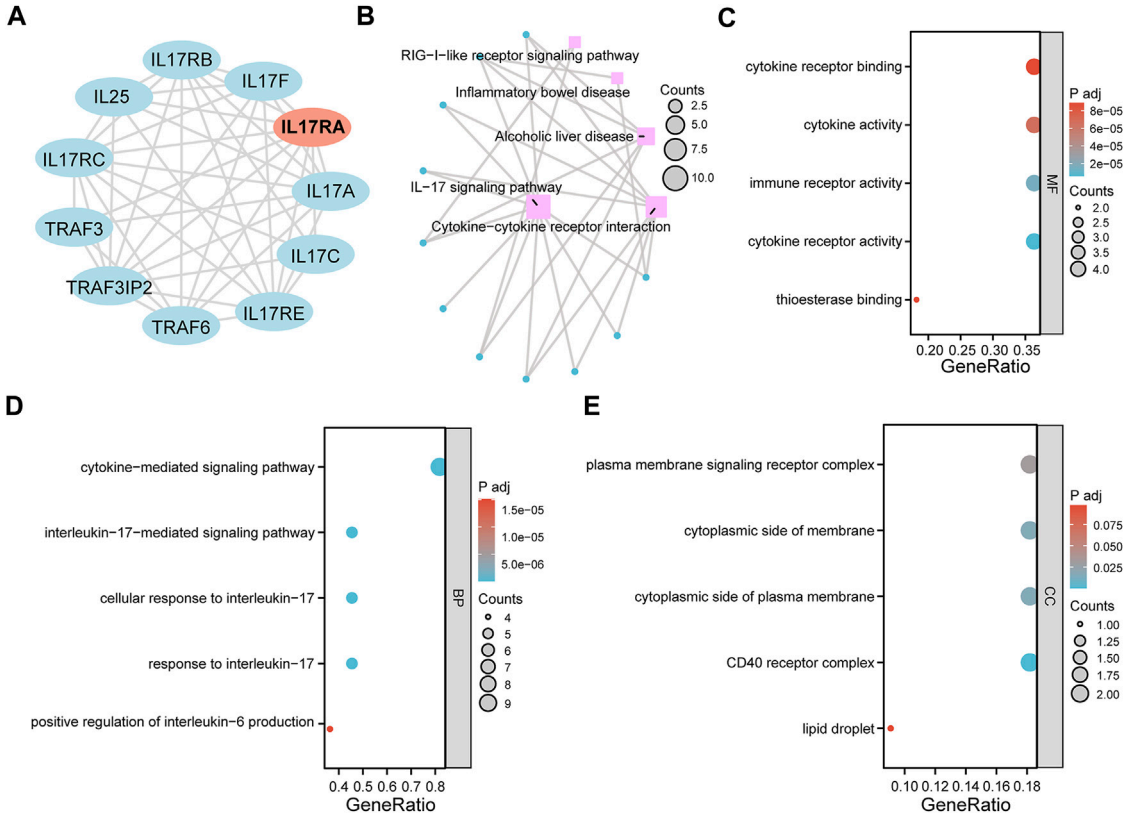


FIGURE 8
GO and KEGG pathway enrichment analysis. **(A)** An PPI network of *IL17RA* constructed with data from STRING. **(B)** KEGG enrichment analysis network diagram. **(C)** MF **(D)** BP **(E)** CC enrichment analysis bubble diagram.

immune signaling pathways are closely-linked through feedback regulation (Kitajima et al., 2018). Previous studies have reported that the MAPK signaling pathway is involved in the regulation of bone metabolism and osteoclast formation (Meng et al., 2021; Wang et al., 2021). Neutrophil chemokines stimulate the growth and development of osteoblasts and chondrocytes (Mori et al., 1997). Cytokine-cytokine receptor interaction, and viral protein interaction with cytokine suggests significant involvement of the immune system and inflammatory cytokines in the progression of osteoporosis. Inflammatory factors inhibit bone formation in part by suppressing osteoblast differentiation, which includes inhibition of Wnt signaling. In addition, they also promote bone resorption by inducing osteoclast differentiation and bone resorption functions, which in turn disrupt bone homeostasis and contribute to the progression of osteoporosis (Ivashkiv et al., 2011). Hence, dysregulation of the immune system can have a detrimental effect on bone integrity, leading to osteoporosis. Our results are consistent with previous findings.

The PPI network analysis revealed 10 key genes associated with osteoporosis: *FGF8*, *KL*, *CCL3*, *FGF4*, *IL9*, *FGF9*, *BMP7*, *IL17RA*, *IL12RB2*, *CD40LG*. Expression level of *IL17RA* was found to be significantly high in osteoporotic patients upon external dataset validation, suggesting that *IL17RA* may be an effective biomarker for osteoporosis. To further confirm the diagnostic performance of *IL17RA*, we verified *IL17RA* expression by RT-qPCR and IHC, and plotted ROC curves. RT-qPCR results showed that the mRNA expression level of *IL17RA* was significantly higher in peripheral blood of osteoporotic patients compared to that of the control group. IHC results were in line with RT-qPCR. The Area Under Curve (AUC) was 0.802, suggesting a high diagnostic value and potential of *IL17RA* as a diagnostic marker for osteoporosis.

The IL-17 family of inflammatory cytokines has gained attention as major contributors to bone formation and bone resorption. Most IL-17 cytokines act by signaling through the receptor complex of *IL17RA*. *IL17RA* signaling in osteoclast precursors were previously demonstrated to contribute to osteoclast formation and subsequent bone loss. Moreover, *IL17RA* deficiency increases bone mass by decreasing the abundance of osteoclast precursors (Roberts et al., 2022). In addition, IL*17RA* in osteoblasts/osteoclasts mediates parathyroid hormone-induced bone loss and enhances osteoblast RANKL production (Li et al., 2019). These studies are in line with our findings. However, Goswami et al. (2009) used the ovariectomy-induced osteoporosis (OVX) model in *IL17RA* (−/−) mice to assess the role of IL*17A* in estrogen deficiency-induced bone loss. The authors showed that *IL17RA* (−/−) mice were consistently more susceptible to OVX-induced

bone loss than controls. *IL17A* inhibits bone resorption-related protease expression and osteoclast differentiation in RAW264.7 cells via *IL17RA* (Kitami et al., 2010). These findings suggests that *IL17RA* signaling plays an osteoprotective role in ovariectomy-induced bone loss. This also shows that the role of *IL17RA* in osteoporosis is still controversial, and an increased sample size is needed for an in-depth analysis.

The concept of CeRNA was introduced in 2011 (Salmena et al., 2011). In the ceRNA network, non-coding RNAs, such as lncRNAs or circRNAs, can compete to bind to miRNAs, and thereby weaken the repression of mRNAs by miRNAs. We identified hsa-miR-128-3p as a key regulatory miRNA for IL17RA in osteoporosis. Previous research has indicated that hsa-miR-128-3p can inhibit osteoblast differentiation of bone marrow mesenchymal stem cells by downregulating *RUNX1*, *YWHAB* and *NTRK2* (Zhang W. et al., 2020). In addition, hsa-miR-128-3p promoted the proliferation, migration and osteoclast differentiation of RAW 264.7 cells and upregulated the osteoclastogenic markers c-Fos, NFATc1 and Ctsk (Zhang et al., 2022a). These findings suggests that hsa-miR-128-3p inhibits osteoblast differentiation and promotes osteoclast formation, which is inconsistent with our findings here. Further studies are needed to explain this paradox, and identify other mechanisms involving has-miR-128-3p in osteoporosis. We hypothesize that *NEAT1* and *SNHG1* target hsa-miR-128-3p. Studies have shown that *NEAT1* promotes the proliferation and differentiation of osteoblasts and, regulates the development and progression of osteoporosis (Zhang Y. et al., 2020; Zhao X. et al., 2022). *SNHG1* expression is up-regulated in OVX mice, which inhibits osteoblast differentiation and angiogenesis while promoting osteoclast formation, leading to osteoporosis (Yu et al., 2021; Yu et al., 2022). *NEAT1* and *SNHG1* are thus promising targets for the treatment of osteoporosis. The above-mentioned findings support the conclusions of our study. We constructed the NEAT1-hsa-miR-128-3p-IL17RA and SNHG1-hsa-miR-128-3p-IL17RA networks to provide a theoretical basis for understanding the molecular mechanisms of *IL17RA* involvement in osteoporosis.

We performed a transcriptional analysis as well. The *ERF* (ETS2 repressor factor) is located on Chromosome 19q13.2, and encodes a transcription factor bound directly by ERK1/2 to regulate the RAS-MEK-ERK signal transduction cascade (von Kriegsheim et al., 2009). A study found that reduced doses of ERF lead to complex cranial suture closure in humans and mice, and highlighted *ERF* as a novel regulator of osteogenic stimulation of RAS-ERK signaling (Sr et al., 2013). IRF8 inhibits osteoclastogenesis, and is involved in the development and progression of osteoporosis (Zhao et al., 2009; Jin et al., 2023). RNA polymerase II subunit A (*POLR2A*) encodes the largest catalytic subunit of the RNA polymerase II complex. Liu et al. (2021) showed that *POLR2A* blocks osteoclastic bone resorption and prevented osteoporosis by interacting with *CREB1*. *ERG* is closely associated with Ewing sarcoma (Dunn et al., 1994), cervical cancer (Zhang Z. et al., 2020) and prostate cancer (Dawoud et al., 2021). However, its role in bone metabolism remains unexplored. The TF network constructed here provides a clear direction to better understand the upstream transcriptional mechanism of *IL17RA*. To further investigate the downstream regulatory role of *IL17RA*, we also performed a functional enrichment analysis of *IL17RA* and its interacting genes. Accordingly, *IL17RA* may be involved in the development and progression of osteoporosis by regulating local immune and inflammatory processes in bone tissue.

There were also some limitations in this study. First, the sample size in the dataset selected for this study was small. Although we standardized the raw data, a larger sample size and a higher quality dataset are still needed to verify the reliability of the results. Secondly, although we validated the diagnostic value of *IL17RA* using patients' peripheral blood samples and bone tissues, the sample size of this study was also limited, and the clinical translational value of IL17RA needs to be validated in a larger number of clinical osteoporosis samples. Finally, a more comprehensive study on molecular biological mechanisms involving *IL17RA* on both cellular and animal levels is needed.

In conclusion, we identified the immune-related gene *IL17RA* as a diagnostic marker of osteoporosis by elucidating its biological function within the immune system. Our findings may provide with a potential immune molecular target for the early diagnosis and treatment of osteoporosis.

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 in the article/Supplementary Material.

Ethics statement

The studies involving human participants were reviewed and approved by Xi'an Daxing Hospital affiliated to Yanan University. The patients/participants provided their written informed consent to participate in this study.

Author contributions

Y-JD and BY contributed to the conception and design of the study. ZL, BW, and JL acquired the data. JM and Q-CL performed the data analysis. XX and XT wrote the first draft of the manuscript. YZ revised the manuscript critically. All authors contributed to the article and approved the submitted version.

Funding

This work was supported by the National Science Foundation of China (No. 82260443).

Acknowledgments

We would like to thank Editage (www.editage.cn) for English language editing.

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/fgene.2023.1219894/full#supplementary-material

References

Adamopoulos, I. E., and Bowman, E. P. (2008). Immune regulation of bone loss by Th17 cells. *Arthritis Res. Ther.* 10, 225. doi:10.1186/ar2502

Bastian, O., Pillay, J., Alblas, J., Leenen, L., Koenderman, L., and Blokhuis, T. (2011). Systemic inflammation and fracture healing. *J. Leukoc. Biol.* 89, 669–673. doi:10.1189/jlb.0810446

Benisch, P., Schilling, T., Klein-Hitpass, L., Frey, S. P., Seefried, L., Raaijmakers, N., et al. (2012). The transcriptional profile of mesenchymal stem cell populations in primary osteoporosis is distinct and shows overexpression of osteogenic inhibitors. *PLoS One* 7, e45142. doi:10.1371/journal.pone.0045142

Borgström, F., Karlsson, L., Ortsäter, G., Norton, N., Halbout, P., Cooper, C., et al. (2020). Fragility fractures in europe: Burden, management and opportunities. *Arch. Osteoporos.* 15, 59. doi:10.1007/s11657-020-0706-y

Brunetti, G., Faienza, M. F., Piacente, L., Ventura, A., Oranger, A., Carbone, C., et al. (2013). High dickkopf-1 levels in sera and leukocytes from children with 21-hydroxylase deficiency on chronic glucocorticoid treatment. *Am. J. Physiol. Endocrinol. Metab.* 304, E546–E554. doi:10.1152/ajpendo.00535.2012

Carey, J. J., Chih-Hsing Wu, P., and Bergin, D. (2022). Risk assessment tools for osteoporosis and fractures in 2022. *Best. Pract. Res. Clin. Rheumatol.* 36, 101775. doi:10.1016/j.berh.2022.101775

Chandra, A., and Rajawat, J. (2021). Skeletal aging and osteoporosis: Mechanisms and therapeutics. *Int. J. Mol. Sci.* 22, 3553. doi:10.3390/ijms22073553

Chen, H., and Boutros, P. C. (2011). VennDiagram: A package for the generation of highly-customizable Venn and euler diagrams in R. *BMC Bioinforma.* 12, 35. doi:10.1186/1471-2105-12-35

Csardi, G., and Nepusz, T. (2006). *The igraph software package for complex network research*.

Davis, S., and Meltzer, P. S. (2007). GEOquery: A bridge between the gene expression omnibus (GEO) and BioConductor. *Bioinformatics* 23, 1846–1847. doi:10.1093/bioinformatics/btm254

Dawoud, M. M., Aiad, H. A.-S., Bahbah, A. M. N. H., and Shaban, M. I. (2021). Comparative study of immunohistochemical expression of ERG and MAGI2 in prostatic carcinoma. *Ann. Diagn Pathol.* 52, 151727. doi:10.1016/j.anndiagpath.2021.151727

Debnath, S., Yallowitz, A. R., McCormick, J., Lalani, S., Zhang, T., Xu, R., et al. (2018). Discovery of a periosteal stem cell mediating intramembranous bone formation. *Nature* 562, 133–139. doi:10.1038/s41586-018-0554-8

Diab, D. L., and Watts, N. B. (2021). Updates on osteoporosis in men. *Endocrinol. Metab. Clin. North Am.* 50, 239–249. doi:10.1016/j.ecl.2021.03.001

Dunn, T., Praissman, L., Hagag, N., and Viola, M. V. (1994). ERG gene is translocated in an Ewing's sarcoma cell line. *Cancer Genet. Cytogenet* 76, 19–22. doi:10.1016/0165-4608(94)90063-9

Goswami, J., Hernández-Santos, N., Zuniga, L. A., and Gaffen, S. L. (2009). A bone-protective role for IL-17 receptor signaling in ovariectomy-induced bone loss. *Eur. J. Immunol.* 39, 2831–2839. doi:10.1002/eji.200939670

Gu, Z., Eils, R., and Schlesner, M. (2016). Complex heatmaps reveal patterns and correlations in multidimensional genomic data. *Bioinformatics* 32, 2847–2849. doi:10.1093/bioinformatics/btw313

Ivashkiv, L. B., Zhao, B., Park-Min, K.-H., and Takami, M. (2011). Feedback inhibition of osteoclastogenesis during inflammation by IL-10, M-CSF receptor shedding, and induction of IRF8. *Ann. N. Y. Acad. Sci.* 1237, 88–94. doi:10.1111/j.1749-6632.2011.06217.x

Jin, W., Chen, F., Fang, Q., Mao, G., and Bao, Y. (2023). Oligosaccharides from Sargassum thunbergii inhibit osteoclast differentiation via regulation of IRF-8 signaling. *Exp. Gerontol.* 172, 112057. doi:10.1016/j.exger.2022.112057

Kanematsu, M., Sato, T., Takai, H., Watanabe, K., Ikeda, K., and Yamada, Y. (2000). Prostaglandin E2 induces expression of receptor activator of nuclear factor-kappa B ligand/osteoprotegrin ligand on pre-B cells: Implications for accelerated osteoclastogenesis in estrogen deficiency. *J. Bone Min. Res.* 15, 1321–1329. doi:10.1359/jbmr.2000.15.7.1321

Kitajima, S., Asahina, H., Chen, T., Guo, S., Quiceno, L. G., Cavanaugh, J. D., et al. (2018). Overcoming resistance to dual innate immune and MEK inhibition downstream of KRAS. *Cancer Cell* 34, 439–452. doi:10.1016/j.ccell.2018.08.009

Kitami, S., Tanaka, H., Kawato, T., Tanabe, N., Katono-Tani, T., Zhang, F., et al. (2010). IL-17A suppresses the expression of bone resorption-related proteinases and osteoclast differentiation via IL-17RA or IL-17RC receptors in RAW264.7 cells. *Biochimie* 92, 398–404. doi:10.1016/j.biochi.2009.12.011

Lei, H., Schmidt-Bleek, K., Dienelt, A., Reinke, P., and Volk, H.-D. (2015). Regulatory T cell-mediated anti-inflammatory effects promote successful tissue repair in both indirect and direct manners. *Front. Pharmacol.* 6, 184. doi:10.3389/fphar.2015.00184

Li, J.-Y., Yu, M., Tyagi, A. M., Vaccaro, C., Hsu, E., Adams, J., et al. (2019). IL-17 receptor signaling in osteoblasts/osteocytes mediates PTH-induced bone loss and enhances osteocytic RANKL production. *J. Bone Min. Res.* 34, 349–360. doi:10.1002/jbmr.3600

Liu, C., Han, Y., Zhao, X., Li, B., Xu, L., Li, D., et al. (2021). POLR2A blocks osteoclastic bone resorption and protects against osteoporosis by interacting with CREB1. *J. Cell Physiol.* 236, 5134–5146. doi:10.1002/jcp.30220

Liu, Y., Wang, L., Kikuiri, T., Akiyama, K., Chen, C., Xu, X., et al. (2011). Mesenchymal stem cell-based tissue regeneration is governed by recipient T lymphocytes via IFN-γ and TNF-α. *Nat. Med.* 17, 1594–1601. doi:10.1038/nm.2542

Lu, Z., Cao, H., and Hu, X. (2023). Circulating miR-340-5p and miR-506-3p as two osteo-miRNAs for predicting osteoporosis in a cohort of postmenopausal women. *J. Environ. Public Health* 2023, 7571696. doi:10.1155/2023/7571696

Luo, C. Y., Wang, L., Sun, C., and Li, D. J. (2011). Estrogen enhances the functions of CD4(+)CD25(+)Foxp3(+) regulatory T cells that suppress osteoclast differentiation and bone resorption *in vitro*. *Cell Mol. Immunol.* 8, 50–58. doi:10.1038/cmi.2010.54

Meng, J., Zhang, X., Guo, X., Cheng, W., Qi, X., Huang, J., et al. (2021). Briarane-type diterpenoids suppress osteoclastogenisis by regulation of Nrf2 and MAPK/NF-kB signaling pathway. *Bioorg Chem.* 112, 104976. doi:10.1016/j.bioorg.2021.104976

Mori, Y., Hiraki, Y., Shukunami, C., Kakudo, S., Shiokawa, M., Kagoshima, M., et al. (1997). Stimulation of osteoblast proliferation by the cartilage-derived growth promoting factors chondromodulin-I and -II. *FEBS Lett.* 406, 310–314. doi:10.1016/s0014-5793(97)00291-3

Omata, Y., Frech, M., Lucas, S., Primbs, T., Knipfer, L., Wirtz, S., et al. (2020). Type 2 innate lymphoid cells inhibit the differentiation of osteoclasts and protect from ovariectomy-induced bone loss. *Bone* 136, 115335. doi:10.1016/j.bone.2020.115335

Rachner, T. D., Khosla, S., and Hofbauer, L. C. (2011). Osteoporosis: Now and the future. *Lancet* 377, 1276–1287. doi:10.1016/S0140-6736(10)62349-5

Ritchie, M. E., Phipson, B., Wu, D., Hu, Y., Law, C. W., Shi, W., et al. (2015). Limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Res.* 43, e47. doi:10.1093/nar/gkv007

Rivollier, A., Mazzorana, M., Tebib, J., Piperno, M., Aitsiselmi, T., Rabourdin-Combe, C., et al. (2004). Immature dendritic cell transdifferentiation into osteoclasts: A novel pathway sustained by the rheumatoid arthritis microenvironment. *Blood* 104, 4029–4037. doi:10.1182/blood-2004-01-0041

Roberts, J. L., Mella-Velazquez, G., Dar, H. Y., Liu, G., and Drissi, H. (2022). Deletion of IL-17ra in osteoclast precursors increases bone mass by decreasing osteoclast precursor abundance. *Bone* 157, 116310. doi:10.1016/j.bone.2021.116310

Sakka, S. D. (2022). Osteoporosis in children and young adults. *Best. Pract. Res. Clin. Rheumatol.* 36, 101776. doi:10.1016/j.berh.2022.101776

Salmena, L., Poliseno, L., Tay, Y., Kats, L., and Pandolfi, P. P. (2011). A ceRNA hypothesis: The rosetta stone of a hidden RNA language? *Cell* 146, 353–358. doi:10.1016/j.cell.2011.07.014

Santiago-Schwarz, F., Anand, P., Liu, S., and Carsons, S. E. (2001). Dendritic cells (DCs) in rheumatoid arthritis (RA): Progenitor cells and soluble factors contained in RA synovial fluid yield a subset of myeloid DCs that preferentially activate Th1 inflammatory-type responses. *J. Immunol.* 167, 1758–1768. doi:10.4049/jimmunol.167.3.1758

Sapra, L., Shokeen, N., Porwal, K., Saini, C., Bhardwaj, A., Mathew, M., et al. (2022). Bifidobacterium longum Ameliorates Ovariectomy-induced bone loss via enhancing anti-osteoclastogenic and immunomodulatory potential of regulatory B cells (Bregs). *Front. Immunol.* 13, 875788. doi:10.3389/fimmu.2022.875788

Sato, K., Suematsu, A., Okamoto, K., Yamaguchi, A., Morishita, Y., Kadono, Y., et al. (2006). Th17 functions as an osteoclastogenic helper T cell subset that links T cell activation and bone destruction. *J. Exp. Med.* 203, 2673–2682. doi:10.1084/jem.20061775

Shih, Y.-R. V., Liu, M., Kwon, S. K., Iida, M., Gong, Y., Sangaj, N., et al. (2019). Dysregulation of ectonucleotidase-mediated extracellular adenosine during postmenopausal bone loss. *Sci. Adv.* 5, eaax1387. doi:10.1126/sciadv.aax1387

Sr, T., E, V., Sj, M., I, P., Al, F., Vp, S., et al. (2013). Reduced dosage of ERF causes complex craniosynostosis in humans and mice and links ERK1/2 signaling to regulation of osteogenesis. *Nat. Genet.* 45, 308–313. doi:10.1038/ng.2539

Sun, W., Meednu, N., Rosenberg, A., Rangel-Moreno, J., Wang, V., Glanzman, J., et al. (2018). B cells inhibit bone formation in rheumatoid arthritis by suppressing osteoblast differentiation. *Nat. Commun.* 9, 5127. doi:10.1038/s41467-018-07626-8

Szklarczyk, D., Gable, A. L., Lyon, D., Junge, A., Wyder, S., Huerta-Cepas, J., et al. (2019). STRING v11: Protein-protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. *Nucleic Acids Res.* 47, D607–D613. doi:10.1093/nar/gky1131

Thiolat, A., Semerano, L., Pers, Y. M., Biton, J., Lemeiter, D., Portales, P., et al. (2014). Interleukin-6 receptor blockade enhances CD39+ regulatory T cell development in rheumatoid arthritis and in experimental arthritis. *Arthritis Rheumatol.* 66, 273–283. doi:10.1002/art.38246

Vi, L., Baht, G. S., Whetstone, H., Ng, A., Wei, Q., Poon, R., et al. (2015). Macrophages promote osteoblastic differentiation *in-vivo*: Implications in fracture repair and bone homeostasis. *J. Bone Min. Res.* 30, 1090–1102. doi:10.1002/jbmr.2422

Vilaca, T., Eastell, R., and Schini, M. (2022). Osteoporosis in men. *Lancet Diabetes Endocrinol.* 10, 273–283. doi:10.1016/S2213-8587(22)00012-2

von Kriegsheim, A., Baiocchi, D., Birtwistle, M., Sumpton, D., Bienvenut, W., Morrice, N., et al. (2009). Cell fate decisions are specified by the dynamic ERK interactome. *Nat. Cell Biol.* 11, 1458–1464. doi:10.1038/ncb1994

Wang, G., Wang, F., Zhang, L., Yan, C., and Zhang, Y. (2021). miR-133a silencing rescues glucocorticoid-induced bone loss by regulating the MAPK/ERK signaling pathway. *Stem Cell Res. Ther.* 12, 215. doi:10.1186/s13287-021-02278-w

Wang, X., Zhiwei, P., Ting, H., Jirigala, A., Siqin, L., Wanxiong, H., et al. (2022). Prognostic analysis and validation of diagnostic marker genes in patients with osteoporosis. *Front. Immunol.* 13, 987937. doi:10.3389/fimmu.2022.987937

Wickham, H. (2009). *ggplot2: Elegant graphics for data analysis*. New York, NY: Springer New York. doi:10.1007/978-0-387-98141-3

Yoshimura, N., Iidaka, T., Horii, C., Muraki, S., Oka, H., Kawaguchi, H., et al. (2022). Trends in osteoporosis prevalence over a 10-year period in Japan: The ROAD study 2005-2015. *J. Bone Min. Metab.* 40, 829–838. doi:10.1007/s00774-022-01352-4

Yu, G., Wang, L.-G., Han, Y., and He, Q.-Y. (2012). clusterProfiler: an R package for comparing biological themes among gene clusters. *OMICS* 16, 284–287. doi:10.1089/omi.2011.0118

Yu, X., Rong, P.-Z., Song, M.-S., Shi, Z.-W., Feng, G., Chen, X.-J., et al. (2021). lncRNA SNHG1 induced by SP1 regulates bone remodeling and angiogenesis via sponging miR-181c-5p and modulating SFRP1/Wnt signaling pathway. *Mol. Med.* 27, 141. doi:10.1186/s10020-021-00392-2

Yu, X., Song, M.-S., Rong, P.-Z., Chen, X.-J., Shi, L., Wang, C.-H., et al. (2022). LncRNA SNHG1 modulates adipogenic differentiation of BMSCs by promoting DNMT1 mediated Opg hypermethylation via interacting with PTBP1. *J. Cell Mol. Med.* 26, 60–74. doi:10.1111/jcmm.16982

Yuxuan, D., Yunyun, W., Qing, S., and Yanxia, J. (2023). Identification of hub genes associated with osteoporosis development by comprehensive bioinformatics analysis. *Front. Genet.* 14, 1028681. doi:10.3389/fgene.2023.1028681

Zhang, H., Chen, L., Wang, Z., Sun, Z., Shan, Y., Li, Q., et al. (2022a). Long noncoding RNA KCNQ1OT1 inhibits osteoclast differentiation by regulating the miR-128-3p/NFAT5 axis. *Aging (Albany NY)* 14, 4486–4499. doi:10.18632/aging.204088

Zhang, W., Zhu, Y., Chen, J., Wang, J., Yao, C., and Chen, C. (2020a). Mechanisms of miR-128-3p in inhibiting osteoblast differentiation from bone marrow-derived mesenchymal stromal cells. *Mol. Med. Rep.* 22, 5041–5052. doi:10.3892/mmr.2020.11600

Zhang, Y., Chen, X.-F., Li, J., He, F., Li, X., and Guo, Y. (2020b). lncRNA Neat1 stimulates osteoclastogenesis via sponging miR-7. *J. Bone Min. Res.* 35, 1772–1781. doi:10.1002/jbmr.4039

Zhang, Z., Chen, F., Li, S., Guo, H., Xi, H., Deng, J., et al. (2020c). ERG the modulates Warburg effect and tumor progression in cervical cancer. *Biochem. Biophys. Res. Commun.* 522, 191–197. doi:10.1016/j.bbrc.2019.11.079

Zhao, B., Takami, M., Yamada, A., Wang, X., Koga, T., Hu, X., et al. (2009). Interferon regulatory factor-8 regulates bone metabolism by suppressing osteoclastogenesis. *Nat. Med.* 15, 1066–1071. doi:10.1038/nm.2007

Zhao, S.-L., Wen, Z.-X., Mo, X.-Y., Zhang, X.-Y., Li, H.-N., Cheung, W.-H., et al. (2022a). Bone-metabolism-related serum microRNAs to diagnose osteoporosis in middle-aged and elderly women. *Diagn. (Basel)* 12, 2872. doi:10.3390/diagnostics12112872

Zhao, X., Zhao, D., Geng, B., Yaobin, W., and Xia, Y. (2022b). A novel ceRNA regulatory network involving the long noncoding NEAT1, miRNA-466f-3p and its mRNA target in osteoblast autophagy and osteoporosis. *J. Mol. Med. Berl.* 100, 1629–1646. doi:10.1007/s00109-022-02255-7

Zhao, Y., Li, W., Zhang, K., Xu, M., Zou, Y., Qiu, X., et al. (2022c). Revealing oxidative stress-related genes in osteoporosis and advanced structural biological study for novel natural material discovery regarding MAPKAPK2. *Front. Endocrinol. (Lausanne)* 13, 1052721. doi:10.3389/fendo.2022.1052721

Zheng, T., Wang, X., and Yim, M. (2014). Miconazole inhibits receptor activator of nuclear factor-κB ligand-mediated osteoclast formation and function. *Eur. J. Pharmacol.* 737, 185–193. doi:10.1016/j.ejphar.2014.04.047

Zhou, S., Zilberman, Y., Wassermann, K., Bain, S. D., Sadovsky, Y., and Gazit, D. (2001). Estrogen modulates estrogen receptor alpha and beta expression, osteogenic activity, and apoptosis in mesenchymal stem cells (MSCs) of osteoporotic mice. *J. Cell Biochem. Suppl. Suppl.* 36, 144–155. doi:10.1002/jcb.1096

frontiers | Frontiers in Genetics

TYPE Perspective
PUBLISHED 10 August 2023
DOI 10.3389/fgene.2023.1209416

OPEN ACCESS

EDITED BY
Nathan Olson,
National Institute of Standards and Technology (NIST), United States

REVIEWED BY
Ettore Mosca,
National Research Council (CNR), Italy

*CORRESPONDENCE
Alberto J. M. Martin,
alberto.martin@uss.cl
Inti Pedroso,
intipedroso@gmail.com

RECEIVED 20 April 2023
ACCEPTED 24 July 2023
PUBLISHED 10 August 2023

CITATION
Latapiat V, Saez M, Pedroso I and Martin AJM (2023), Unraveling patient heterogeneity in complex diseases through individualized co-expression networks: a perspective.
Front. Genet. 14:1209416.
doi: 10.3389/fgene.2023.1209416

Unraveling patient heterogeneity in complex diseases through individualized co-expression networks: a perspective

Verónica Latapiat[1,2,3], Mauricio Saez[4,5], Inti Pedroso[2]* and Alberto J. M. Martin[3,6]*

[1]Programa de Doctorado en Genómica Integrativa, Vicerrectoría de Investigación, Universidad Mayor, Santiago, Chile, [2]Vicerrectoría de Investigación, Universidad Mayor, Santiago, Chile, [3]Laboratorio de Redes Biológicas, Centro Científico y Tecnológico de Excelencia Ciencia & Vida, Fundación Ciencia & Vida, Santiago, Chile, [4]Centro de Oncología de Precisión, Facultad de Medicina y Ciencias de la Salud, Universidad Mayor, Santiago, Chile, [5]Laboratorio de Investigación en Salud de Precisión, Departamento de Procesos Diagnósticos y Evaluación, Facultad de Ciencias de la Salud, Universidad Católica de Temuco, Temuco, Chile, [6]Escuela de Ingeniería, Facultad de Ingeniería, Arquitectura y Diseño, Universidad San Sebastián, Santiago, Chile

This perspective highlights the potential of individualized networks as a novel strategy for studying complex diseases through patient stratification, enabling advancements in precision medicine. We emphasize the impact of interpatient heterogeneity resulting from genetic and environmental factors and discuss how individualized networks improve our ability to develop treatments and enhance diagnostics. Integrating system biology, combining multimodal information such as genomic and clinical data has reached a tipping point, allowing the inference of biological networks at a single-individual resolution. This approach generates a specific biological network per sample, representing the individual from which the sample originated. The availability of individualized networks enables applications in personalized medicine, such as identifying malfunctions and selecting tailored treatments. In essence, reliable, individualized networks can expedite research progress in understanding drug response variability by modeling heterogeneity among individuals and enabling the personalized selection of pharmacological targets for treatment. Therefore, developing diverse and cost-effective approaches for generating these networks is crucial for widespread application in clinical services.

KEYWORDS

personalized medicine, omics, transcriptomic, co-expression, networks, diseases

1 Introduction

Complex diseases arise from the intricate interplay of multiple genetic and environmental risk factors. The phenomenon of simplexity, where simplicity at the phenotypic level coexists with complexity at lower organizational and molecular levels (Stewart and Cohen, 2000; Kauffman et al., 1993), suggests the existence of disease subtypes (Wallstrom et al., 2013) and emphasizes the uniqueness of each patient despite shared characteristics with others (Smith, 2011). Unfortunately, most approaches to studying complex diseases rely on identifying differences between groups based on average biomarker values, overlooking the intricate biological intricacies of these diseases. For this reason, it is necessary to use a more holistic approach that considers the molecular

complexity of diseases, which involves thousands of genes across multiple cell types in different body parts (H. Zhang et al., 2019) and poses challenges for developing personalized, targeted therapies (Sierksma et al., 2020; Rouzier et al., 2005; Shipitsin et al,. 2007; Charitou et al., 2016; Khurana et al., 2013; Chan and Loscalzo, 2012).

Network biology is a rapidly developing area of research that recognizes that biological processes are not chiefly controlled by individual proteins or by discrete, unconnected linear pathways but rather by a complex system-level network of molecular interactions (X.-M. Zhang et al., 2021; Khurana et al., 2013; Charitou et al., 2016). Graph neural networks and deep-learning-based data integration models can predict disease progression and identify disease subtypes more accurately by integrating multimodal data from disparate sources, such as genetic, clinical, and imaging data (X.-M. Zhang et al., 2021; Zhou et al., 2022). Therefore, a more holistic approach that considers the molecular complexity of diseases and integrates multimodal data can provide a more comprehensive understanding of complex diseases, leading to the development of personalized, targeted therapies and improved patient outcomes in the era of precision medicine.

Cancer is a prime example of disease heterogeneity, where variability exists in various aspects, including driver mutations, making it challenging to identify causal mutations from an average view of the entire patient cohort (Lengerich et al., 2018). Moreover, diseases such as Autism spectrum disorders and epilepsy exhibit vast degrees of heterogeneity at multiple levels, including genotypes and phenotypes, resulting in diverse clinical differentiations and treatment responses (Lombardo et al., 2019). The clinical variability observed in diseases like Parkinson's and Alzheimer's further highlights the need to go beyond mean values and explore other approaches that capture the heterogeneous nature of complex diseases (Freudenberg-Hua et al., 2018; Ma et al., 2018).

Clinical studies of diseases often suffer from biases due to demographic, social, genetic, and ethnic factors, leading to the underrepresentation of specific population groups (Prosperi et al., 2018). This underrepresentation hampers the generalizability of conclusions to a larger population, hindering the development of effective treatments (Kessler et al., 2016; Popejoy and Fullerton, 2016; Popejoy et al., 2018; Gurdasani et al., 2019). The failure of numerous clinical trials and the lack of a cure for diseases like Alzheimer's emphasize the need to account for population heterogeneity in trial design and consider the underlying biological mechanisms for disease subtyping (Devi and Scheltens, 2018).

While challenges exist in identifying biomarkers for heterogeneous diseases, scale-out learning approaches often need more specificity and may not be applicable in clinical practice (Khurana et al., 2013). Additionally, invasive and costly procedures or limited access to relevant tissues hinder studying central nervous system diseases (Koníčková et al., 2022). Therefore, it is necessary to adopt new approaches that precisely consider the underlying biological mechanisms in disease subtyping (Yin et al., 2019), incorporating clinical and omics analyses to improve treatment responses (Zhou et al., 2022; X.-M; Zhang et al., 2021).

The study of complex diseases is not only a scientific effort but also a public health concern. The increasing availability of drugs that can contribute to molecular-tailored treatments based on predictive biomarkers underscores the importance of improving our understanding of individual patients to enhance their quality of life (Zhou et al., 2022). To address these challenges, we require new approaches that exponentially scale up learning on complex diseases, enabling a deeper understanding of each individual and more effective interventions (X.-M. Zhang et al., 2021). By embracing these novel approaches, we can advance our knowledge of complex diseases, refine disease subtyping, and guide the selection of personalized treatment strategies to improve patient outcomes and enhance public health.

1.1 Individualized networks and personalized medicine

Individualized networks and personalized medicine are essential for accelerating the development of new therapies for complex diseases. Unlike the current reductionist approach, we require a system-level understanding of individuals, which can be achieved through biological networks (Ahn et al., 2006; Younesi and Hofmann-Apitius, 2013). Biological networks provide a systems-level understanding of disease mechanisms, enabling the identification of differential molecular mechanisms altered in different subtypes of disease and the disease's progression trajectory. Networks integrate data from multiple patients to predict disease subtypes and progression, facilitating the identification of prognostic biomarkers (Furlong, 2013; Younesi and Hofmann-Apitius, 2013; McGillivray et al., 2018). Computational strategies for biological network inference have been developed to improve our understanding of biological systems (Browne et al., 2009; Liu et al., 2016; Lengerich et al., 2018; Van Der Wijst et al., 2018; Zanin et al., 2018).

Developing new therapies requires a system-level understanding of individuals with complex diseases. Biological networks are a powerful tool for this approach, enabling the modeling of complex systems (Ahn et al., 2006; Younesi and Hofmann-Apitius, 2013). By integrating data from several patients, biological networks can predict differential molecular mechanisms altered in different disease subtypes and identify the progression trajectory of the disease (Fröhlich et al., 2018). Network analysis can lead to identifying prognostic sets of biomarkers and constructing explanatory models proving their value for precision medicine. Computational strategies through biological network inference have been developed and widely validated to improve our understanding of biological systems (Browne et al., 2009). Networks can be analyzed based on graph theory tools, such as determining node properties like degree, betweenness, and other centralities (Mulder et al., 2014), and global or local graph-theoretical features describing the network may constitute potential prognostic biomarkers instead of or in addition to traditional covariates. Machine learning and artificial intelligence techniques have been employed to analyze networks (Zitnik and Leskovec, 2017; Agrawal et al., 2018; Ma et al., 2018; Zitnik et al., 2018), allowing for the identification of gene signatures that serve as prognostic markers, as demonstrated in clear renal cell carcinoma patients (Büttner et al., 2019). Several authors have developed computational strategies through biological network inference (Liu et al., 2016; Lengerich et al., 2018; Van Der Wijst et al., 2018; Zanin et al., 2018), and network-based analytics plays an

TABLE 1 Summary of study design in biological networks.

	Networks in a whole population	Case versus control network comparison	Personalized networks
Experimental design	Generation of one network from a population	Generation of two or more networks representing cases and controls	Generation of one network per sample/individual
Analytical protocol	To obtain network modules and associate each of them with disease status	To find condition-specific clusters of individuals based on the comparison of networks	Network comparison to identify modules for each sample
		To identify structural network differences associated with modules in disease status	Association of network structure and the presence/absence of modules to disease status
Pros	Allow study correlation relation among genes in samples	Allow finding in general sense differences and making comparisons among control and case samples.	Network for each individual allows representing of what happens in each subject
cons	The resultant network does not represent the variation in the population	Network of cases and control allows represent a consensus of the group studied	Coexpression network methods have minimal samples to consider in analysis (30 samples) to reach optimal performance

increasingly important role in precision medicine (W. Zhang et al., 2017). These strategies provide a comprehensive approach to modeling biological systems, enabling construction of explanatory models that can inform precision medicine.

Furthermore, individual-specific network analysis is valuable for prediction modeling in medicine and applied health research, identifying potential prognostic biomarkers, and discovering relationships between gene modules and disease traits. Addressing these points would make the perspective more informative and engaging for readers interested in personalized medicine and the use of biological networks, machine learning, and artificial intelligence in disease research. However, it is important to carefully validate and interpret the results of the network-based analysis to ensure that they are biologically meaningful and clinically relevant (Sonawane et al., 2019; Galindez et al., 2023). Therefore, the clinical application of precision medicine will likely require a fusion of approaches tailored to each clinical problem (Duffy, 2016).

Individualized networks provide a powerful data integration and analysis paradigm, offering a systems-level understanding of disease mechanisms and underlying causes (Furlong, 2013; McGillivray et al., 2018). Combining biomedical data with appropriate network modeling approaches makes it possible to derive disease-associated information and outcomes, including biomarkers, therapeutic targets, phenotype-specific genes, survival prediction, and interactions between molecules and disease subtypes (Sonawane et al., 2019). An emergent area known as Network Medicine (Loscalzo, 2019), these approaches have allowed the stratification of cancer into subtypes predictive of clinical outcomes, such as response to therapy, patient survival, and tumor histology (Hofree et al., 2013). However, there are limitations to network-based approaches for precision medicine, such as accounting for patient heterogeneity and variability and constructing appropriate network models that depend on study design, molecular entities measured, and the type and size of data (Sonawane et al., 2019). The field should strive to integrate genomic and clinical data to build networks that detect differences for each sample. This new avenue will allow us to classify complex diseases into clinically and biologically homogeneous subtypes, leading to a better understanding of disease pathophysiology and developing more targeted interventions (Sørlie et al., 2001). By employing computational and systems biology applications to develop individualized protocols, it is possible to minimize patient suffering while maximizing treatment effectiveness, allowing for the progression of precision medicine and exploring differences between individuals (Barh et al., 2020).

The advantage of individualized protocols seen from the network paradigm over other strategies is that we can study one network per sample, make identification of modules in each network, compare patients by comparing their respective networks, cluster individuals based on sample-specific networks, and associate networks (sub-) structure to disease status (more detailed in Table 1).

1.2 Approaches for generating individualized networks

Nonetheless, it is possible to identify pathways and further elucidate the molecular mechanisms of disease for individual patients using biological systems strategies. Evaluating correlations or other quantitative measures between molecules for each individual, which are usually unavailable in clinical practice, is the goal of the individualized network approach. However, this requirement for molecular data seriously limits the application of this methodology in personalized medicine (Galindez et al., 2023). Recently, several authors have developed new strategies to infer networks at the individual level, which can facilitate the discovery of differentiated disease modules or different candidate mechanisms. Although the traditional aggregated or averaged networks have allowed us to gain important insights across a wide range of biological systems and diseases, they only capture processes shared across a population of samples (Figure 1). Therefore, individualized network approaches have the potential to advance precision medicine by enabling the identification of molecular pathways that underlie complex disease phenotypes (Van Der Wijst et al., 2018; Galindez et al., 2023).

Each of the individualized networks is representative of the wiring of a specific individual and can characterize the specific disease state of an individual, as opposed to more traditional methods in which the network represents a population or cohort (Sonawane et al., 2019). Moreover, several approaches have been

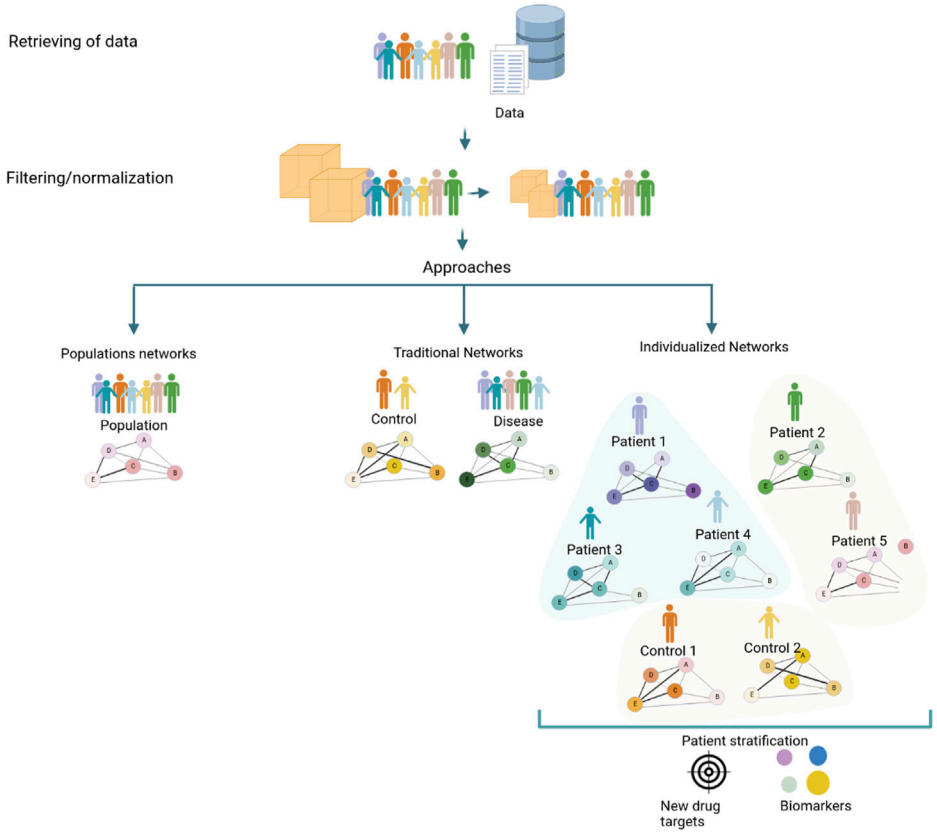


FIGURE 1
Strategies to generate a coexpression network using a conventional approach that implies a population network, a traditional (control/diseases) network, and the new individualized coexpression approach. The network generation process to generate networks with different approaches consists of a series of steps: obtention of data from patients, clinics, and/or databases, normalizing data, and filtering features for ameliorating inconsistencies. Strategies commonly employed in studies of diseases through networks, population, and traditional (case and control) networks consider mean values of populations that limit known processes that can occur in unique patients; for this reason, individualized networks between genes in samples could trigger give knowledge about changes at the level of pathways associated with diseases, with the potential to discover new drug targets and biomarkers.

suggested for exploring sample-level network information (Zanin et al., 2018; Liu et al., 2016; Kuijjer et al., 2019; Dai et al., 2019; Campos-Laborie et al., 2019; X. et al., 2021) (summarized in Table 2). Furthermore, several authors focus on single-cell analysis due to the sparsity and heterogeneity of transcript counts. Authors such as (Liu et al., 2016; Liu et al., 2016; Dai et al., 2019; Dai et al., 2019) used individualized network strategies to study scRNA-seq heterogeneity in different cell types present in the same sample (R.-S. et al., 2023). These methods can also be applied similarly to construct individual networks of each bulk RNA-seq patient data sample. However, there are potential challenges and limitations in multi-omics network medicine approaches, and the clinical application of precision medicine will likely require a fusion of approaches tailored to each clinical problem (Duffy, 2016; Sonawane et al., 2019). To use knowledge of individualized biological co-expression networks in clinical settings its necessary collect individual-level data, construct and analyze co-expression networks to detect disease-relevant gene clusters and identify personalized biomarkers and therapeutic targets (Harikumar et al., 2021). This analysis can guide the selection of personalized therapies, leading to improved treatment outcomes and reduced side effects. Therefore, it is important to carefully validate and interpret the results of individualized network approaches to ensure that they are biologically meaningful and clinically relevant (Galindez et al., 2023).

1.3 The potential of individualized gene networks in personalized medicine

Individualized gene networks have emerged as valuable tools for personalized medicine, allowing for identifying disease-associated biomarkers with diagnostic and prognostic value (Emmert-Streib et al., 2014). By unraveling molecular interactions, these networks enhance the accuracy and timeliness of disease diagnosis and facilitate the selection of more effective treatment options. Furthermore, specific network-building strategies enable the prediction of individual drug responses, minimizing exposure to ineffective drugs and reducing side effects (Van Der Wijst et al., 2018). Individualized networks also reveal novel therapeutic targets specific to each patient's genetic and molecular profile, paving the way for precise and effective therapies (Yan et al., 2022). Integrating genetic, environmental, and lifestyle factors into personalized gene

TABLE 2 Summary of sample-specific methods.

Method	Type of network (nodes/edges)	Context
Convergence/divergence network creation Zanin, Tuñas, and Menasalvas. (2018)	Nodes correspond to the study subjects. Weight is further associated with the link between two nodes representing the distance between their features	Works assume that each disease is characterized by a high internal coherence (or homogeneity), but they explore the opposite possibility in this work
Sample specific network Zanin, Tuñas, and Menasalvas. (2018); Liu et al. (2016); Kuijjer et al. (2019); Dai et al. (2019); Campos-Laborie et al. (2019); Wang, Choi, and Roeder. (2021)	Nodes correspond to genes. Edge represents the distance between their genes	They developed a statistical method that allows constructing of individual-specific networks based on molecular expressions of a single sample to characterize various human diseases at a network level
LIONESS (Linear Interpolation to Obtain Network Estimates for Single Samples) Zanin, Tuñas, and Menasalvas. (2018); Liu et al. (2016); Kuijjer et al. (2019); Dai et al. (2019); Campos-Laborie et al. (2019); Wang, Choi, and Roeder. (2021)	Model regulatory network in individual samples. Network in which "nodes" represent genes and "edges" represent a single estimate for the likelihood of interaction between those genes	Aggregate or traditional network models fail to capture population heterogeneity. They propose a method to reverse engineer *sample-specific* networks from aggregate networks. They used these networks to study changes in network topology across time and to characterize shifts in gene regulation using linear interpolation to the predictions made by existing aggregate network inference approaches
Cell-specific network Zanin, Tuñas, and Menasalvas. (2018); Liu et al. (2016); Kuijjer et al. (2019); Dai et al. (2019); Campos-Laborie et al. (2019); Wang, Choi, and Roeder. (2021)	Nodes are genes and edges are gene–gene associations, based on statistical dependency	This method transforms the data from 'unstable' gene expression form to 'stable' gene association form on a single-cell basis to obtain a network for one cell from scRNA-seq data. This method can find differential gene associations for every single cell. Traditional differential gene expression analyses ignore even 'dark' genes that play important roles at the network level. And can be applied to construct an individual network of each sample bulk RNA-seq data
locCSN Zanin, Tuñas, and Menasalvas. (2018); Liu et al. (2016); Kuijjer et al. (2019); Dai et al. (2019); Campos-Laborie et al. (2019); Wang, Choi, and Roeder. (2021)	Nodes are genes, and edges are gene–gene associations	They develop an approach that estimates cell-specific networks for each cell, preserving information about cellular heterogeneity that is lost with other approaches

regulatory networks empowers healthcare providers to predict disease risk in susceptible individuals and implement early, personalized preventive measures (Van Der Wijst et al., 2018). Moreover, studying gene networks in individual cells enables the identification of molecular markers that predict disease progression and treatment response, enabling personalized treatment and real-time therapy monitoring (Emmert-Streib et al., 2014). These advancements in personalized medicine are crucial for understanding the genetic basis of common diseases and discovering new treatments and therapies (Ahmed et al., 2020).

Network individualization significantly impacts clinical applications, treatments, medications, and omics exams, contributing to more accurate and effective medical care in personalized medicine (Infante et al., 2020). Here are some ways individualization can improve patient care:

1.3.1 Personalized treatments

Understanding a patient's genetic and molecular characteristics enables doctors to design tailored treatments, including selecting specific medications, dosage adjustments, and identifying the most effective combination therapies (Suwinski et al., 2019).

1.3.2 Safer medications and therapies

Individualization helps identify patients more likely to experience side effects or adverse reactions to certain medications. By better understanding the molecular interaction networks within individual patients, personalized therapeutic targets can be identified, leading to more effective and safer treatments (Goetz and Schork, 2018).

1.3.3 Personalized omics exams

Performing omics exams, such as whole genome sequencing, gene expression profiling, and protein analysis, individually provides accurate and relevant data for guiding diagnosis, prognosis, and treatment (Mathur and Sutton, 2017; Ahmed et al., 2020; Williams et al., 2022).

1.3.4 Early diagnosis of genetic diseases

Individualized medicine enables omics tests, such as genome sequencing, to identify specific genetic mutations associated with diseases, allowing for accurate and early diagnosis of genetic disorders and a better understanding of genetic predisposition (Aspinall and Hamermesh, 2007).

1.3.5 Facilitating drug approval

By considering patients' genetic and molecular characteristics, individualization can identify specific subgroups that may benefit more from certain drugs, expediting the drug approval process and providing access to more effective treatments for selected patients (FDA, 2022).

2 Challenges and perspectives of using individualized networks in precision medicine

The challenges of using individualized networks in precision medicine include the requirement for molecular data, which is usually unavailable in clinical practice, and the need to develop

new strategies to infer networks at the individual level (Van Der Wijst et al., 2018; R.-S. et al., 2023). The clinical application of precision medicine will likely require a fusion of approaches tailored to each clinical problem, which can be complex and require significant computational resources (Duffy, 2016). Furthermore, the statistical rigor of network predictions comes from the study design and the size of the datasets, which can be a limitation (Galindez et al., 2023). Current approaches may need more samples to infer coexpression networks that accurately capture the complexity of individualized networks. The search space of possible coexpression networks is vast and decreased uncertainty and reduced statistical power due to the small sample size may limit the generalizability of the constructed networks (Liesecke et al., 2019).

Obtaining many samples with comprehensive genomic data can be challenging, especially for rare diseases or specific patient populations. With limited sample sizes, the statistical power to detect meaningful coexpression relationships may be reduced, leading to false positives or missing important connections. One approach to address these limitations is leveraging existing knowledge from larger datasets or databases, incorporating prior knowledge about gene-gene interactions, regulatory relationships, or functional annotations. Integrating multi-omics data from different modalities (e.g., genomics, transcriptomics, proteomics) could provide a more comprehensive view of individual-specific networks. Collaboration among researchers and data sharing can help increase sample sizes and improve the statistical power of coexpression network inference (Escorcia-Rodríguez et al., 2023). The development of novel statistical methods specifically designed for analyzing individualized coexpression networks can improve the accuracy and reliability of the inferred networks (Yu et al., 2018).

Finally, developing more sophisticated algorithms and computational methods can help extract meaningful information from smaller sample sizes and incorporate prior knowledge, improving the accuracy and robustness of individualized coexpression networks (Colby et al., 2018). For example, Liesecke et al. proposed the idea of conserved coexpression links between two genes over several datasets, reinforcing the coexpression relationship (Liesecke et al., 2019). However, there are still challenges to overcome. When merging expression data, the size increase should outweigh the noise inclusion, and graph structure should be considered when integrating the inferences (Escorcia-Rodríguez et al., 2023). The potential bias introduced by relying on external datasets should also be considered, as they may only partially represent the specific biological context of the individual sample. Moreover, methods inferring coexpression networks should no longer be assessed solely based on standard performance metrics and graph structural properties.

Overall, while individualized networks have the potential to advance precision medicine, they require careful validation and interpretation of results to ensure they are biologically meaningful and clinically relevant. For other hand, the cost of using transcriptomic data has decreased over time, making it more accessible for researchers and clinicians, and it is important to consider the potential benefits of, and funding opportunities for research in personalized medicine; for this reason, it is addressing these challenges and limitations is crucial for their success and from a perspective. Stratification makes possible the design of new clinical trials to reevaluate previously tested drugs without such stratification and determine possible new therapies or treatments for each molecular subtype of patients (Rajewsky et al., 2020).

3 Conclusion

Personalized medicine, with its focus on individualized medical treatment based on patient characteristics, has the potential to revolutionize healthcare by improving patient outcomes and enhancing the quality of care. Developing individualized therapy protocols considering patient heterogeneity can minimize patient suffering while maximizing treatment effectiveness; this necessitates the refinement of disease categorization to understand the biological differences among subtypes better and guide personalized treatment strategies.

Novel individualized gene coexpression networks offer a paradigm shift in studying complex diseases by revealing patient-specific gene expression patterns and modules. By integrating multimodal information and considering patient-specific characteristics, these networks enhance our understanding of disease pathogenesis, treatment response, and diagnostic accuracy. They provide a more comprehensive understanding of complex diseases, refine disease subtyping, and guide the selection of personalized treatment strategies to improve patient outcomes.

Network medicine, which integrates diverse biological networks, is emerging as a powerful approach to offer a systems-level understanding of disease mechanisms and underlying causes. By analyzing gene-gene interactions in individual samples and systematically comparing them, we can identify pathways, subtypes of disease states, and key components in the networks that can be targeted in clinical practice. Multiscale mathematical and computational tools and integrating genomic and clinical data enable the construction of individualized networks with single-individual resolution.

While the potential impact of individualized coexpression networks on clinical practice is significant, further research and interdisciplinary collaboration are needed to realize their transformative powerfully. Standardization and robustness of data-gathering approaches, including imaging, multi-omic approaches, and clinical information, are critical for scalability to larger patient cohorts. Deep-learning-based data integration models hold promise in accurately predicting disease progression and identifying disease subtypes by leveraging multimodal data from various sources.

Addressing the limitations of current approaches to infer coexpression networks requires leveraging existing knowledge, integrating multi-omics data, collaborative efforts among researchers, and developing novel statistical methods and improved algorithms. These potential solutions represent promising directions for overcoming current limitations and advancing the inference of individualized coexpression networks.

In conclusion, individualized coexpression networks have the potential to significantly advance our knowledge of complex diseases, refine disease subtyping, and guide the selection of personalized treatment strategies. By integrating diverse biological networks and considering patient-specific characteristics, these networks enhance our understanding of disease mechanisms and improve patient outcomes in the era of precision medicine. As we continue to explore the transformative potential of network medicine, interdisciplinary collaboration, further research, and methodological

advancements are vital to fully harness the power of individualized coexpression networks and improve healthcare outcomes for patients.

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 authors.

Author contributions

VL, AM, MS, and IP: conceptualization. IP and VL: methodology. VL and AM: writing and original draft preparation. VL and AM: writing, review, and editing. VL, AM, MS, and IP: supervision and funding resources. All authors contributed to the article and approved the submitted version.

Funding

This research has been financed mainly by ANID Doctoral Fellowship 21181311 and FONDECYT Inicio 11171015, and Centro Ciencia & Vida, FB210008, Financiamiento Basal para Centros Cientificos y Tecnológicos de Excelencia de ANID.

Acknowledgments

VL gratefully acknowledges ANID, Chile, for Ph.D. fellowship 21181311. PoweredNLHPC (ECM-02): this research was partially supported by the supercomputing infrastructure of the NLHPC (ECM-02). Figures were created with BioRender.com. This publication had the support of the Vicerrectoría de Investigación y Doctorados of Universidad San Sebastián–Fondo VRID_APC23/11.

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.

References

Agrawal, M., Zitnik, M., and Leskovec, J. (2018). Large-scale analysis of disease pathways in the human interactome. *Pac. Symposium Biocomput.* 23, 111–122. doi:10.1142/9789813235533_0011

Ahmed, Z., Zeeshan, S., Mendhe, D., and Dong, X. (2020). Human gene and disease associations for clinical-genomics and precision medicine research. *Clin. Transl. Med.* 10 (1), 297–318. doi:10.1002/ctm2.28

Ahn, A. C., Tewari, M., Poon, C-S., and Phillips, R. S. (2006). The limits of reductionism in medicine: could systems biology offer an alternative? *PLoS Med.* 3 (6), e208. doi:10.1371/journal.pmed.0030208

Aspinall, M. G., and Hamermesh, R. G. (2007). Realizing the promise of personalized medicine. *Harv. Bus. Rev.* 85 (10), 108–117.

Barh, D., Ch Yiannakopoulou, E., Salawu, E. O., Bhattacharjee, A., Chowbina, S., Nalluri, J. J., et al. (2020). *In silico* disease model: from simple networks to complex diseases. *Anim. Biotechnol.* 441. doi:10.1016/B978-0-12-811710-1.00020-3

Browne, F., Wang, H., Zheng, H., and Azuaje, F. (2009). Grip: a web-based system for constructing gold standard datasets for protein-protein interaction prediction. *Source Code Biol. Med.* 4 (1), 2. doi:10.1186/1751-0473-4-2

Büttner, R., Longshore, J. W., López-Ríos, F., Merkelbach-Bruse, S., Normanno, N., Rouleau, E., et al. (2019). Implementing TMB measurement in clinical practice: considerations on assay requirements. *ESMO Open* 4 (1), e000442. doi:10.1136/esmoopen-2018-000442

Campos-Laborie, F. J., Risueño, A., Ortiz-Estévez, M., Rosón-Burgo, B., Droste, C., Fontanillo, C., et al. (2019). Deco: decompose heterogeneous population cohorts for patient stratification and discovery of sample biomarkers using omic data profiling. *Bioinformatics* 35 (19), 3651–3662. doi:10.1093/bioinformatics/btz148

Chan, S. Y., and Joseph, L. (2012). The emerging paradigm of network medicine in the study of human disease. *Circulation Res.* 111 (3), 359–374. doi:10.1161/CIRCRESAHA.111.258541

Charitou, T., Bryan, K., and Lynn, D. J. (2016). Using biological networks to integrate, visualize and analyze genomics data. *Genet. Sel. Evol. GSE* 48 (3), 27. doi:10.1186/s12711-016-0205-1

Colby, S. M., Ryan, S. M., Overall, C. C., Renslow, R. S., McDermott, J. E., Renslow, R. S., et al. (2018). Improving network inference algorithms using resampling methods. *BMC Bioinforma.* 19 (1), 376. doi:10.1186/s12859-018-2402-0

Dai, H., Lin, L., Zeng, T., and Chen, L. (2019). Cell-specific network constructed by single-cell RNA sequencing data. *Nucleic Acids Res.* 47 (11), e62. doi:10.1093/nar/gkz172

Devi, G., and Scheltens, P. (2018). Heterogeneity of alzheimer's disease: consequence for drug trials? *Alzheimer's Res. Ther.* 10 (1), 122. doi:10.1186/s13195-018-0455-y

Duffy, David J. (2016). Problems, challenges and promises: perspectives on precision medicine. *Briefings Bioinforma.* 17 (3), 494–504. doi:10.1093/bib/bbv060

Emmert-Streib, F., Dehmer, M., and Haibe-Kains, B. (2014). Gene regulatory networks and their applications: understanding biological and medical problems in terms of networks. *Front. Cell Dev. Biol.* 2 (8), 38. doi:10.3389/fcell.2014.00038

Escorcia-Rodríguez, J., Juan, M., Gaytan-Nuñez, E., Hernandez-Benitez, E. M., Zorro-Aranda, A., Tello-Palencia, M. A., et al. (2023). Improving gene regulatory network inference and assessment: the importance of using network structure. *Front. Genet.* 14 (2), 1143382. doi:10.3389/fgene.2023.1143382

FDA (2022). Focus area: individualized therapeutics and precision medicine. September 6, 2022 Available at: www.fda.gov/science-research/focus-areas-regulatory-science-report/focus-area-individualized-therapeutics-and-precision-medicine (Accessed November 29, 2022).

Freudenberg-Hua, Y., Li, W., and Davies, P. (2018). The role of genetics in advancing precision medicine for alzheimer's disease-A narrative review. *Front. Med.* 5 (4), 108. doi:10.3389/fmed.2018.00108

Fröhlich, H., Balling, R., Beerenwinkel, N., Kohlbacher, O., Kumar, S., Lengauer, T., et al. (2018). From hype to reality: data science enabling personalized medicine. *BMC Med.* 16, 150. doi:10.1186/s12916-018-1122-7

Furlong, L. I. (2013). Human diseases through the lens of network biology. *Trends Genet. TIG* 29 (3), 150–159. doi:10.1016/j.tig.2012.11.004

Galindez, G., Sadegh, S., Jan, B., Kacprowski, T., and List, M. (2023). Network-based approaches for modeling disease regulation and progression. *Comput. Struct. Biotechnol. J.* 21 (1), 780–795. doi:10.1016/j.csbj.2022.12.022

Goetz, L. H., and Schork, N. J. (2018). Personalized medicine: motivation, challenges, and progress. *Fertil. Steril.* 109 (6), 952–963. doi:10.1016/j.fertnstert.2018.05.006

Gurdasani, D., Barroso, I., Zeggini, E., and Sandhu, M. S. (2019). Genomics of disease risk in globally diverse populations. *Nat. Rev. Genet.* 20 (9), 520–535. doi:10.1038/s41576-019-0144-0

Harikumar, H., Quinn, T. P., Rana, S., Gupta, S., and Venkatesh, S. (2021). Personalized single-cell networks: a framework to predict the response of any gene to any drug for any patient. *BioData Min.* 14 (1), 37. doi:10.1186/s13040-021-00263-w

Hofree, M., Shen, J. P., Carter, H., Gross, A., and Ideker, T. (2013). Network-based stratification of tumor mutations. *Nat. Methods* 10 (11), 1108–1115. doi:10.1038/nmeth.2651

Infante, T., Del Viscovo, L., Luisa De Rimini, M., Padula, S., Caso, P., and Napoli, C. (2020). Network medicine: a clinical approach for precision medicine and personalized therapy in coronary heart disease. *J. Atheroscler. Thrombosis* 27 (4), 279–302. doi:10.5551/jat.52407

Kauffman, A.Member of the Santa Fe Institute and Professor of Biochemistry Stuart A Kauffman (1993). *The origins of order: Self-organization and selection in evolution*. USA: Oxford University Press.

Kessler, M. D., Yerges-Armstrong, L., Taub, M. A., Shetty, A. C., Maloney, K., Linda Jo, B. J., et al. (2016). Challenges and disparities in the application of personalized genomic medicine to populations with african ancestry. *Nat. Commun.* 7 (1), 12521–12528. doi:10.1038/ncomms12521

Khurana, E., Fu, Y., Chen, J., and Gerstein, M. (2013). Interpretation of genomic variants using a unified biological network approach. *PLoS Comput. Biol.* 9 (3), e1002886. doi:10.1371/journal.pcbi.1002886

Koníčková, D., Menšíková, K., Tučková, L., Hényková, E., Strnad, M., Friedecký, D., et al. (2022). Biomarkers of neurodegenerative diseases: biology, taxonomy, clinical relevance, and current research status. *Biomedicines* 10 (7), 1760. doi:10.3390/biomedicines10071760

Kuijjer, M. L, Tung, M. G., Yuan, G., Quackenbush, J., and Glass, K. (2019). Estimating sample-specific regulatory networks. *iScience* 14 (4), 226–240. doi:10.1016/j.isci.2019.03.021

Lengerich, B. J., Aragam, B., and Xing, E. P. (2018). Personalized regression enables sample-specific pan-cancer analysis. *Bioinformatics* 34, i178–i186. doi:10.1093/bioinformatics/bty250

Liesecke, F., Johan-Owen, D. C., Besseau, S., Vincent, C., Clastre, M., Vergès, V., et al. (2019). Improved gene Co-expression network quality through expression dataset down-sampling and network aggregation. *Sci. Rep.* 9 (1), 14431–14516. doi:10.1038/s41598-019-50885-8

Liu, X., Wang, Y., Ji, H., Aihara, K., and Chen, L. (2016). Personalized characterization of diseases using sample-specific networks. *Nucleic Acids Res.* 44 (22), e164. doi:10.1093/nar/gkw772

Lombardo, M. V., Lai, M-C., and Baron-Cohen, S. (2019). Big data approaches to decomposing heterogeneity across the autism spectrum. *Mol. Psychiatry* 24 (10), 1435–1450. doi:10.1038/s41380-018-0321-0

Loscalzo, J. (2019). Network medicine and type 2 diabetes mellitus: insights into disease mechanism and guide to precision medicine. *Endocrine* 66 (3), 456–459. doi:10.1007/s12020-019-02042-4

Ma, J., Ku Yu, M., Fong, S., Ono, K., Sage, E., Demchak, B., et al. (2018). Using deep learning to model the hierarchical structure and function of a cell. *Nat. Methods* 15 (4), 290–298. doi:10.1038/nmeth.4627

Mathur, S., and Sutton, J. (2017). Personalized medicine could transform healthcare. *Biomed. Rep.* 7 (1), 3–5. doi:10.3892/br.2017.922

McGillivray, P., Clarke, D., Meyerson, W., Zhang, J., Lee, D., Gu, M., et al. (2018). Network analysis as a grand unifier in biomedical data science. *Annu. Rev. Biomed. Data Sci.* 1 (1), 153–180. doi:10.1146/annurev-biodatasci-080917-013444

Mulder, N. J., Akinola, R. O., Mazandu, G. K., and Rapanoel., H. (2014). Using biological networks to improve our understanding of infectious diseases. *Comput. Struct. Biotechnol. J.* 11 (18), 1–10. doi:10.1016/j.csbj.2014.08.006

Popejoy, A. B., and Fullerton, S. M. (2016). Genomics is failing on diversity. *Nat. Publ. Group U. K.* 538, 161–164. October 12, 2016. doi:10.1038/538161a

Popejoy, A. B., Ritter, D. I., Crooks, K., Currey, E., Fullerton, S. M., Hindorff, L. A., et al. (2018). The clinical imperative for inclusivity: race, ethnicity, and ancestry (REA) in genomics. *Hum. Mutat.* 39 (11), 1713–1720. doi:10.1002/humu.23644

Prosperi, M., Min, J. S., Jiang, B., and Modave, F. (2018). Big data hurdles in precision medicine and precision public health. *BMC Med. Inf. Decis. Mak.* 18 (1), 139. doi:10.1186/s12911-018-0719-2

Rajewsky, N., Almouzni, G., Gorski, S. A., Aerts, S., Amit, I., Bertero, M. G., et al. (2020). LifeTime and improving European healthcare through cell-based interceptive medicine. *Nature* 587 (7834), 377–386. doi:10.1038/s41586-020-2715-9

Rouzier, R., Perou, C. M., Fraser Symmans, W., Ibrahim, N., Cristofanilli, M., Anderson, K., et al. (2005). Breast cancer molecular subtypes respond differently to preoperative chemotherapy. *Clin. Cancer Res. Official J. Am. Assoc. Cancer Res.* 11 (16), 5678–5685. doi:10.1158/1078-0432.CCR-04-2421

Shipitsin, M., Campbell, L. L., Argani, P., Weremowicz, S., Bloushtain-Qimron, N., Yao, J., et al. (2007). Molecular definition of breast tumor heterogeneity. *Cancer Cell* 11 (3), 259–273. doi:10.1016/j.ccr.2007.01.013

Sierksma, A., Escott-Price, V., and De Strooper, B. (2020). Translating genetic risk of alzheimer's disease into mechanistic insight and drug targets. *Science* 370 (6512), 61–66. doi:10.1126/science.abb8575

Smith, G. D. (2011). Epidemiology, epigenetics and the 'gloomy prospect': embracing randomness in population health research and practice. *Int. J. Epidemiol.* 40, 537–562. doi:10.1093/ije/dyr117

Sonawane, A. R., Weiss, S. T., Glass, K., and Sharma, A. (2019). Network medicine in the age of biomedical big data. *Front. Genet.* 10 (4), 294. doi:10.3389/fgene.2019.00294

Sørlie, T., Perou, C. M., Tibshirani, R., Aas, T., Geisler, S., Johnsen, H., et al. (2001). Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. *Proc. Natl. Acad. Sci. U. S. A.* 98 (19), 10869–10874. doi:10.1073/pnas.191367098

Stewart, I., and Cohen, J. (2000). *The collapse of chaos: Discovering simplicity in a complex world*. Penguin UK: Penguin Books Ltd.

Suwinski, P., Ong, C., Maurice, H. T. L., Yang, M. P., Khan, A. M., and Ong, H. S. (2019). Advancing personalized medicine through the application of whole exome sequencing and big data analytics. *Front. Genet.* 10 (2), 49. doi:10.3389/fgene.2019.00049

Van Der WijstMonique, G. P., De Vries, D. H., Brugge, H., Jan, W., and Franke, L. (2018). An integrative approach for building personalized gene regulatory networks for precision medicine. *Genome Med. Dec.* 10, 96. doi:10.1186/s13073-018-0608-4

Wallstrom, G., Anderson, K. S., and LaBaer, J. (2013). Biomarker discovery for heterogeneous diseases. *Cancer Epidemiol. Biomarkers Prev.* 22 (5), 747–755. A Publication of the American Association for Cancer Research, Cosponsored by the American Society of Preventive Oncology. doi:10.1158/1055-9965.EPI-12-1236

Wang, R-S., Maron, B. A., and Joseph, L. (2023). Multiomics network medicine approaches to precision medicine and therapeutics in cardiovascular diseases. *Arteriosclerosis, Thrombosis, Vasc. Biol.* 43 (4), 493–503. doi:10.1161/ATVBAHA.122.318731

Wang, X., Choi, D., and Roeder, K. (2021). Constructing local cell-specific networks from single-cell data. *Proc. Natl. Acad. Sci.* 118 (51), e2113178118. doi:10.1073/pnas.2113178118

Williams, C. G., Lee, H. J., Asatsuma, T., Vento-Tormo, R., and Haque, A. (2022). An introduction to spatial transcriptomics for biomedical research. *Genome Med.* 14 (1), 68. doi:10.1186/s13073-022-01075-1

Yan, J., Hu, Z., Li, Z-W., Sun, S., and Guo, W-F. (2022). Network control models with personalized genomics data for understanding tumor heterogeneity in cancer. *Front. Oncol.* 12 (5), 891676. doi:10.3389/fonc.2022.891676

Yin, L., Chau, C. K. L., Sham, P-C., and So, H-C. (2019). Integrating clinical data and imputed transcriptome from GWAS to uncover complex disease subtypes: applications in psychiatry and cardiology. *Am. J. Hum. Genet.* 105 (6), 1193–1212. doi:10.1016/j.ajhg.2019.10.012

Younesi, E., and Hofmann-Apitius, M. (2013). From integrative disease modeling to predictive, preventive, personalized and participatory (P4) medicine. *EPMA J.* 4 (1), 23. doi:10.1186/1878-5085-4-23

Yu, H., Jiao, B., Lu, Lu, Wang, P., Chen, S., Liang, C., et al. (2018). NetMiner-an ensemble pipeline for building genome-wide and high-quality gene Co-expression network using massive-scale RNA-seq samples. *PloS One* 13 (2), e0192613. doi:10.1371/journal.pone.0192613

Zanin, M., Juan Manuel, T., and Menasalvas, E. (2018). Understanding diseases as increased heterogeneity: a complex network computational framework. *J. R. Soc. Interface/R. Soc.* 15 (145), 20180405. doi:10.1098/rsif.2018.0405

Zhang, H., Klareskog, L., Matussek, A., Pfister, S. M., and Benson, M. (2019). Translating genomic medicine to the clinic: challenges and opportunities. *Genome Med.* 11 (1), 9. doi:10.1186/s13073-019-0622-1

Zhang, W., Chien, J., Jeongsik, Y., and Kuang, R. (2017). Network-based machine learning and graph theory algorithms for precision oncology. *Npj Precis. Oncol.* 1 (1), 1–15. doi:10.1038/s41698-017-0029-7

Zhang, X-M., Liang, L., Liu, L., and Tang, M-J. (2021). Graph neural networks and their current applications in bioinformatics. *Front. Genet.* 12 (7), 690049. doi:10.3389/fgene.2021.690049

Zhou, K., Bhagya, S. K., Seeya, A. M., Zhang, Z., Draghici, S., and Arslanturk, S. (2022). Integration of multimodal data from disparate sources for identifying disease subtypes. *Biology* 11 (3), 360. doi:10.3390/biology11030360

Zitnik, M., Agrawal, M., and Leskovec, J. (2018). Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics* 34 (13), i457–i466. doi:10.1093/bioinformatics/bty294

Zitnik, M., and Leskovec, J. (2017). Predicting multicellular function through multi-layer tissue networks. *Bioinformatics* 33 (14), i190–i198. doi:10.1093/bioinformatics/btx252

frontiers | Frontiers in Genetics

TYPE Correction
PUBLISHED 21 September 2023
DOI 10.3389/fgene.2023.1286081

OPEN ACCESS

APPROVED BY
Frontiers Editorial Office,
Frontiers Media SA, Switzerland

*CORRESPONDENCE
Inti Pedroso,
intipedroso@gmail.com
Alberto J. M. Martin,
alberto.martin@uss.cl

RECEIVED 30 August 2023
ACCEPTED 31 August 2023
PUBLISHED 21 September 2023

CITATION
Latapiat V, Saez M, Pedroso I and Martin AJM (2023), Corrigendum: Unraveling patient heterogeneity in complex diseases through individualized co-expression networks: a perspective. *Front. Genet.* 14:1286081. doi: 10.3389/fgene.2023.1286081

Corrigendum: Unraveling patient heterogeneity in complex diseases through individualized co-expression networks: a perspective

Verónica Latapiat[1,2,3], Mauricio Saez[4,5], Inti Pedroso[2]* and Alberto J. M. Martin[3,6]*

[1]Programa de Doctorado en Genómica Integrativa, Vicerrectoría de Investigación, Universidad Mayor, Santiago, Chile, [2]Vicerrectoría de Investigación, Universidad Mayor, Santiago, Chile, [3]Laboratorio de Redes Biológicas, Centro Científico y Tecnológico de Excelencia Ciencia & Vida, Fundación Ciencia & Vida, Santiago, Chile, [4]Centro de Oncología de Precisión, Facultad de Medicina y Ciencias de la Salud, Universidad Mayor, Santiago, Chile, [5]Laboratorio de Investigación en Salud de Precisión, Departamento de Procesos Diagnósticos y Evaluación, Facultad de Ciencias de la Salud, Universidad Católica de Temuco, Temuco, Chile, [6]Escuela de Ingeniería, Facultad de Ingeniería, Arquitectura y Diseño, Universidad San Sebastián, Santiago, Chile

KEYWORDS

personalized medicine, omics, transcriptomic, co-expression, networks, diseases

A Corrigendum on
Unraveling patient heterogeneity in complex diseases through individualized co-expression networks: a perspective

by Latapiat V, Saez M, Pedroso I and Martin AJM (2023). Front. Genet. 14:1209416. doi: 10.3389/fgene.2023.1209416

In the published article, there was an error in **Affiliation** 3. Instead of "Laboratorio de Redes Biológicas, Centro Científicoy Tecnológico de Excelencia Ciencia & Vida, Fundación Ciencia & Vida, Escuela de Ingeniería, Facultad de Ingeniería, Arquitecturay Diseño, Universidad San Sebastián, Santiago, Chile", it should be "Laboratorio de Redes Biológicas, Centro Científico y Tecnológico de Excelencia Ciencia & Vida, Fundación Ciencia & Vida, Santiago, Chile."

In the published article, there was an error regarding the **Affiliation** for Alberto J. M. Martin. As well as having affiliation(s) 3, he should also have "Escuela de Ingeniería, Facultad de Ingeniería, Arquitectura y Diseño, Universidad San Sebastián, Santiago, Chile."

In the published article, there was an error in the **Acknowledgements** statement. We unfortunately missing to acknowledge the USS funds that partially covered the article APC, thus this statement is incomplete "VL gratefully acknowledges ANID, Chile, for Ph.D. fellowship 21181311. PoweredNLHPC (ECM-02): this research was partially supported by the supercomputing infrastructure of the NLHPC (ECM-02). Figure was created with BioRender.com." The correct Funding statement appears below.

"VL gratefully acknowledges ANID, Chile, for Ph.D. fellowship 21181311. PoweredNLHPC (ECM-02): this research was partially supported by the supercomputing

infrastructure of the NLHPC (ECM-02). Figures were created with BioRender.com. This publication had the support of the Vicerrectoría de Investigación y Doctorados of Universidad San Sebastián–Fondo VRID_APC23/11."

The authors apologize for these errors and state that this does not change the scientific conclusions of the article in any way. The original article has been updated.

Publisher's note

frontiers | Frontiers in Genetics

TYPE Brief Research Report
PUBLISHED 21 August 2023
DOI 10.3389/fgene.2023.1254827

OPEN ACCESS

EDITED BY
Lei Chen,
Shanghai Maritime University, China

REVIEWED BY
Lesong Wei,
King Abdullah University of Science and Technology, Saudi Arabia
Hao Lin,
University of Electronic Science and Technology of China, China

*CORRESPONDENCE
Xin Chen,
chenxinghmu@163.com
Jie Bai,
baij@zucc.edu.cn

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

RECEIVED 07 July 2023
ACCEPTED 31 July 2023
PUBLISHED 21 August 2023

CITATION
Ju H, Bai J, Jiang J, Che Y and Chen X (2023), Comparative evaluation and analysis of DNA N4-methylcytosine methylation sites using deep learning. *Front. Genet.* 14:1254827. doi: 10.3389/fgene.2023.1254827

Comparative evaluation and analysis of DNA N4-methylcytosine methylation sites using deep learning

Hong Ju[1†], Jie Bai[2*], Jing Jiang[3†], Yusheng Che[1] and Xin Chen[4*]

[1]Heilongjiang Agricultural Engineering Vocational College, Harbin, China, [2]Engineering Research Center of Integration and Application of Digital Learning Technology, Ministry of Education, Hangzhou, China, [3]Beidahuang Industry Group General Hospital, Harbin, China, [4]Department of Neurosurgical Laboratory, The First Affiliated Hospital of Harbin Medical University, Harbin, China

DNA N4-methylcytosine (4mC) is significantly involved in biological processes, such as DNA expression, repair, and replication. Therefore, accurate prediction methods are urgently needed. Deep learning methods have transformed applications that previously require sequencing expertise into engineering challenges that do not require expertise to solve. Here, we compare a variety of state-of-the-art deep learning models on six benchmark datasets to evaluate their performance in 4mC methylation site detection. We visualize the statistical analysis of the datasets and the performance of different deep-learning models. We conclude that deep learning can greatly expand the potential of methylation site prediction.

KEYWORDS

4mC DNA methylation, deep learning, classification, feature, visualization, interpretable ability

Introduction

The rapid progress in genome sequencing technologies has facilitated the investigation of the functional effects of DNA chemical modifications with unprecedented precision (Larranaga et al., 2006; Jiao and Du, 2016; Hamdy et al., 2022). DNA methylation, as a vital epigenetic modification, plays a crucial role in normal organism development and essential biological processes (Lv et al., 2021). In the genomes of both prokaryotic and eukaryotic organisms, the most prevalent kinds of DNA methylation include N6-methyladenine (6mA) (Huang et al., 2020; Li et al., 2021; Chen et al., 2022), C5-methylcytosine (5mC) (Cao et al., 2022), and N4-methylcytosine (4mC) (Moore et al., 2013; Plongthongkum et al., 2014; Ao et al., 2022a; Zulfiqar et al., 2022a; Zulfiqar et al., 2022b). The distribution of 4mC sites in the genome is highly significant as they play a crucial role in regulating gene expression and maintaining genome stability. Accurate identification and analysis of 4mC sites allow for a deeper understanding of the role of DNA methylation in gene regulation and disease mechanisms. This has important implications for the study of epigenetics, cancer etiology, biological evolution, and potential therapeutic strategies. Therefore, the development of efficient and accurate methods for detecting and identifying 4mC sites is of great importance for understanding biological processes and disease research (Razin and Cedar, 1991; Kulis and Esteller, 2010).

Several experimental techniques have been utilized to identify epigenetic 4mC sites. These methodologies include methylation-specific PCR, mass spectrometry, 4mC-Tet-

assisted bisulfite-sequencing (4mCTABseq), whole-genome bisulfite sequencing, nanopore sequencing, and single-molecule real-time (SMRT) sequencing (Buryanov and Shevchuk, 2005; Laird, 2010; Chen et al., 2016; Chen et al., 2017; Ni et al., 2019). These experiment-based methods suffer from limitations such as low throughput, high cost, and restricted detection sensitivity. Nowadays, machine learning has been widely utilized and are successful technology in bioinformatics for extracting knowledge from huge data (Larranaga et al., 2006; Dwyer et al., 2018; Hu et al., 2020; Hu et al., 2021; Hu et al., 2022a; Zeng et al., 2022a; Zeng et al., 2022b; Li et al., 2023; Xu et al., 2023) and numerous computer techniques have been created to anticipate DNA 4mC sites. Both standard machine learning techniques and more current deep learning algorithms have been used to provide a strong result. In the field of 4mC site prediction, researchers have made significant strides by leveraging machine learning algorithms. These approaches utilize computational models to identify and classify 4mC sites within DNA sequences. Various machine learning techniques have been explored, including support vector machine (SVM) (Chen et al., 2017), random forest (RF), Markov model (MM), and ensemble methods. Additionally, advanced techniques such as extreme gradient boosting (XGBoost) and Laplacian Regularized Sparse Representation have also been employed in this context (Chen et al., 2017; Manavalan et al., 2019; He et al., 2019; Hasan et al., 2020; Zhao et al., 2020; Ao et al., 2022b; Xiao et al., 2022). However, traditional machine learning algorithms rely significantly on data representations known as features for appropriate performance, and it's tough to figure out which features are best for a certain task. Deep learning overcomes the limitations of traditional methods by offering adaptivity, fault tolerance, nonlinearity, and improved input-to-output mapping. Deep learning methods, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have been developed for the detection of 4mC sites, leveraging their ability to capture sequence patterns and dependencies, thereby contributing to accurate identification of these sites and enhancing our understanding of DNA methylation in gene regulation and epigenetics (Xu et al., 2021; Liu et al., 2022). Yet there are still many deep learning methods that have not been applied, which have achieved great success in various application scenarios, including computer vision, speech recognition, biomarker identification (Zeng et al., 2020; Cai et al., 2021) and drug discovery (Chen et al., 2021; Zhang et al., 2021; Hu et al., 2022b; Dong et al., 2022; Pan et al., 2022; Song et al., 2022).

Choosing an appropriate deep learning model for bioinformatics problems poses a significant challenge for biologists. Understanding and comparing the performance of different models on specific datasets is of paramount importance for guiding practical applications. Therefore, our research focuses on evaluating the performance of multiple deep learning models on the 4mC datasets, aiming to assist biologists in making informed decisions when selecting suitable models.

We selected several common deep learning models, including RNN (Recurrent Nerual Network) (Rumelhart et al., 1986), long short-term memory (LSTM) (Graves, 2012), bi-directional long short-term memory (Bi-LSTM) (Graves and Schmidhuber, 2005; Sharma and Srivastava, 2021), text convolutional neural network (Text-CNN) (Kim, 2014), and bidirectional encoder representations from transformers (BERT) (Ji et al., 2021; Tran and Nguyen, 2022), and compared their performances on the 4mC datasets through optimization of model hyperparameters. Our research findings provide strong evidence-based support for biologists, aiding them in making informed choices when addressing bioinformatics problems on the 4mC datasets. By comparing the performance of multiple models, we can offer recommendations tailored to different problems and datasets, enabling biologists to better understand and leverage the advantages of deep learning models.

Materials and methods

The implementation of our experiments relies on the DeepBIO (Wang et al., 2022) platform, which provides a wide selection of deep learning models and a visual comparison of multiple models. Figure 1 illustrates the overall framework of our works. We selected four deep learning models (RNN, LSTM, Bi-LSTM, Text-CNN) and pre-trained BERT models from the DeepBIO platform, and BERT is used as our main method to compare with other methods.

Datasets

The first step in creating a strong and trustworthy classification model is creating high-quality benchmark datasets. In this study, six benchmark datasets were utilized (Yu et al., 2021). Table 1 provides a statistical summary of the datasets. The positive samples consisted of sequences that were 41 base pairs (bp) in length and contained a 4mC (4-methylcytosine) site located in the middle. These datasets have undergone rigorous preprocessing and quality control measures to ensure data accuracy and consistency (Jin et al., 2022). By training and evaluating the model on data from multiple species, including humans, animals, and plants, we ensure its broad applicability and provide valuable insights for biologists in selecting deep learning models.

Input feature matrix

Deep learning algorithms possess the capability to autonomously extract valuable features from data, distinguishing them from conventional machine learning methods that necessitate manual feature engineering. Nonetheless, when dealing with a string of nucleotide letters (A, C, G, and T), a conversion into a matrix format is required prior to feeding it into a neural network layer. Unlike prior methods that used several feature encodings schemes to represent the sequence as the input to train the model, this method uses a single feature encoding scheme. We took the dictionary encoding approaches for representing DNA sequences. To represent DNA sequences, we utilized a dictionary encoding method where each nucleotide (A, C, G, and T) is assigned a numeric value. Specifically, A is represented by 1, C by 2, G by 3, and T by 4. This encoding scheme allows us to convert the sequence into an N-dimensional vector, facilitating its input into the neural network for further analysis.

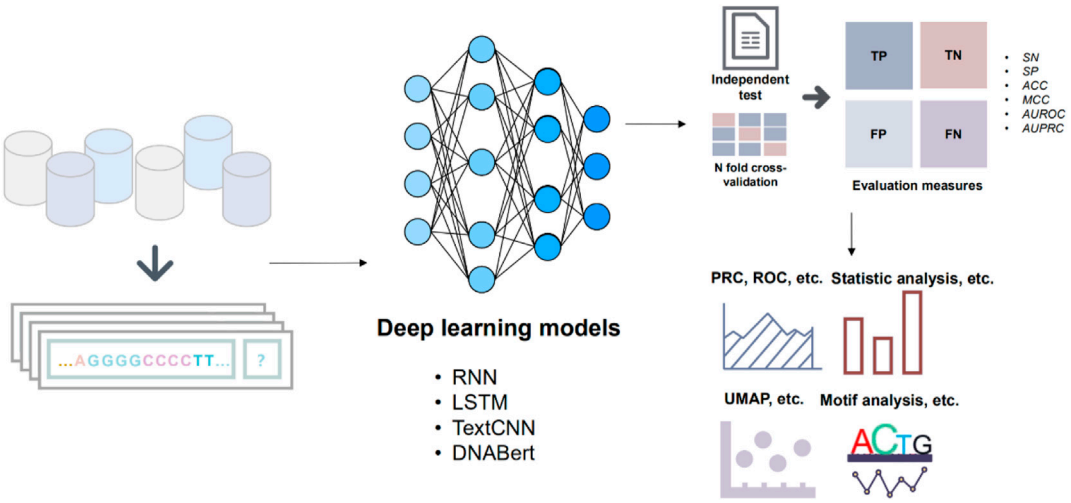


FIGURE 1
The workflow of the main modules. The benchmark datasets are initially divided into training and test sets. Subsequently, the divided dataset is fed into various deep learning models for prediction. The results of the predictions from different models are then evaluated. Finally, the data generated throughout these steps are visualized and analyzed for further insights.

TABLE 1 Statistical summary of benchmark datasets.

Species	Positive sample	Negative sample	Total
C. elegans	1,554	1,554	3,108
D. melanogaster	1,769	1,769	3,538
A. thaliana	1,978	1,978	3,956
E. coli	388	388	776
G. subterraneus	906	906	1,812
G. pickeringii	569	569	1,138

Model construction and parameters

We have selected deep-learning models that have received a lot of attention in recent years as follows: RNN, LSTM, Bi-LSTM, Text-CNN, and BERT. The first four deep learning models we used are the models provided by the DeepBIO platform with parameters already set and the BERT model we used is pre-trained DNABERT (Ji et al., 2021; Ren et al., 2022), which achieves the best performance on several DNA sequence classification tasks.

RNN is a type of neural network where the output of the previous neuron is fed back as input to the current neuron, creating temporal memory and enabling the processing of dynamic input sequences. RNNs find wide applications in various domains, including voice recognition, time series analysis, DNA sequences, and sequential data processing. One notable variant of RNNs that addresses the issue of capturing long-term dependencies is Long Short-Term Memory (LSTM). LSTM introduces a cell state that serves as a memory component, allowing the network to retain relevant information over extended periods. The forget gate in LSTM controls which information should be discarded and retained by using a sigmoid activation function. Additionally, Bidirectional LSTM (BiLSTM) processes input data in both forward and backward directions, effectively incorporating information from both past and future states. This bidirectional approach enables BiLSTM to capture intricate sequential relationships between words and sentences, making it particularly advantageous for Natural Language Processing (NLP) tasks that require contextual information from both preceding and succeeding elements in the input sequence. The RNN, LSTM, and Bi-LSTM architectures consist of stacked RNN cells, LSTM cells, and bidirectional LSTM cells, respectively. All these architectures share a similar structure, featuring 128 hidden neurons and a single layer for optimal performance. To prevent overfitting and promote generalization, a dropout rate of 0.2 was applied, and the output layer utilized sigmoid activation with a single neuron.

Text-CNN, a powerful deep learning approach for language classification tasks, such as sentiment analysis and question categorization, is a convolutional neural network tailored for text processing. The core structure comprises four layers: an embedding layer, a convolution layer, a pooling layer, and a fully connected layer. In our implementation, we set four convolutional kernel sizes (1, 2, 4, 8), and the number of convolutional kernels is uniformly set to 128. The embeddings undergo convolutional operations with a sliding kernel, producing convolutions that are subsequently downsampled through a Max Pooling layer to manage complexity and computational requirements. The scalar pooling outputs are then concatenated to form a vector representation of the input sequence. To mitigate overfitting, regularization methods, including a dropout layer with a rate of 0.2 and ReLU activation, are employed in the penultimate layer, preventing overfitting of the hidden layer.

BERT, an abbreviation for Bidirectional Encoder Representations from Transformers, originates from the Transformer architecture. In the Transformer model, every output element is intricately connected to every input element, with dynamically calculated weightings based on their

connections. BERT is a pre-trained model that benefits from its ability to learn rich contextualized representations by considering the entire input sequence during training. Our study employs the pre-trained DNABert model, which has demonstrated superior performance in several DNA sequence classification tasks. We specifically fine-tune the 6mer-BERT variant on the 4mC methylation site benchmark dataset. Fine-tuning a pre-trained model on a task-specific dataset allows us to transfer the knowledge acquired during pre-training, enabling the model to achieve state-of-the-art performance in predicting DNA 4mC methylation sites. The incorporation of BERT's pre-trained knowledge provides significant advantages, as the model has already learned from vast amounts of data and captures intricate sequence patterns and dependencies. By leveraging pre-trained models like BERT, we achieve robust and accurate predictions, even in scenarios with limited training data.

Evaluation metrics

In order to compare with previous related work, we selected the commonly used evaluation indicators comprised of accuracy (ACC), sensitivity (SN), specificity (SP), Matthews' coefficient correlation (MCC), and area under the receiver operating characteristic curve (AUC). These indicators are calculated by the following formula:

$$\begin{cases} ACC = \dfrac{TP + TN}{TP + TN + FP + FN} \\ Sensitivity = \dfrac{TP}{TP + FN} \\ Specificity = \dfrac{TN}{TN + FP} \\ MCC = \dfrac{TP \times TN - FP \times FN}{\sqrt{(TP + FN)(TP + FP)(TN + FP)(TN + FN)}} \end{cases}$$

where *TP* represents true positives, which is the number of correctly predicted positive samples; *TN* represents true negatives, which is the number of correctly predicted negative samples; *FP* represents false positives, which is the number of negative samples wrongly predicted as positive; and *FN* represents false negatives, which is the number of positive samples wrongly predicted as negative.

Experimental setup

In our experimental design, we adopted the default settings of DeepBIO for other hyperparameters. For instance, when performing data set deduplication, we limited the duplication rate to 0.8 using the CDHIT algorithm integrated in the DeepBIO platform. Furthermore, we conducted a grid search on hyperparameters such as learning rate and batch size for each model. Grid search is a method for hyperparameter tuning, where different combinations of hyperparameters are tried to determine the optimal model configuration. Such experimental settings ensure that the models achieve their maximum potential performance while maintaining the reliability, fairness, and accuracy of the experiments.

Result

In this section, we evaluate the performance of the different models and analyze the features extracted by the different models. In addition, we also compare the features learnt from deep-learning models with the traditional manual feature extraction methods applied in other studies to further demonstrate the superiority of deep learning in solving the 4mC methylation site detection problem. To ensure a balanced representation, the samples were randomly divided into training and test datasets for each species. The division was done in a ratio of 9:1, with 90% of the samples allocated to the training dataset and the remaining 10% assigned to the test dataset.

Performance evaluation of multiple models

We conducted a comprehensive performance evaluation of four different models on six datasets to assess their performance in various data environments. The evaluation process involved the use of common binary classification metrics, such as accuracy (ACC), sensitivity, specificity, area under the curve (AUC), and Matthews correlation coefficient (MCC), to provide a comprehensive understanding of the models' classification capabilities and highlight their performance differences. In addition to these metrics, we also employed receiver operating characteristic (ROC) curves and precision-recall curves (PRC) to further analyze the models' performance.

Throughout our evaluation, we observed variations in performance across different datasets. While certain models demonstrated superior predictive performance on most datasets, their performance might vary on specific datasets. As shown in Figures 2A, B, the RNN and TextCNN models exhibited promising performance on the *G. pickeringii* dataset, while DNABERT outperformed others on the *G. subterraneus* dataset. Overall, DNABERT consistently showcased superior performance across the evaluated datasets.

Furthermore, let's consider the results obtained on the *E. coli* dataset. The density distribution of prediction confidence by different deep learning models (Figure 2C) provides insights into the prediction preferences of each model. In the case of LSTM and Text-CNN, their density distribution shows a preference towards the center part of the X-axis, around 0.5. This indicates their poor binary classification ability and confusion in distinguishing between positive and negative instances. On the other hand, the density distribution for DNABERT is skewed towards the right side of the X-axis, indicating a better classification performance. This suggests that DNABERT exhibits a stronger ability to differentiate between positive and negative instances. And this is consistent with the conclusions drawn from the performance comparison in Figure 2A.

We also performed statistics on the overlap of predictions between different models for the same dataset. Take the results obtained on the *G. subterraneus* dataset as an example, the distribution of sets classified as negative classes by different models in the test set is shown in Figure 2D. In the VN diagram on the left, 41.4% of the test set is judged as negative by all models

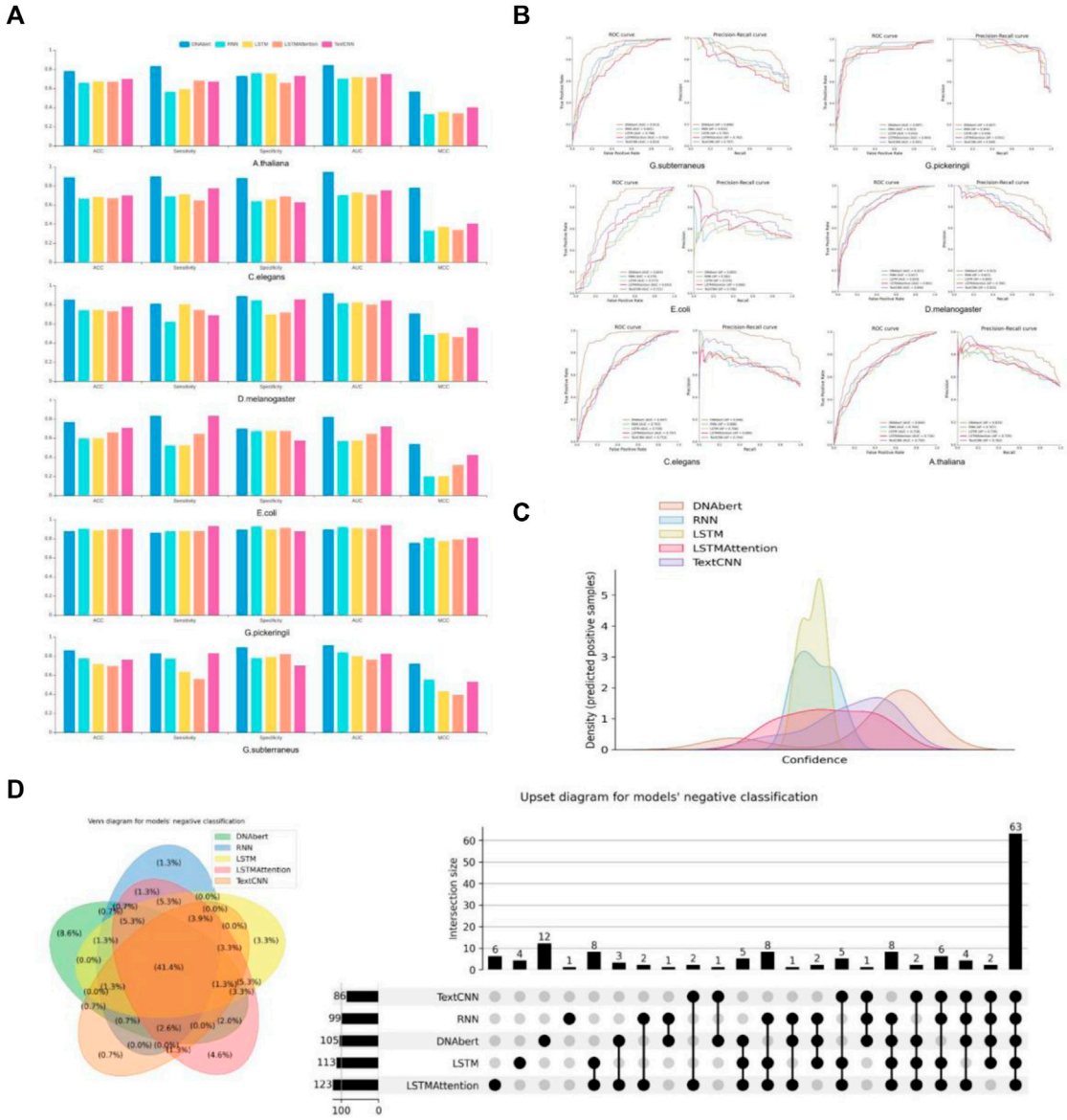


FIGURE 2
Performance evaluation of multiple models. **(A)** The basic statistics of ACC, Sensitivity, Specificity, AUC, and MCC in different models. **(B)** Performance comparison between DNABERT and other state-of-the-art methods on the benchmark datasets. **(C)** Density distribution of the prediction confidence by different deep learning models. **(D)** VENN and Upset plots show the overlap of different models' predictions.

(negative classes account for 50% of the test set in total). The difference in quantity is shown more clearly in the right figure, and we can find that DNABERT may be one of the less effective models for classification under this dataset, as it predicts more negative cases individually. However, given that most of the model predictions converge on the same, we can conclude that most of the models are consistent in their classification results.

Deep learning model feature analysis

We conduct a comparative study on the features learned by deep learning from biological information. This includes comparisons between different deep learning models as well as comparisons between deep learning features and manually designed features. By conducting feature comparisons, we aim to further validate the superiority of deep learning methods and enhance the interpretability of deep learning models. We select ANF, binary, CKSNAP, and DNC approaches to extracting features and using SVM for unsupervised classification to compare with our deep learning models. Figure 3A presents the ROC and PR curves for all models on the *G. pickeringii* dataset. We only display the two best-performing traditional manual feature methods for comparison. It is evident that most of the deep learning methods outperform the traditional approaches in terms of classification performance.

To visualize the results of deep learning features, we utilized UMAP (Uniform Manifold Approximation and Projection) and

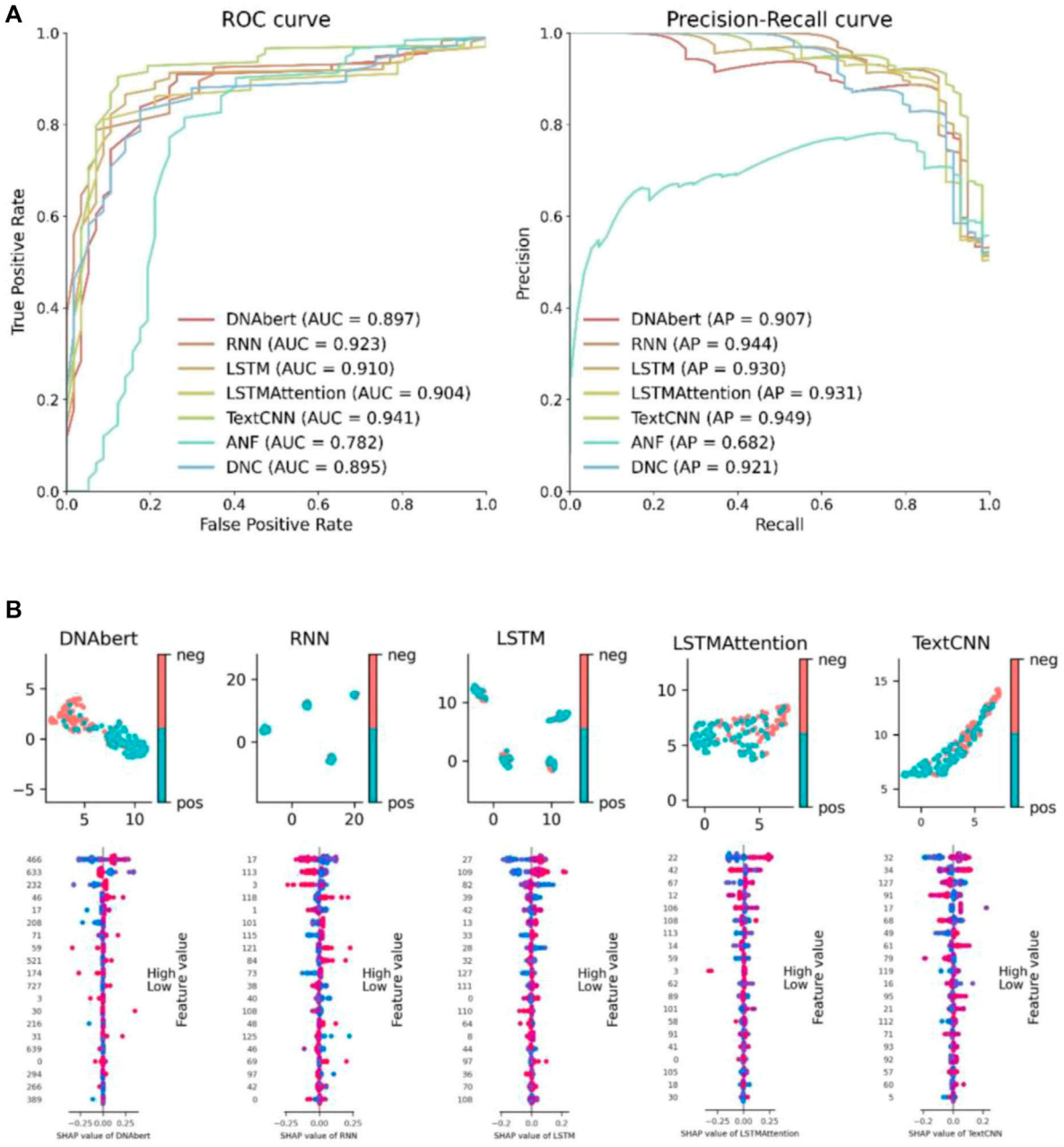


FIGURE 3
Deep learning model feature analysis. **(A)** Feature performance comparison between hand-crafted features and the features learned by deep learning models. **(B)** UMAP feature visualization and SHAP feature importance visualization.

SHAP (Shapley Additive Explanations) plots for display (Figure 3B). The UMAP plot reduces the dimensionality of the features while preserving the underlying data structure. It enables data clustering and categorization by mapping high-dimensional features into a lower-dimensional space, allowing for an analysis of feature similarity between positive and negative instances. The SHAP plot facilitates the understanding of feature importance and contribution to model predictions, providing interpretability to the model and enabling comparison of feature impacts. It helps to comprehend the significance of features in model predictions, enhancing interpretability and facilitating comparison among different features. In the feature visualization figure, each row corresponds to a specific feature, and the x-axis represents the snap value, providing a clearer understanding of the feature. The color gradient indicates the feature value, with higher values represented by redder colors and lower values represented by bluer colors. Each line represents a feature, and the horizontal position represents the SHAP value assigned to that feature in a particular sample. Each point represents a sample. The intensity of the color reflects the impact of the feature, with redder colors

indicating a larger impact and bluer colors indicating a smaller impact. The scattered distribution of points indicates a greater influence of the feature.

Conclusion

In this study, we use several currently popular deep learning models on the problem of 4mC methylation detection of DNA. We first present the current status of DNA 4mC methylation site detection, followed by the design of deep learning model workflows on six benchmark datasets, and finally, we evaluate the output of all models and conclude that deep learning has great potential for methylation detection, leading the way to future sequencing technologies along with newer bio-experimental methods. In fact, deep learning methods consistently outperformed traditional machine learning methods on all datasets. Furthermore, it was observed that pre-trained deep learning models with a higher number of parameters exhibited even better performance. We believe this may be because deep learning models with more parameters capture more features and analyze the features acquired by each model. By attempting to explain the model's internal workings and shed light on its internal representations, we aim to define its "black box" behavior.

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 authors.

Author contributions

HJ: Data curation, Validation, Writing–original draft, Writing–review and editing. JB: Writing–review and editing. JJ: Data curation, Writing–review and editing. YC: Data curation, Writing–review and editing. XC: Writing–review and editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This research work was supported by the Innovation Fund of the Ministry of Education's Engineering Research Center for the Integration and Application of Digital Learning Technologies, under project grant number 1221001.

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.

References

Ao, C., Jiao, S., Wang, Y., Yu, L., and Zou, Q. (2022a). Biological sequence classification: A review on data and general methods. *Research* 2022, 0011. doi:10.34133/research.0011

Ao, C., Zou, Q., and Yu, L. (2022b). NmRF: Identification of multispecies RNA 2'-O-methylation modification sites from RNA sequences. *Briefings Bioinforma.* 23, bbab480. doi:10.1093/bib/bbab480

Buryanov, Y. I., and Shevchuk, T. (2005). DNA methyltransferases and structural-functional specificity of eukaryotic DNA modification. *Biochem. Mosc.* 70, 730–742. doi:10.1007/s10541-005-0178-0

Cai, L., Wang, L., Fu, X., and Zeng, X. (2021). Active semisupervised model for improving the identification of anticancer peptides. *ACS Omega* 6, 23998–24008. doi:10.1021/acsomega.1c03132

Cao, C., Wang, J., Kwok, D., Cui, F., Zhang, Z., Zhao, D., et al. (2022). webTWAS: a resource for disease candidate susceptibility genes identified by transcriptome-wide association study. *Nucleic Acids Res.* 50, D1123–D1130. doi:10.1093/nar/gkab957

Chen, J., Zou, Q., and Li, J. (2022). DeepM6ASeq-EL: Prediction of human N6-methyladenosine (m6A) sites with LSTM and ensemble learning. *Front. Comput. Sci.* 16, 162302. doi:10.1007/s11704-020-0180-0

Chen, K., Zhao, B. S., and He, C. (2016). Nucleic acid modifications in regulation of gene expression. *Cell Chem. Biol.* 23, 74–85. doi:10.1016/j.chembiol.2015.11.007

Chen, W., Yang, H., Feng, P., Ding, H., and Lin, H. (2017). iDNA4mC: identifying DNA N4-methylcytosine sites based on nucleotide chemical properties. *Bioinformatics* 33, 3518–3523. doi:10.1093/bioinformatics/btx479

Chen, Y., Yang, X., Wang, J., Song, B., and Zeng, X. (2021). Muffin: Multi-scale feature fusion for drug–drug interaction prediction. *Bioinformatics* 37, 2651–2658. doi:10.1093/bioinformatics/btab169

Dong, J., Zhao, M., Liu, Y., Su, Y., and Zeng, X. (2022). Deep learning in retrosynthesis planning: Datasets, models and tools. *Briefings Bioinforma.* 23, bbab391. doi:10.1093/bib/bbab391

Dwyer, D. B., Falkai, P., and Koutsouleris, N. (2018). Machine learning approaches for clinical psychology and psychiatry. *Annu. Rev. Clin. Psychol.* 14, 91–118. doi:10.1146/annurev-clinpsy-032816-045037

Graves, A. (2012). "Long Short-Term Memory," in *Supervised Sequence Labelling with Recurrent Neural Networks. Studies in Computational Intelligence* (Berlin, Heidelberg: Springer) 385, 37–45.

Graves, A., and Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Netw.* 18, 602–610. doi:10.1016/j.neunet.2005.06.042

Hamdy, R., Maghraby, F. A., and Omar, Y. M. K. (2022). ConvChrome: Predicting gene expression based on histone modifications using deep learning techniques. *Curr. Bioinforma.* 17, 273–283. doi:10.2174/1574893616666211214110625

Hasan, M. M., Manavalan, B., Shoombuatong, W., Khatun, M. S., and Kurata, H. (2020). i4mC-Mouse: Improved identification of DNA N4-methylcytosine sites in the mouse genome using multiple encoding schemes. *Comput. Struct. Biotechnol. J.* 18, 906–912. doi:10.1016/j.csbj.2020.04.001

He, W., Jia, C., and Zou, Q. (2019). 4mCPred: Machine learning methods for DNA N4-methylcytosine sites prediction. *Bioinformatics* 35, 593–601. doi:10.1093/bioinformatics/bty668

Hu, Y., Sun, J. Y., Zhang, Y., Zhang, H., Gao, S., Wang, T., et al. (2021). rs1990622 variant associates with Alzheimer's disease and regulates TMEM106B expression in human brain tissues. *BMC Med.* 19, 11. doi:10.1186/s12916-020-01883-5

Hu, Y., Zhang, H., Liu, B., Gao, S., Wang, T., Han, Z., et al. (2020). rs34331204 regulates TSPAN13 expression and contributes to Alzheimer's disease with sex differences. *Brain* 143, e95. doi:10.1093/brain/awaa302

Hu, Y., Zhang, Y., Zhang, H., Gao, S., Wang, L., Wang, T., et al. (2022a). Cognitive performance protects against Alzheimer's disease independently of educational attainment and intelligence. *Mol. Psychiatry* 27, 4297–4306. doi:10.1038/s41380-022-01695-4

Hu, Y., Zhang, Y., Zhang, H., Gao, S., Wang, L., Wang, T., et al. (2022b). Mendelian randomization highlights causal association between genetically increased C-reactive protein levels and reduced Alzheimer's disease risk. *Alzheimers Dement.* 18, 2003–2006. doi:10.1002/alz.12687

Huang, Q. F., Zhang, J., Wei, L. Y., Guo, F., and Zou, Q. (2020). 6mA-RicePred: A method for identifying DNA N (6)-methyladenine sites in the rice genome based on feature fusion. *Front. Plant Sci.* 11, 4. doi:10.3389/fpls.2020.00004

Ji, Y., Zhou, Z., Liu, H., and Davuluri, R. V. (2021). Dnabert: Pre-trained bidirectional encoder representations from transformers model for DNA-language in genome. *Bioinformatics* 37, 2112–2120. doi:10.1093/bioinformatics/btab083

Jiao, Y., and Du, P. (2016). Performance measures in evaluating machine learning based bioinformatics predictors for classifications. *Quant. Biol.* 4, 320–330. doi:10.1007/s40484-016-0081-2

Jin, J., Yu, Y., Wang, R., Zeng, X., Pang, C., Jiang, Y., et al. (2022). iDNA-ABF: multi-scale deep biological language learning model for the interpretable prediction of DNA methylations. *Genome Biol.* 23, 219–223. doi:10.1186/s13059-022-02780-1

Kim, Y. (2014). *Convolutional neural network for sentence classification[J]*. Waterloo, ON: University of Waterloo. arXiv preprint arXiv:1408.5882.

Kulis, M., and Esteller, M. (2010). DNA methylation and cancer. *Adv. Genet.* 70, 27–56. doi:10.1016/B978-0-12-380866-0.60002-2

Laird, P. W. (2010). Principles and challenges of genome-wide DNA methylation analysis. *Nat. Rev. Genet.* 11, 191–203. doi:10.1038/nrg2732

Larranaga, P., Calvo, B., Santana, R., Bielza, C., Galdiano, J., Inza, I., et al. (2006). Machine learning in bioinformatics. *Briefings Bioinforma.* 7, 86–112. doi:10.1093/bib/bbk007

Li, J., He, S. D., Guo, F., and Zou, Q. (2021). HSM6AP: A high-precision predictor for the Homo *sapiens* N6-methyladenosine (m6 A) based on multiple weights and feature stitching. *Rna Biol.* 18, 1882–1892. doi:10.1080/15476286.2021.1875180

Li, Z., Zhu, S., Shao, B., Zeng, X., Wang, T., and Liu, T. Y. (2023). DSN-DDI: An accurate and generalized framework for drug–drug interaction prediction by dual-view representation learning. *Briefings Bioinforma.* 24, bbac597. doi:10.1093/bib/bbac597

Liu, C., Song, J., Ogata, H., and Akutsu, T. (2022). MSNet-4mC: Learning effective multi-scale representations for identifying DNA N4-methylcytosine sites. *Bioinformatics* 38, 5160–5167. doi:10.1093/bioinformatics/btac671

Lv, H., Dao, F. Y., Zhang, D., Yang, H., and Lin, H. (2021). Advances in mapping the epigenetic modifications of 5-methylcytosine (5mC), N6-methyladenine (6mA), and N4-methylcytosine (4mC). *Biotechnol. Bioeng.* 118, 4204–4216. doi:10.1002/bit.27911

Manavalan, B., Basith, S., Shin, T. H., Lee, D. Y., Wei, L., and Lee, G. (2019). 4mCpred-EL: An ensemble learning framework for identification of DNA N4-methylcytosine sites in the mouse genome. *Cells* 8, 1332. doi:10.3390/cells8111332

Moore, L. D., Le, T., and Fan, G. (2013). DNA methylation and its basic function. *Neuropsychopharmacology* 38, 23–38. doi:10.1038/npp.2012.112

Ni, P., Huang, N., Zhang, Z., Wang, D. P., Liang, F., Miao, Y., et al. (2019). DeepSignal: Detecting DNA methylation state from nanopore sequencing reads using deep-learning. *Bioinformatics* 35, 4586–4595. doi:10.1093/bioinformatics/btz276

Pan, X., Lin, X., Cao, D., Zeng, X., Yu, P. S., He, L., et al. (2022). Deep learning for drug repurposing: Methods, databases, and applications. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* 12, e1597. doi:10.1002/wcms.1597

Plongthongkum, N., Diep, D. H., and Zhang, K. (2014). Advances in the profiling of DNA modifications: Cytosine methylation and beyond. *Nat. Rev. Genet.* 15, 647–661. doi:10.1038/nrg3772

Razin, A., and Cedar, H. (1991). DNA methylation and gene expression. *Microbiol. Rev.* 55, 451–458. doi:10.1128/mr.55.3.451-458.1991

Ren, S. J., Yu, L., and Gao, L. (2022). Multidrug representation learning based on pretraining model and molecular graph for drug interaction and combination prediction. *Bioinformatics* 38, 4387–4394. doi:10.1093/bioinformatics/btac538

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature* 323, 533–536. doi:10.1038/323533a0

Sharma, A. K., and Srivastava, R. (2021). Protein secondary structure prediction using character bi-gram embedding and Bi-LSTM. *Curr. Bioinforma.* 16, 333–338. doi:10.2174/1574893615999200601122840

Song, B., Luo, X., Luo, X., Liu, Y., Niu, Z., and Zeng, X. (2022). Learning spatial structures of proteins improves protein–protein interaction prediction. *Briefings Bioinforma.* 23, bbab558. doi:10.1093/bib/bbab558

Tran, H. V., and Nguyen, Q. H. (2022). iAnt: Combination of convolutional neural network and random forest models using PSSM and BERT features to identify antioxidant proteins. *Curr. Bioinforma.* 17, 184–195. doi:10.2174/1574893616666210820095144

Wang, R., Jiang, Y., Jin, J., Yin, C., Yu, H., Wang, F., et al. (2022). DeepBIO is an automated and interpretable deep-learning platform for biological sequence prediction, functional annotation, and visualization analysis, 2022.2009.2029.509859. bioRxiv. doi:10.1101/2022.09.29.509859

Xiao, Z. C., Wang, L. Z., Ding, Y. J., and Yu, L. A. (2022). iEnhancer-MRBF: Identifying enhancers and their strength with a multiple Laplacian-regularized radial basis function network. *Methods* 208, 1–8. doi:10.1016/j.ymeth.2022.10.001

Xu, H., Jia, P., and Zhao, Z. (2021). Deep4mC: Systematic assessment and computational prediction for DNA N4-methylcytosine sites by deep learning. *Briefings Bioinforma.* 22, bbaa099. doi:10.1093/bib/bbaa099

Xu, J., Meng, Y., Lu, C., Cai, L., Zeng, X., Nussinov, R., et al. (2023). Graph embedding and Gaussian mixture variational autoencoder network for end-to-end analysis of single-cell RNA sequencing data. *Cell Rep. Methods* 3, 100382. doi:10.1016/j.crmeth.2022.100382

Yu, Y., He, W., Jin, J., Xiao, G., Cui, L., Zeng, R., et al. (2021). iDNA-ABT: advanced deep learning model for detecting DNA methylation with adaptive features and transductive information maximization. *Bioinformatics* 37, 4603–4610. doi:10.1093/bioinformatics/btab677

Zeng, X., Wang, F., Luo, Y., Kang, S. G., Tang, J., Lightstone, F. C., et al. (2022a). Deep generative molecular design reshapes drug discovery. *Cell Rep. Med.* 4, 100794. doi:10.1016/j.xcrm.2022.100794

Zeng, X., Xiang, H., Yu, L., Wang, J., Li, K., Nussinov, R., et al. (2022b). Accurate prediction of molecular properties and drug targets using a self-supervised image representation learning framework. *Nat. Mach. Intell.* 4, 1004–1016. doi:10.1038/s42256-022-00557-6

Zeng, X., Zhu, S., Lu, W., Liu, Z., Huang, J., Zhou, Y., et al. (2020). Target identification among known drugs by deep learning from heterogeneous networks. *Chem. Sci.* 11, 1775–1797. doi:10.1039/c9sc04336e

Zhang, Y., Lin, J., Zhao, L., Zeng, X., and Liu, X. (2021). A novel antibacterial peptide recognition algorithm based on BERT. *Briefings Bioinforma.* 22, bbab200. doi:10.1093/bib/bbab200

Zhao, Z., Zhang, X., Chen, F., Fang, L., and Li, J. (2020). Accurate prediction of DNA N4-methylcytosine sites via boost-learning various types of sequence features. *BMC genomics* 21, 627. doi:10.1186/s12864-020-07033-8

Zulfiqar, H., Huang, Q. L., Lv, H., Sun, Z. J., Dao, F. Y., and Lin, H. (2022b). Deep-4mCGP: A deep learning approach to predict 4mC sites in geobacter pickeringii by using correlation-based feature selection technique. *Int. J. Mol. Sci.* 23, 1251. doi:10.3390/ijms23031251

Zulfiqar, H., Sun, Z. J., Huang, Q. L., Yuan, S. S., Lv, H., Dao, F. Y., et al. (2022a). Deep-4mCW2V: A sequence-based predictor to identify N4-methylcytosine sites in *Escherichia coli*. *Methods* 203, 558–563. doi:10.1016/j.ymeth.2021.07.011

frontiers | Frontiers in Genetics

TYPE Technology and Code
PUBLISHED 24 August 2023
DOI 10.3389/fgene.2023.1228552

OPEN ACCESS

EDITED BY
Nathan Olson,
National Institute of Standards and Technology (NIST), United States

REVIEWED BY
Safdar Ali,
Aliah University, India
Nancy Manchanda,
Orna Therapeutics, Inc., United States

*CORRESPONDENCE
Rommel Thiago Jucá Ramos,
rommelramos@ufpa.br

†PRESENT ADDRESS
Carlos Willian Dias Dantas,
Laboratory of Biological Engineering, Biological Science Institute, Park of Science and Technology, Federal University of Pará, Belém, Brazil

RECEIVED 25 May 2023
ACCEPTED 28 July 2023
PUBLISHED 24 August 2023

CITATION
Alves SIA, Ferreira VBC, Dantas CWD, Silva ALdCd and Ramos RTJ (2023), EasySSR: a user-friendly web application with full command-line features for large-scale batch microsatellite mining and samples comparison.
Front. Genet. 14:1228552.
doi: 10.3389/fgene.2023.1228552

EasySSR: a user-friendly web application with full command-line features for large-scale batch microsatellite mining and samples comparison

Sandy Ingrid Aguiar Alves[1], Victor Benedito Costa Ferreira[1], Carlos Willian Dias Dantas[2†], Artur Luiz da Costa da Silva[1] and Rommel Thiago Jucá Ramos[1*]

[1]Laboratory of Biological Engineering, Biological Science Institute, Park of Science and Technology, Federal University of Pará, Belém, Brazil, [2]Institute of Biological Sciences, Federal University of Minas Gerais, Belo Horizonte, Brazil

Microsatellites, also known as SSRs or STRs, are polymorphic DNA regions with tandem repetitions of a nucleotide motif of size 1–6 base pairs with a broad range of applications in many fields, such as comparative genomics, molecular biology, and forensics. However, the majority of researchers do not have computational training and struggle while running command-line tools or very limited web tools for their SSR research, spending a considerable amount of time learning how to execute the software and conducting the post-processing data tabulation in other tools or manually—time that could be used directly in data analysis. We present EasySSR, a user-friendly web tool with command-line full functionality, designed for practical use in batch identifying and comparing SSRs in sequences, draft, or complete genomes, not requiring previous bioinformatic skills to run. EasySSR requires only a FASTA and an optional GENBANK file of one or more genomes to identify and compare STRs. The tool can automatically analyze and compare SSRs in whole genomes, convert GenBank to PTT files, identify perfect and imperfect SSRs and coding and non-coding regions, compare their frequencies, abundancy, motifs, flanking sequences, and iterations, producing many outputs ready for download such as PTT files, interactive charts, and Excel tables, giving the user the data ready for further analysis in minutes. EasySSR was implemented as a web application, which can be executed from any browser and is available for free at https://computationalbiology.ufpa.br/easyssr/. Tutorials, usage notes, and download links to the source code can be found at https://github.com/engbiopct/EasySSR.

KEYWORDS

batch, genome, microsatellites, motifs, large scale, web tool, comparison, bioinformatics

1 Introduction

Microsatellites, also known as Simple Sequence Repeats (SSRs) or Short Tandem Repeats (STRs), are polymorphic DNA regions with tandem repetitions of a nucleotide motif ranging 1–6 bp, also called mononucleotide, di-, tri-, tetra-, penta-, and hexanucleotide repeats (Pinheiro et al., 2022). They can be categorized into perfect, imperfect, and compound and

are found in both coding and non-coding regions in eukaryotes, prokaryotes, and viruses (Mudunuri and Nagarajaram, 2007; Beier et al., 2017). The SSRs have various clinical implications and a broad range of applications in many fields, such as conservation and evolutionary studies, comparative genomics, molecular biology, biotechnology, oncology, and forensics (Laskar et al., 2022; Pinheiro et al., 2022).

With the application of computational approaches in biological data along with the advance of Next-Generation Sequencing technologies (NGS), many tools for SSR mining have been developed over the years, with IMEx (Mudunuri and Nagarajaram, 2007), MISA (Beier et al., 2017), TRF (Benson, 1999), and Repeat Masker (Tarailo-Graovac and Chen, 2009) among the most popular and widely used tools, as reviewed by Mudunuri et al. (2010a), Lim et al. (2013), Mathur et al. (2020).

However, many researchers need advanced computational training and therefore have difficulty using these tools as most of these tools: i) Need significant investment of time for the user to comprehend, install, and run those pieces of software; ii) Are command-line based without graphical interface; iii) Require device storage and dependencies for installation; iv) Have many parameters and dependencies that might confuse inexperienced users; v) Require specific file formats as input, e.g., PTT files, which are not easily obtainable for inexperienced users who would rather use FASTA and GenBank files; and vi) Are not available anymore, principally web servers. vii) Lastly, the few web tools still available are very limited in many aspects, such as the limited size of the input files, rare flexibilization of parameters, and the lack of identification of flanking sequences, downloadable outputs, post-processed graphical outputs, and features for online sample comparison, or they do not focus solely on Microsatellites motifs (1–6 bp) but also on other Tandem repeats such as Minisatellites (10–30 bp) and Satellites (>100 pb); indeed, in some cases, even if the web service does exist, the full functionality is restricted to the command-line version, limiting the online service to basic and small analysis (Lim et al., 2013).

In this way, many scientists end up choosing to use command-line tools for full functionality and spend a considerable amount of time learning how to install and execute the software, in addition to performing post-processing data tabulation on other tools or manually, instead of focusing more time on data analysis; thus, there is a need for a web application that can be an easy tool for online analysis that can do the same as command-line tools, filling in the gaps of other software without sacrificing the full-fledged and accurate results already obtained (Oliveira et al., 2008; Pinheiro et al., 2022).

Given these lacunae, we present EasySSR, an intuitive web tool that implements command-line IMEx versatile and accurate SSR mining with novel settings by automatizing the analysis from data input, converting individual files, and performing the post-processing analysis of the individual outputs, fully summarizing those data into statistics sheets and graphs available online for the user. It was designed for practical and intuitive use in batch identifying perfect and imperfect SSRs in large-scale data from one or many individual FASTA sequences, draft, or complete genomes, with full functionality and data visualization directly from the web without the need for any software installation, their dependencies, or complicated bioinformatic skills to run, giving the user results that can be easily interpreted, enabling even traditional non-bioinformatician scientists with limited computational experience and resources to use SSRs in their research (Mudunuri and Nagarajaram, 2007).

2 Methods

2.1 Workflow and implementation

EasySSR is a web tool hosted in a standard Linux server, developed using the Django v4.1.7 framework (Django Software Foundation, 2023), based on the Python language v3.11, with information stored in a MariaDB database v10.10.2, and it executes several helper scripts in Python and Perl to automate the following summarized workflow in the back-end, as summarized in Figure 1.

EasySSR receives the User Information—User Project name (required), Email (optional); Input Files—FASTA files (required), GENBANK files (optional); and Parameters—Default or Custom when the user clicks the upload button. EasySSR uses secure HTTPS (Hypertext Transfer Protocol Secure) connections to transfer data between the client and the server. Step 1 starts when the files are uploaded. If the user uploaded GenBank files, the script verifies if every FASTA file has a corresponding GenBank annotation file and if both have the same filename with less than 35 characters. Then, it converts the GenBank files to PTT format through a script in Perl. If no GenBank file was uploaded, EasySSR considers everything as non-coding by default. In the web interface, the process from upload to GBK-PTT conversion is shown as Step 1 to the user. Step 2 starts with a script in Python for batch execution of IMEX v2.1 for each FASTA file. This step might be slower or faster depending on the size of the input files and the complexity of the annotation and the parameters. For Step 3, EasySSR scans the folders generated by IMEX, reads the IMEX TXT outputs, and records each result in the database created for that project. After extraction, the interactive charts and tables from SQL queries in the database are rendered for the web with a color-blind-friendly palette using the Chart and jQuery v3.6 JavaScript libraries with the DataTables plugin. The front-end of EasySSR was encoded with Bootstrap v4.0 and jQuery v3.6 libraries, generating user-appealing interfaces in the web interface and exhibiting the post-processed outputs in HTML format, which are available for download alongside the IMEX outputs. The project data are stored through a project ID in the EasySSR database for a month-long period.

2.2 Tool validation

In order to validate EasySSR, a web tool with full command-line functionalities that is suitable for large-scale comparative analysis, it was availed by three different perspectives: i) Firstly, to demonstrate the functionality of EasySSR against other web tools, it was compared with the most cited tools that have an active web service with a feature for the identification of Microsatellites. However, as the online tools do not support the analysis of SSRs in large datasets, and this is the main

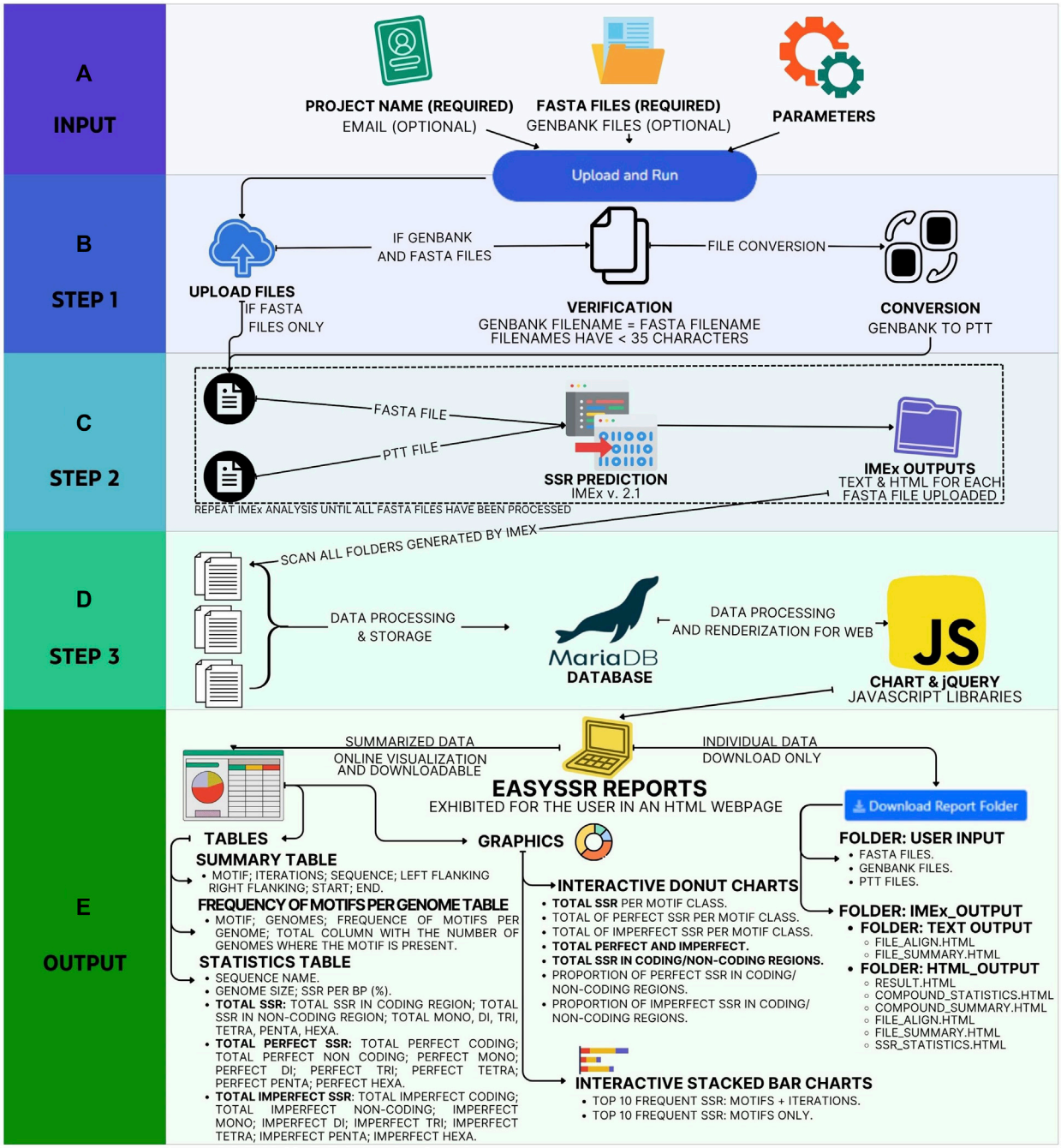


FIGURE 1
EasySSR workflow from user input to output. **(A)** In input, EasySSR receives user information, user, and parameters. **(B)** In Step 1, it receives the input, verifies the data, and converts GENBANK to PTT files. **(C)** With each pair of FASTA files-PTT files ready, EasySSR starts Step 2 by analyzing every file with IMEx, repeating the process until all files have been processed. **(D)** Then, in Step 3, EasySSR processes all IMEX outputs, stores the data in a new project at the database, and processes the summarized data into sheets and charts. **(E)** The output is exhibited through a HTML page, and the data are made available for download.

distinguished attribute of EasySSR, performance validation had to be executed in comparison with command-line tools. In this way, for ii), benchmark testing was used for two datasets previously validated by Beier et al. (2017), Mudunuri and Nagarajaram (2007), in order to measure the efficiency against the main similar software and their specific datasets, for both prokaryotes and eukaryotes, and with FASTA input only or both FASTA and GenBank. The first dataset had a homogeneous set of small artificial prokaryotic chromosomes used for benchmark EasySSR performance while running intraspecific analysis for perfect SSRs, using only FASTA files as input. The second dataset had a heterogeneous set of complete prokaryote genomes, eukaryotic chromosomes, and a human gene and was used for benchmark

EasySSR performance while running interspecific analysis for imperfect SSRs, using both FASTA and GenBank files as input. ii) Lastly, to demonstrate EasySSR capacity to process large datasets of complete genomes, the program was executed with a dataset validated by Pinheiro et al. (2022), for batch comparison of 54 whole genomes of *Corynebacterium pseudotuberculosis*, running interspecific analysis for perfect SSRs, using both FASTA and GenBank files as input.

2.2.1 Function comparison against web tools

Many web services offer features for microsatellite mining. However, they are widely different in terms of functionality and the analysis, input, output content, and output return style (Mudunuri et al., 2010b). In this way, EasySSR was compared to other web tools in order to demonstrate the main functionalities that are common to them or exclusive to our tool. For this validation, six review articles were screened to discover web tools that have a feature for the identification of Microsatellites (Leclercq et al., 2007; Sharma et al., 2007; Merkel and Gemmell, 2008; Mudunuri et al., 2010b; Lim et al., 2013; Mathur et al., 2020). The publishing articles for each tool were analyzed in April 2023, and the platforms were tested through the links available in the articles to check if they were still active. If the tool was functional, the article citation rates were analyzed through Google Scholar, and these data alongside with the search link were tabulated. The 10 most cited web tools were used for features comparison against EasySSR. The features used for comparison were partially based on the ones analyzed by Merkel and Gemmell (2008), Mudunuri et al. (2010b) in their articles. Besides the Citations and Author/Publishing Year, the following categories and features were used in this comparison: i) ANALYSIS: Microsatellite only, Maximum motif length, Perfect SSRs, Imperfect SSRs, Compound SSRs, Flexible Parameters, and Large-scale analysis; ii) INPUT: Limits Max, File Size, Analyze web of many whole genomes, Accepts multiple FASTA files, Integration with NCBI, and Box for cut-and-paste small sequences; iii) OUTPUT CONTENT: Text file, HTML file, PTT file, Coding/Non-coding, Flanking Sequences, Sample comparison sheets, and Sample comparison graphs; iv) OUTPUT RETURN: Web results, Email results, and Download results.

2.2.2 Benchmark testing against web servers and command-line tools

2.2.2.1 Intraspecific analysis for perfect SSRs in prokaryotes, using only FASTA files as input with custom parameters

For this benchmark testing, the dataset employed by Beier et al. (2017) was used to validate Misa-Web, a set of small barley bacterial artificial chromosomes (BACs) available in the NCBI database under the accession numbers: AC256511.1 (113 kb), AC257258.1 (124 kb), AC259365.1 (118 kb), AC261250.1 (91 kb), AC263353.1 (33 kb), AC264961.1 (126 kb), AC265197.1 (113 kb), AC266636.1 (167 kb), AC267178.1 (121 kb), and AC269605.1 (119 kb). For this comparison, the sequence assemblies were obtained with the same version used in their original article, through their NCBI accession numbers, and analyzed for perfect SSRs. Only the FASTA files were used as input in the analysis as the annotation available in NCBI consists only of gaps and has no gene information. This dataset is also available at EasySSR webpage and GitHub as "Dataset 1—Misa."

The detected microsatellites and execution time of EasySSR were compared against tools that also have settings for perfect SSR search only, also known as Misa-mode, those being the web servers of MISA-web (Beier et al., 2017) and TRF web (Benson, 1999) and command-line tools ProGeRF (Lopes et al., 2015), GMATo (Wang et al., 2013), mreps (Kolpakov, 2003), and SciRoKo (Kofler et al., 2007). The analysis was executed with the same parameters as the original benchmark test: minimum repeat copy number - Mono:5, Di: 5, Tri: 5, Tetra: 5, Penta: 5, Hexa: 5); Imperfection and Mismatches–0 (Perfect SSR only–Misa mode); dMAX compound SSR–0 bp.

2.2.2.2 Interspecific analysis for imperfect SSR in prokaryotes and eukaryotes, using both FASTA and GenBank files as input, with custom parameters

For the second benchmark testing, the dataset validated by Mudunuri and Nagarajaram (2007) was used to validate IMEX 1.0 through the analysis of an interspecific sequence set composed of the human atrophin1 gene, 5 kb (BC051795); two eukaryote chromosomes - *Plasmodium falciparum* chromosome IV, 1,193 kb (NC_004318.1) and yeast chromosome IV, 1,518 kb (NC_001136.8); and two complete prokaryote genomes - *Mycobacterium tuberculosis* H37Rv, 4,370 kb (NC_000962.2) and *Escherichia coli* K12, 4,596 kb (NC_000913.2). The sequences were obtained through their NCBI accession numbers, with the same version as their original article, downloaded as FASTA and GenBank annotation files, which were renamed to: ("Ecoli_K12.fasta," "Ecoli_K12.gb"); ("Human_Atrophin1.fasta," "Human_Atrophin1.gb"); ("MTB_H37Rv.fasta," "MTB_H37Rv.gb"); ("Plasmodium_Chr4.fasta," "Plasmodium_Chr4.gb"); and ("Yeast_Chr4.fasta," "Yeast_Chr4.fasta"), in a way that both FASTA and GenBank have the same filename besides the extensions, and the filename has less than 35 characters. This dataset is also available at EasySSR webpage and GitHub as "Dataset 2—IMEx."

The detected microsatellites and execution time of EasySSR were compared against tools that also have settings for imperfect SSR search: TRF (Benson, 1999), IMEx 1.0 (Mudunuri and Nagarajaram, 2007 original article data), IMEx 2.1 (Mudunuri et al., 2010a), and Sputnik (Morgante et al., 2002). The following parameters were used, those being the same ones applied by Mudunuri and Nagarajaram, 2007: minimum repeat copy number–Mono:5, Di: 3, Tri: 2, Tetra: 2, Penta: 2, Hexa: 2, Imperfection of all tracts to 10%, mismatches - Mono: 1, Di: 1, Tri: 1, Tetra: 2, Penta: 2, Hexa: 3; with the additional parameters of dMAX cSSR of 0 bp, 15 bp for flanking sequences, and standardization level 3.

2.2.3 Large-scale interspecific analysis for imperfect SSR, using both FASTA and GenBank files as input with default parameters

Differently from the benchmark tests, this comparison aimed to demonstrate the capacity of EasySSR to handle large datasets while being a versatile shortcut for online data analysis. For this, 54 complete genomes of *C. pseudotuberculosis* (CP) were selected, which have been previously studied by Pinheiro et al. (2022), who also used IMEx 2.1 as the microsatellite mining tool. The sequences were obtained at NCBI through the accession numbers stated in Table 4, with the same version as the ones stated in the original

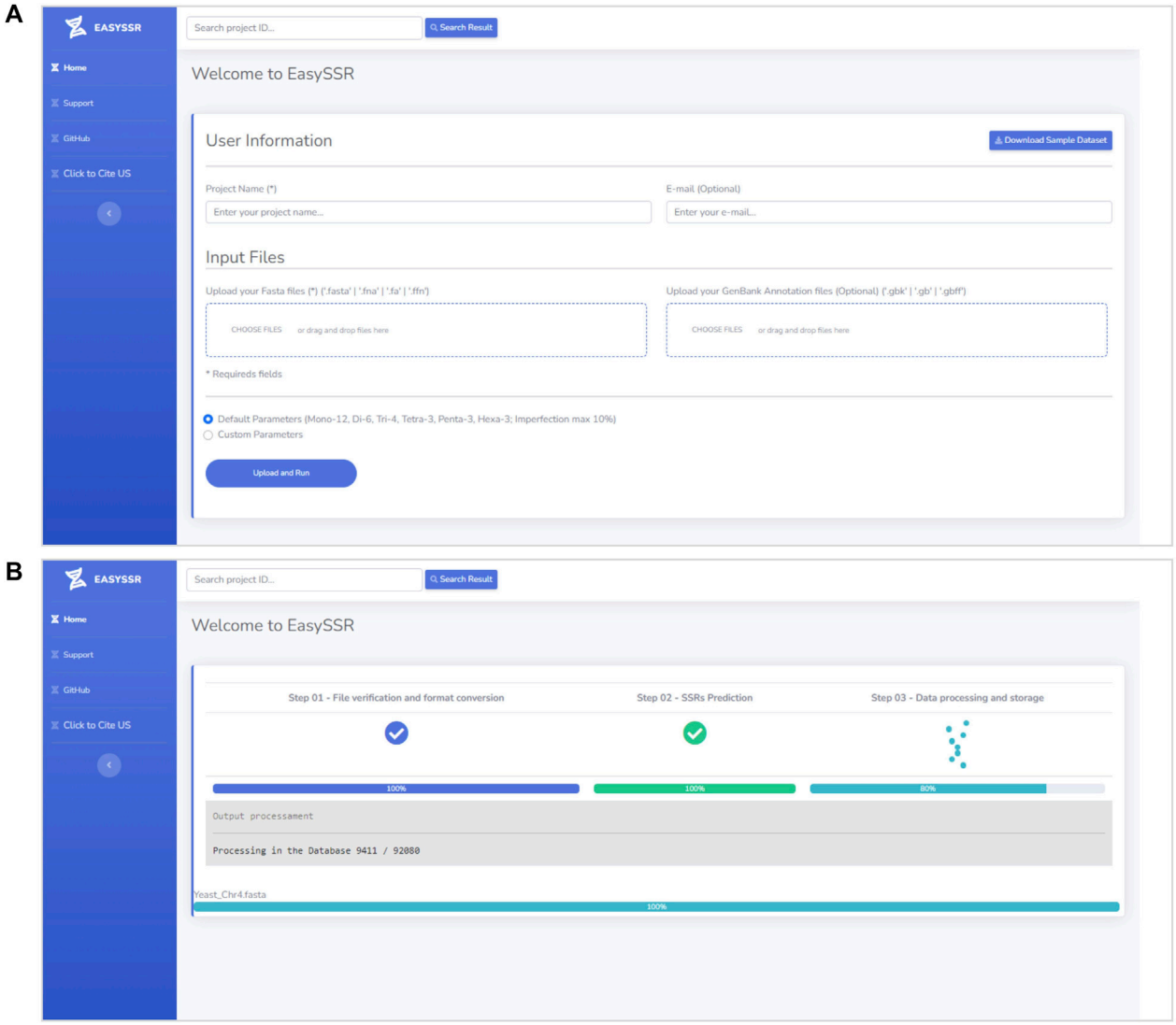


FIGURE 2
(A) EasySSR input screen. **(B)** EasySSR loading screen.

article by Pinheiro et al. (2022), and downloaded as FASTA and GenBank annotation files.

For this analysis, the dataset was processed in EasySSR with slightly different parameters, in custom mode and default mode. In general, the main parameters were the same for both analyses: Minimum Repeat Number–Mono:12, Di: 6, Tri: 4, Tetra: 3, Penta: 3, Hexa: 3, flanking sequences of size 15 bp, dMax compound of 0, Standardization level 3, extracting all types of SSR, and yes for identify coding/non-coding regions, generate alignment, and text outputs. However, the first analysis was conducted by searching for perfect SSRs only, with the same parameters as Pinheiro et al. (2022), by using the custom parameters mode and setting the imperfection and mismatches as 0, expecting to have the same results as them. Then, the second analysis was conducted by searching for perfect and imperfect SSRs, using the EasySSR default parameters, which were also based on and adapted from Pinheiro et al. (2022), but with Imperfection % - Mono: 10%, Di: 10%, Tri: 10%, Tetra: 10%, Penta: 10%, Hexa:10% and Mismatch in Pattern: Mono: 1; Di:1; Tri:1; Tetra:2; Penta:2; Hexa:2. The results were compared with Pinheiro et al. (2022) through the graphs and charts generated as the output of EasySSR.

3 Results and discussion

3.1 Tool overview

EasySSR is an intuitive web server designed in order to facilitate the SSR research, which does not require mandatory registration or work in any browser and is freely available to non-commercial users at https://computationalbiology.ufpa.br/easyssr/(Figure 2A), with tutorials, usage note, and source code available at https://github.com/engbiopct/EasySSR.

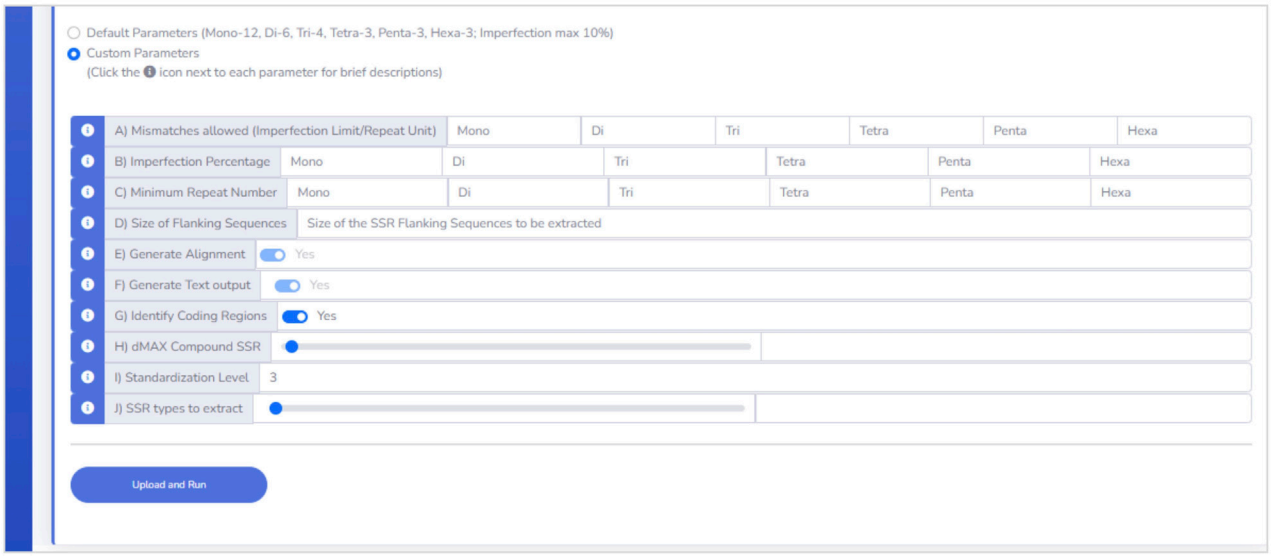


FIGURE 3
Custom parameters interface.

It offers many automatized extra features for data visualization and sample comparison, besides the IMEX sensitivity and its advanced functions to identify microsatellites, such as searching perfect microsatellites separately, getting the coding/non-coding information of the microsatellite tracts, generating alignments with consensus microsatellite tracts, restricting the imperfection limit for the repeat unit of each size, setting the imperfection percentage threshold of each repeat size, restricting the minimum number of repeat units of a tract of each size, searching for repeats of a particular size or all sizes, setting the flanking sequence size limit, and standardizing the repeats.

As for the automatized features unique to EasySSR, it can automatically convert GenBank to PTT files, it summarizes SSRs frequencies, abundancy, flanking sequences, and iterations of motifs, producing many outputs ready to download such as PTT files, IMEX HTML/TXT discover-friendly outputs, interactive charts, and summarized data/statistics Excel tables for comparison of the samples, giving the user the data ready for further analysis in a computationally feasible time. This reduces a significant amount of time worth of data tabulation, minimizing tedious manual operations and therefore decreasing the chance of errors.

As the information about compound SSRs is restricted to IMEX HTML files, this version of EasySSR does not include compound SSRs in the summary tables, including only their raw data of each file analyzed in the downloadable folder IMEx outputs, focusing their comparison on perfect and imperfect SSRs and their respective positions in coding/non-coding regions.

3.1.1 Input files

EasySSR requires only a project name and one or more FASTA files containing nucleotide sequences or genomes (draft/complete) for the identification and comparison of STRs (Figure 2A). If the user intends to identify coding/non-coding regions, a GENBANK file should also be uploaded for each FASTA file. Only the FASTA file is mandatory, whereas the GENBANK file is optional. When an annotation file is not uploaded, the algorithm will automatically assume that all sequences in the FASTA file are non-coding. However, with an annotation file, the algorithm will leverage the provided information to calculate the distribution of motifs in coding and non-coding regions. In the case of a multi-FASTA file input, EasySSR will identify SSRs, but the file will be treated and analyzed as a single draft genome. The algorithm treats each FASTA file as an independent genome, comparing them separately, and utilizes the input FASTA files filename as the sequence name in the EasySSR outputs. This web application uses secure HTTPS (Hypertext Transfer Protocol Secure) connections to transfer data between the client and the server, ensuring that the data are not intercepted during transmission and not used for purposes other than the intended analysis, with the project data being stored in the EasySSR database for a month-long period.

3.1.2 Default parameters

The tool runs with intuitive default or custom flexible parameters and has no limit size for input (Figure 2A). In this way, users can load as many genomes as they want for their analysis, depending only on the computational structure available. The user does not need to input any parameter in the default parameters mode but, rather, just select this option and execute EasySSR. The preset default parameters are based on Pinheiro et al. (2022): Repeat Number: 1–12, 2–6, 3–4, 4–3, 5-3, and 6–3; adapted to allow the imperfection maximum of 10% with 1 or 2 mismatches: Imperfection % (p%): 1%–10%, 2%–10%, 3%–10%, 4%–10%, 5%–10%, 6%–10%; and Mismatch in Pattern: 1–1; 2–1; 3–1; 4–2; 5–2; 6–2. Maximum distance for compound SSR: 0 bp; Standardization Level: Level 3; Flanking Sequences: 15 bp; Extract all SSR types, Generate Alignment, and Text Output: “Yes.” In this way, the user can easily write a project name, input the files to be analyzed, and press the “Upload and Run” button, as shown in Figure 2A. The

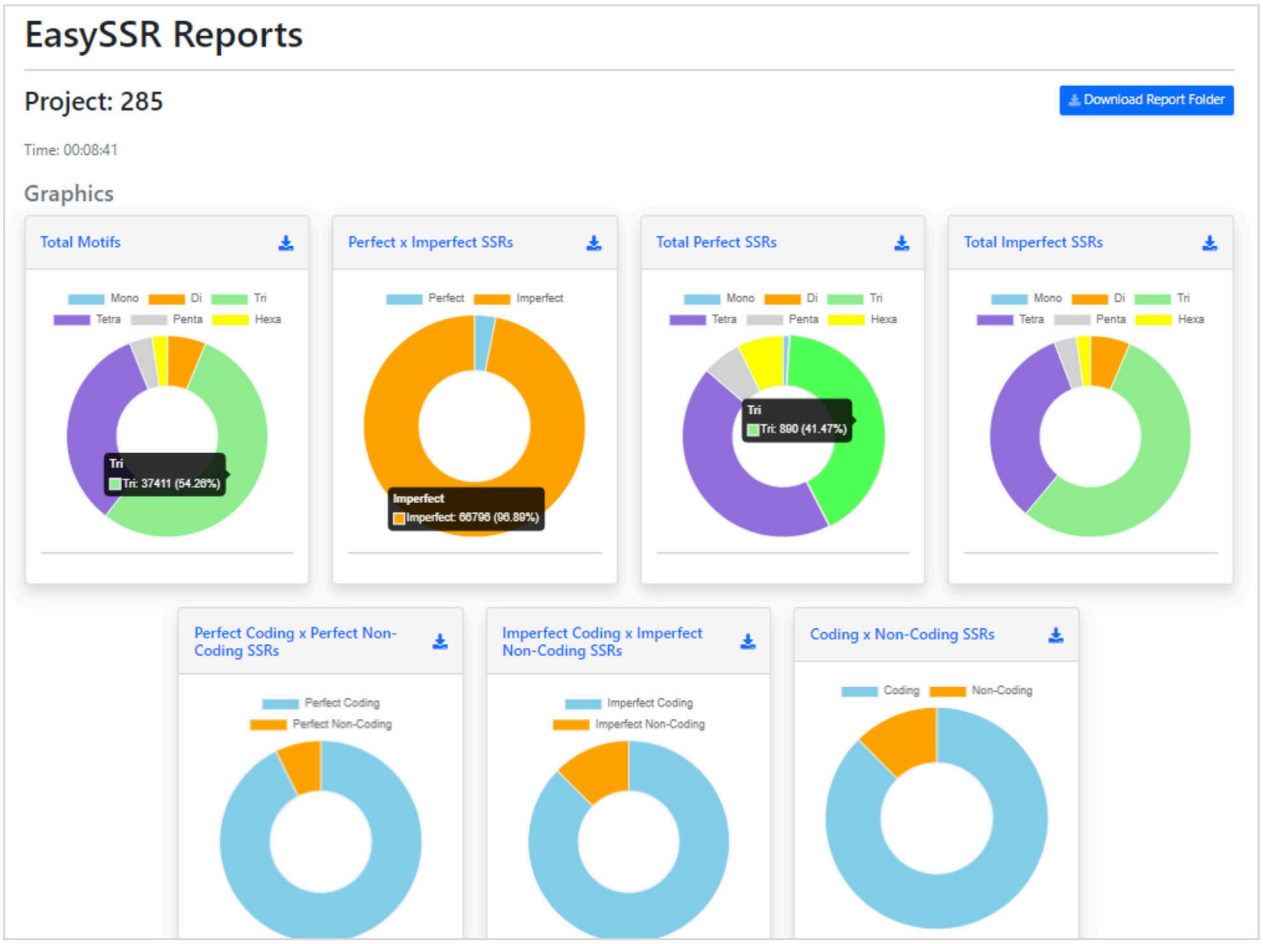


FIGURE 4
Easy SSR output screen part 1, with time of analysis, download report folder, and donut comparison charts. Demonstration of EasySSR Reports from the batch comparison of perfect and imperfect SSR in 54 complete genomes of *Corynebacterium pseudotuberculosis* with gene annotation.

loading screen will be then exhibited, as demonstrated in Figure 2B, until the analysis is complete.

3.1.3 Beyond the default parameters

EasySSR Custom mode (Figure 3) enables users to adjust analysis parameters (A to J) based on preferences, with brief descriptions conveniently accessible via the information icon i). This user-friendly feature aids in selecting suitable values, empowering customization to specific requirements. The only mandatory fields for user input in Custom mode are from A to D: (A) Mismatches; (B) Imperfection %. To restrict the analysis to perfect SSR only, also known as Misa-mode, the user can define all the settings in parameters (A) and (B) to 0; (C) Minimum Repeat Number; and (D) Size of Flanking Sequences. The other parameters, from (E) to (J), can be used as the preset: (E) Generate Alignment and (F) Generate Text output are fixed in YES since EasySSR processes those files to generate the summarized outputs, charts, and tables; (G) Identify Coding Regions is preset as YES but can be set as NO; (H) Maximum distance for Compound SSR is preset at 0 but can be set from −1 to 100; (I) Standardization level is preset at 3 but can be set as 0, 1, 2, 3, or F; (J) SSR types to extract is preset at 0 to extract all SSR types, but users can set from 1 to 6 to extract only a type of SSR.

3.1.4 Outputs

After the analysis, the web page is updated automatically, and the EasySSR reports page is exhibited (Figure 4). The user can see a blue button to download the report folder in ZIP format, containing both the files used for input (FASTA, GenBank, and the generated PTT) and the complete IMEX output files for each genome individually, in HTML and TEXT formats comprising summary, align, results, and statistics about compound, perfect, and imperfect.

Back to the EasySSR Reports interface, the user has 07 interactive donut charts with the comparative analysis of total motifs, perfect, and imperfect proportions, total of perfect SSR per motif class, total of imperfect SSR per motif class, proportion of perfect motifs in coding/non-coding regions, proportion of imperfect motifs in coding/non-coding regions, and the general comparison of SSR in coding/non-coding regions (Figure 4). It also plots 02 interactive bar charts containing the top 10 SSR motifs present in the genomes analyzed (Figure 5). The first stacked bar chart (Figure 5A) depicts the frequency distribution of the motif

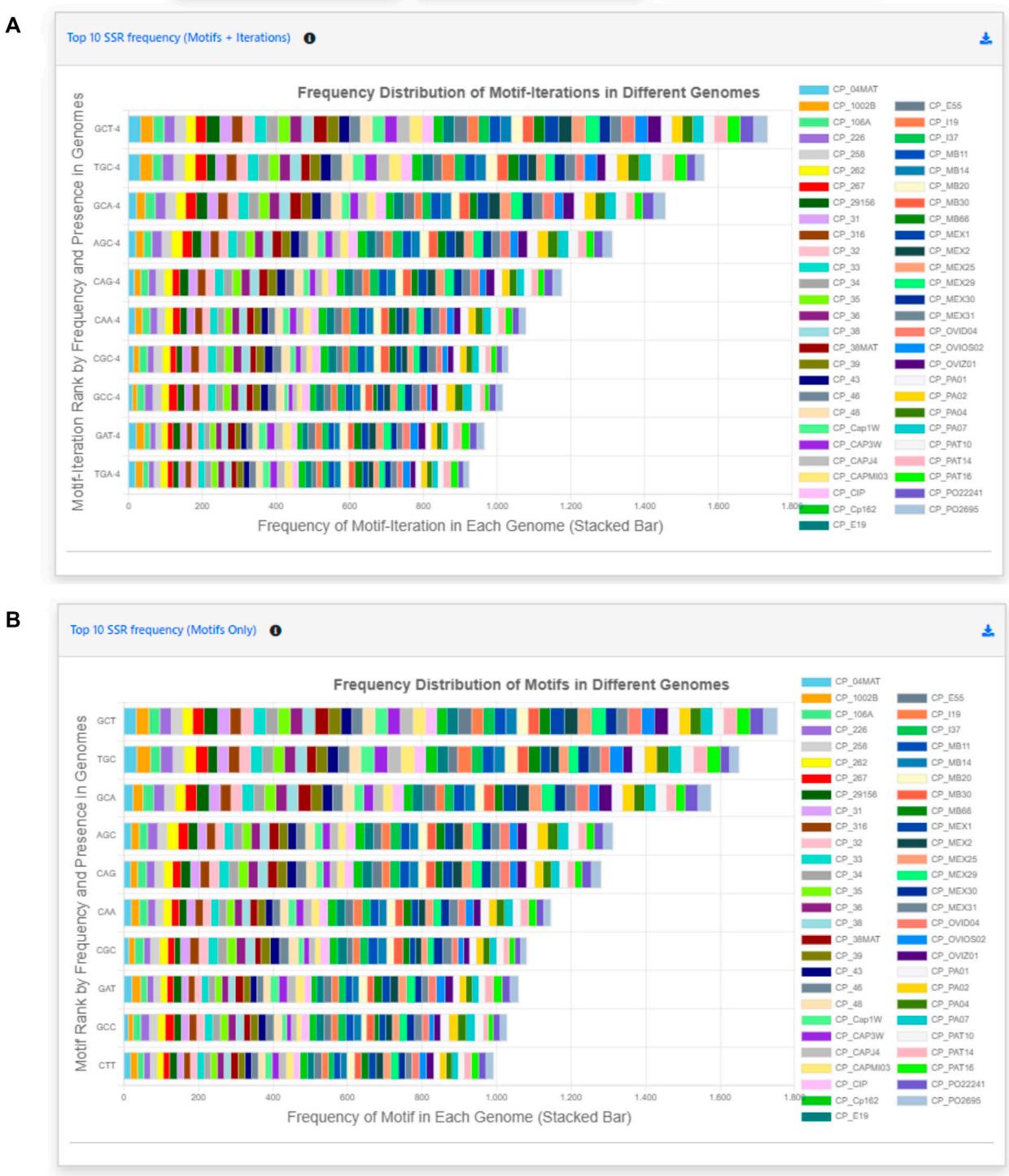


FIGURE 5
Easy SSR output screen part 2, from the large-scale analysis and comparison of perfect and imperfect SSR in 54 complete genomes of *Corynebacterium pseudotuberculosis* with gene annotation. **(A)** Interactive stacked bar chart summarizing the top 10 motifs with iteration present in most genomes, with their frequency per genome. **(B)** Interactive stacked bar chart summarizing the top 10 motifs present in most genomes, with their frequency per genome.

iterations present in all the analyzed genomes. In contrast, the second chart (Figure 5B) represents the frequency distribution of the motifs across the genomes. The x-axis displays the frequency of the motif (Figure 5B) and motif iteration (Figure 5A) in each genome. At the same time, the stacked bars represent the absolute frequency of the motif (Figure 5B) and motif iteration

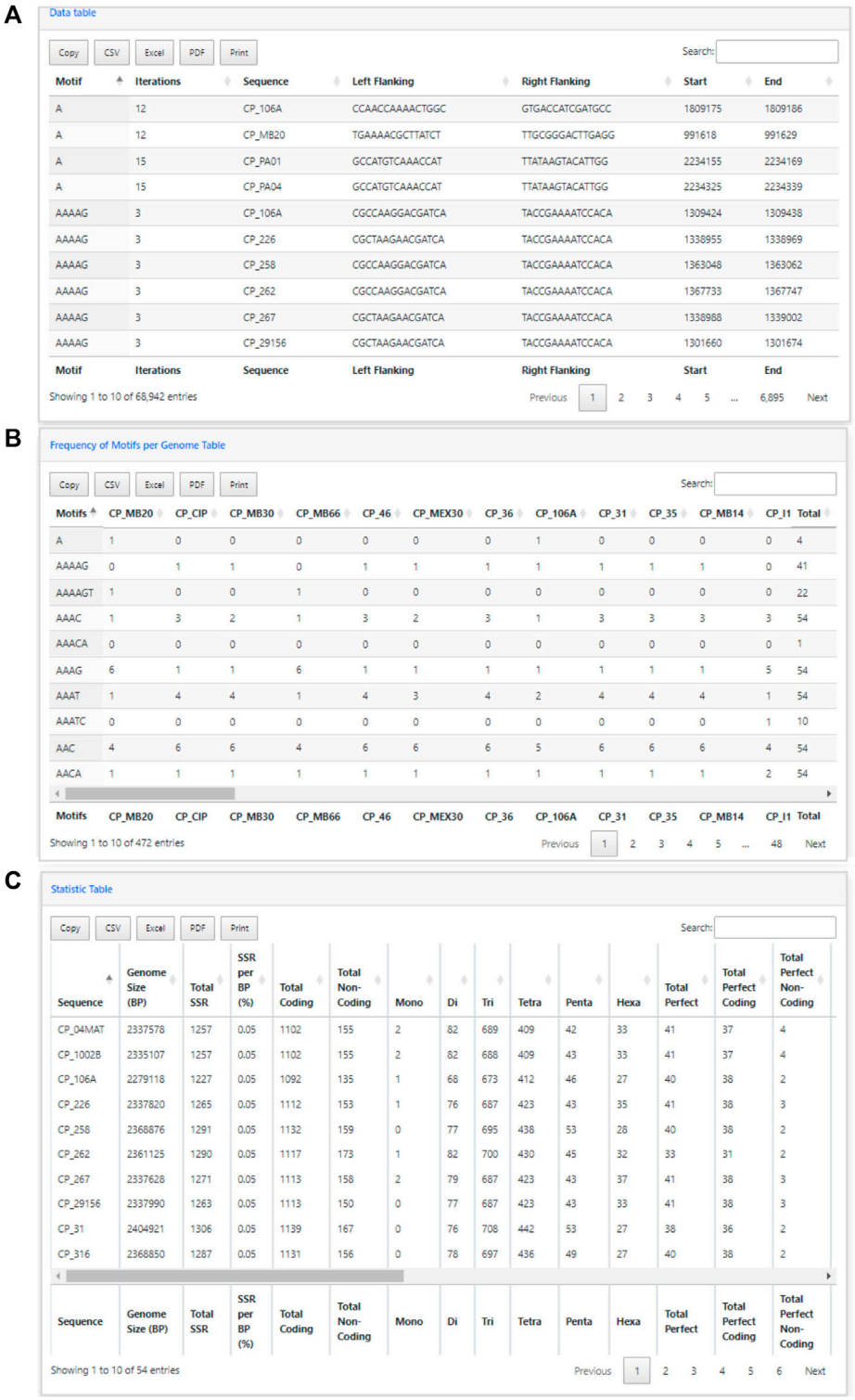


FIGURE 6
Easy SSR output screen part 3, from the large-scale analysis and comparison of perfect and imperfect SSRs in 54 complete genomes of *Corynebacterium pseudotuberculosis*, with gene annotation. **(A)** Data table, **(B)** Frequency of Motifs per Genome table, and **(C)** Statistics table ordered by sequence name.

(Figure 5A) across all genomes. The y-axis ranks the motif (Figure 5B) and motif iterations (Figure 5A) from highest to lowest based on their frequency and presence in the genomes. The top of the y-axis corresponds to the motif (Figure 5B) and motif iteration (Figure 5B) that is present in the highest number of genomes and has the highest absolute frequency in the stacked bar.

TABLE 1 Web tool's function comparison made with EasySSR and the most-cited top 10 web tools available in April 2023.

Name	EasySSR	TRF web	Repeat masker web	Misa-web	Batch Primer3	Mreps	Websat	SSR locator	STAR	Imperfect SSR finder	PolyMorph predict*
Citations	This article	7077	1860	927	909	459	348	262	137	11	10
Author/Year	This article	Benson 1999	Smit 1996 apud Tarailo-Graovac 2009	Beier 2017	You 2008	Kolpakov 2003	Martins 2009	Da Maia 2008	Delgrange 2004	Stieneke 2007	Das 2019
ANALYSIS											
Microsatellites only	Yes	No	No	Yes	Yes	No	Yes	Yes	No	Yes	Yes
Maximum motif length	1–6 pb	1–2000 pb	No limit	1–6 pb	2–6 pb	No limit	1–6 pb	2–10 pb	No limit	2–10 pb	1–6 pb
Perfect SSRs	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
Imperfect SSRs	Yes	Yes	Yes	No	No	Yes	No	Yes	Yes	Yes	No
Compound SSRs	Yes	No	No	Yes	No	No	No	No	No	Yes	Yes
Flexible Parameters	Yes	Yes	No	No	No	Yes	No	No	No	Yes	No
Large-scale analysis	Yes	No	No	No	Yes	Yes	No	No	No	No	No
INPUT											
Limits Max. File Size	No	10 Mb	10 Mb	2 Mb	No	No	150 kb	No	1 Mb	No	No
Analyze web of many whole genomes	Yes	No	No	No	No	No	No	No	No	No	No
Accepts multiple FASTA files	Yes	No	No	No	No	No	No	No	No	No	No
Integration with NCBI	No	No	No	Yes	No	No	No	No	No	No	No
Box for cut and paste small sequences	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	No
OUTPUT CONTENT											
Text file	Yes	No	Yes	Yes	Yes	No	No	No	Yes	Yes	Yes
HTML file	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	No	Yes	No
PTT file	Yes	No	No	No	No	No	No	No	No	No	No
Coding/Non-coding	Yes	No	No	No	No	No	No	No	No	No	No
Flanking Sequences	Yes	Yes	No	No	Yes	No	No	No	No	No	No
Sample comparison sheets	Yes	No	No	No	No	No	No	No	No	No	No
Sample comparison graphs	Yes	No	No	No	No	No	No	No	No	No	Yes

(Continued on following page)

TABLE 1 (*Continued*) Web tool's function comparison made with EasySSR and the most-cited top 10 web tools available in April 2023.

Name	EasySSR	TRF web	Repeat masker web	Misa-web	Batch Primer3	Mreps	Websat	SSR locator	STAR	Imperfect SSR finder	PolyMorph predict*
OUTPUT RETURN											
Web results	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	No	Yes	No
Email results	No	No	Yes	Yes	No	No	No	No	Yes	No	Yes
Download results	Yes	Yes	Yes	Yes	Yes	No	Yes	No	Yes	Yes	Yes

"Yes" to facilitate easier identification of tools that possess the specific feature.

TABLE 2 Comparison of detected perfect microsatellites and execution time (in seconds) of SSR tools analyzed by Beier 2017 and EasySSR.

Sequence	GMATo	TRF	Mreps	SciRoKo	ProGeRF	MISA-web	EasySSR
AC256511.1 (113 kb)	549	580	56	549	560	549	588
AC257258.1 (124 kb)	938	943	85	938	901	938	984
AC259365.1 (118 kb)	641	666	76	641	628	641	666
AC261250.1 (91 kb)	498	457	60	498	456	498	529
AC263353.1 (33 kb)	153	173	–	153	142	153	167
AC264961.1 (126 kb)	654	620	–	654	605	654	728
AC265197.1 (113 kb)	505	496	44	505	503	505	549
AC266636.1 (167 kb)	839	865	79	839	811	839	861
AC267178.1 (121 kb)	517	530	46	516	496	517	540
AC269605.1 (119 kb)	728	676	76	728	700	728	762
Sum	6,022	6,006	522	6,021	5,802	6,022	6,374
Execute time per batch (seconds)	7.5	30.7	1.2	0.6	21	1.8	5

In addition to the charts, EasySSR analysis includes three tables with filters and search options (Figure 6). The first table (Figure 6A) provides data on each motif, including its iterations, Genome, Left Flanking, Right Flanking, Start, and End positions. The second table, Frequency of Motifs per Genome (Figure 6B), has been created to enhance the representation of motif frequency distribution across the different genomes. It offers a detailed count of each motif's occurrence in the genomes and a "total" column indicating the number of genomes in which each motif is present. This addition offers a more comprehensive and user-friendly view of the data. The third table is the statistic table (Figure 6C). It contains various summarized quantitative data about the perfect and imperfect SSRs identified in each genome. These statistics include the genome size, total SSR count, percentage proportion of SSRs per base pair (calculated using the formula = [(SSR*100)/genome_size)], total SSR in Coding/Non-coding regions, total SSR per motif class, and subgroup analyses of perfect/imperfect and coding/non-coding SSRs.

These data are available for individual download. The plotted charts are in PNG/JPEG format and the tables in CSV, Excel (.xlsx), and PDF formats, also with the copy/print options. The user can save the EasySSR Reports HTML page using their browser option or write down the project number to consult within a month.

3.2 Tool validation

3.2.1 Function comparison against web tools

Web-tools for microsatellite mining are important as they simplify the search and analysis of microsatellite data; they do not require an investment of time for the user to install and run the software, neither do they require device storage and dependencies for installation (Sousa et al., 2018). Plenty of web tools have been released over time, but many accession links available in the articles are not functional totally or partially anymore, as is the case with ATRhunter (Wexler et al., 2004), Tandem Swan (Boeva et al., 2006), STRING (Parisi et al., 2003), MICAS and IMEx web (Sreenu, 2003), MsatFinder (Thurston and Field, 2005), RISA (Kim et al., 2012), and LSAT (Biswas et al., 2018). The web tools still available have a variety of specific features but are very limited in many aspects in comparison to command-line tools. After analyzing the citation rates and checking their availability, we defined the top 10 most-cited SSR web tools that were still operational in April 2023: TRF web (Benson, 1999), Repeat Masker web (Tarailo-Graovac and Chen, 2009), Misa-Web (Yang et al., 2018), Batch Primer3 (You et al., 2008), Mreps (Kolpakov, 2003), Websat (Martins et al., 2009), SSR Locator (da Maia et al., 2008), STAR (Delgrange and Rivals, 2004), Imperfect SSR Finder (Stieneke and Eujayl, 2007), and PolyMorph Predict (Das et al., 2019), Their features were compared with EasySSR and summarized in Table 1.

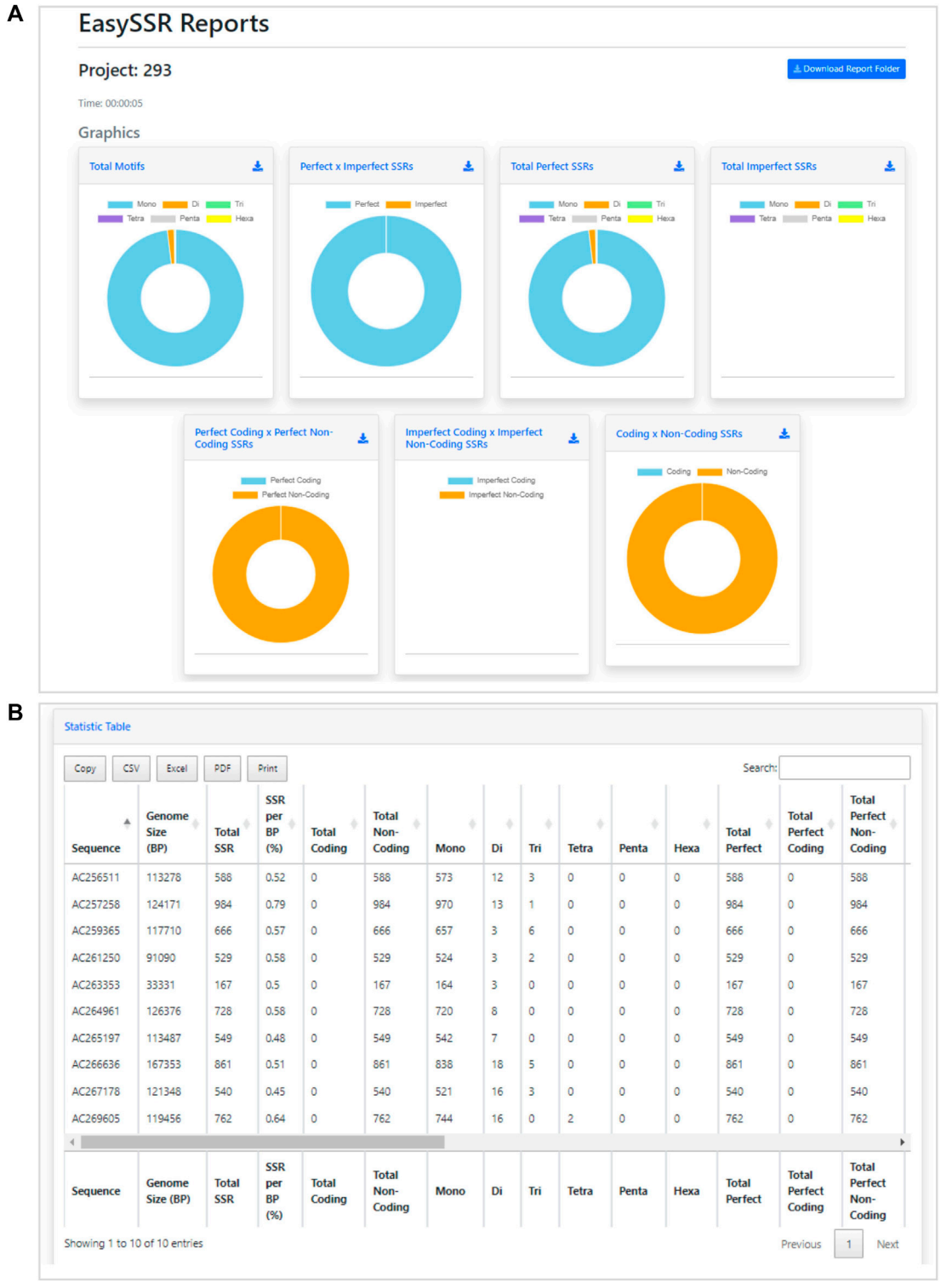


FIGURE 7
Demonstration of EasySSR Reports from the batch comparison of perfect SSRs in 10 BAC genomes without gene annotation. **(A)** EasySSR comparison charts with graphs for imperfect SSRs are blank due to the parameters set for mining perfect SSRs only, and coding/non-coding graphs are all in one color because no annotation file was input **(B)** EasySSR statistics table reports in web mode, with all coding information as 0 because no annotation file was input.

The main limitations observed were the limited size of the input files, rare flexibilization of parameters, and the lack of identification of flanking sequences, downloadable outputs, summarized and post-processed graphical outputs, and features for online sample comparison, and that there is no exclusive focus on Microsatellites motifs (1–6 bp) but also on other Tandem repeats

TABLE 3 Comparison of detected microsatellites and execution time (in seconds) of SSR tools analyzed by Mudunuri and Nagarajaram (2007), IMEX 2.1, and EasySSR.

Sequence	TRF	Sputnik	IMEx 1.0 (2007)	Imex 2.1 (2023)	EasySSR
Yeast Chr4 (1,531 Kb)	7308	2,831	39,759	40,239	40,239
Plasmodium Chr4 (1,204 Kb)	25,601	10,810	54,232	55,693	55,693
MTB H37Rv (4,411 Kb)	16,439	9,412	111,113	111,583	111,583
Human Atrophin 1 (4,43 Kb)	50	19	146	146	146
E.coli K12 (4,639 Kb)	12,043	5,387	105,392	106,243	106,243
Sum	61,441	28,459	310,642	313,904	313,904
Execute time per batch (seconds)	108.5	402.5	30.8	51.7	72.0

such as Minisatellites (10–30 bp) and Satellites (>100 pb). In some cases, even if the web service does exist, the full functionality is restricted to the command-line version, limiting the online service to basic and small analysis.

TRF (Benson, 1999) and Repeats Masker (Tarailo-Graovac and Chen, 2009) are by far the most used tools, according to the citation rate. Alongside Mreps (Kolpakov, 2003) and STAR (Delgrange and Rivals, 2004), they are tools that are not limited to microsatellites but aim to identify all tandem repeats, including other types such as Minisatellites and Satellites. STAR is a tool focused on locating a given motif in a DNA sequence, instead of screening all motifs like the other Tandem Repeat tools (Delgrange and Rivals, 2004). To individuals who need to focus just on microsatellites, SSR-specific web applications such as EasySSR, Misa-web (Beier et al., 2017), Websat (Martins et al., 2009), SSR Locator (da Maia et al., 2008), and Imperfect SSR finder (Stieneke and Eujayl, 2007) may be more appropriate due to their specific range of motifs.

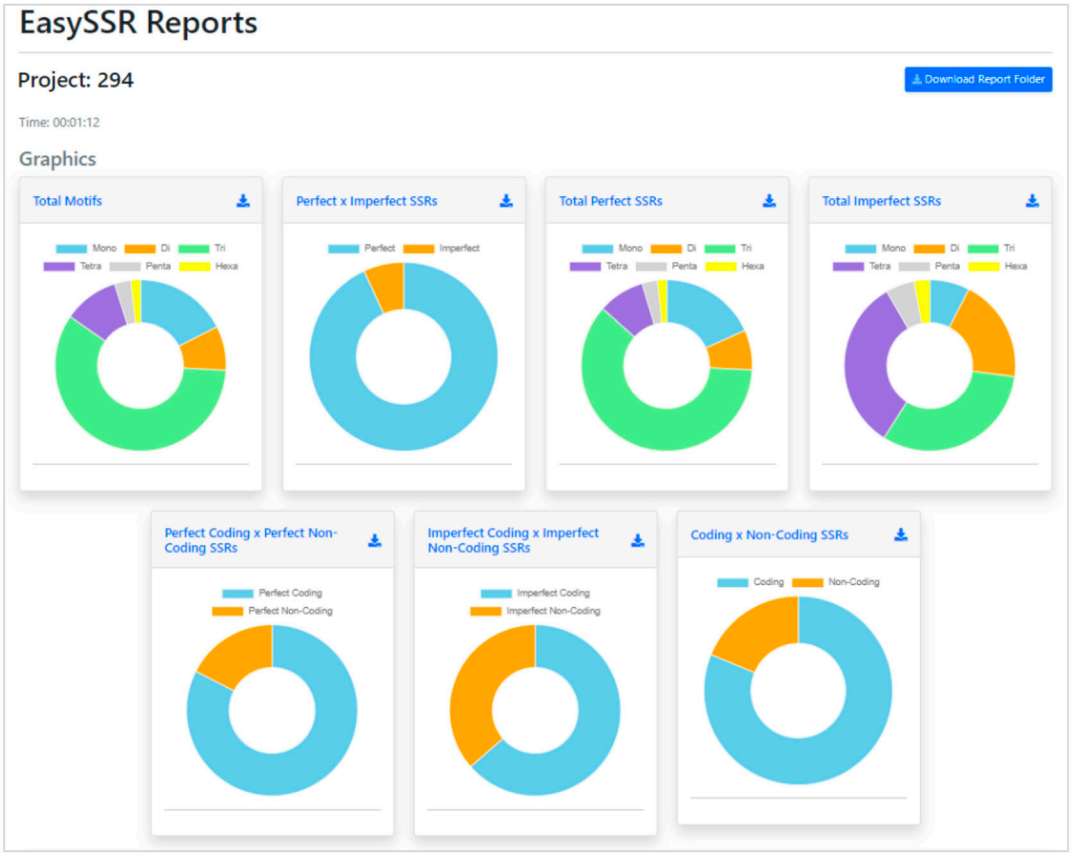
Batch Primer3 (You et al., 2008), Websat (Martins et al., 2009), and Polymorph predict (Das et al., 2019), in contrast to EasySSR, have integrated the primer design function. Nevertheless, at the time this work was being produced, Polymorph predict (Das et al., 2019) was malfunctioning by running only their native sample data ("Chromosome 2") instead of the user input. Websat (Martins et al., 2009) restricts accepting input files containing more than 150,000 characters. Furthermore, its primary focus lies in designing primers for a limited number of manually selected SSRs, making it unsuitable for users needing comprehensive, automated online analysis on a large scale, a capability provided by BatchPrimer3 and EasySSR. BatchPrimer3 (You et al., 2008) functions well for large-scale primer analysis and SSR screening because the output is a list containing the identified SSRs and their respective flanking primers with details, statistics, and outputs in HTML, Text file, and Excel, but it does not analyze imperfect and compound SSRs, nor does it determine whether they are in coding or non-coding regions, and it does not perform online sample comparison like EasySSR.

The command-line version of Misa (Thiel et al., 2003; Beier et al., 2017) is a versatile tool that provides analysis of perfect and compound SSRs, being one of the gold standards in SSR mining. Many tools, such as Polymorph predict (Das et al., 2019), integrate Misa in their analysis, while others write additional advanced scripts to process Misa outputs, such as Galasso and Ponzoni (2015). However, many of the applications are limited to computational experts who can develop scripts or at least comprehend how to execute them in the command-line. For non-experienced users, command-line tools are not as user-friendly as online services. Misa also has a web-server, but it does not provide the user all the features and capabilities of the command line, accepting only a single file with a maximum size of 2 Mb as input. Unfortunately, many users may find this to be a significant impediment to their research because a single prokaryote genome may be larger than 2 Mb. Misa-web results are two files: raw SSR data and statistics, not shown on a web interface but instead transmitted over email. On the other hand, EasySSR is able to process many genomes in a single run, with no maximum or minimum size limit, and summarize and compare them. It analyzes not only perfect and compound SSRs but also imperfect SSRs, offering the user the flexibility to include or exclude imperfects from their SSR mining. By running IMEX (Mudunuri and Nagarajaram, 2007) for SSR identification, EasySSR has the same or greater accuracy than Misa, as shown through the benchmark tests in Table 2. Furthermore, EasySSR is a web-based service that offers more functionalities with the same analysis as command-line tools, identifies coding/non-coding regions, and performs the post-processing and data comparison instead of giving the user only the raw data as output.

Among the webtools, Imperfect SSR finder (Stieneke and Eujayl, 2007) and EasySSR are the only ones to be able to analyze perfect, imperfect, and compound SSR. However, even though Imperfect SSR finder has no cap for input size, it does not accept more than one FASTA file, does not compare samples, has no information in the output about flanking sequences or the SSR position in coding non-coding regions, and does not generate user-friendly outputs as charts.

An overall comparison of EasySSR and the most-cited 10 web tools for SSR mining shows that EasySSR clearly distinguishes itself by being a web tool that accepts for input both multi-FASTA and multiple FASTA files, in the same run, without a maximum size limit. Among all web tools, EasySSR is the only one to have the same features as command-line tools, being able to identify coding/non-coding information if an annotation file is uploaded, compare large datasets, and return processed outputs for online or local analyses.

A



B

EasySSR - Reports

Sequence	Genome Size (BP)	Total SSR	SSR per BP (%)	Total Coding	Total Non-Coding	Mono	Di	Tri	Tetra	Penta	Hexa	Total Perfect	Total Perfect Coding	Total Perfect Non-Coding	Perfect Mono	Perfect Di	Perfect Tri	Perfect Tetra	Perfect Penta	Perfect Hexa	Total Imperfect	Total Imperfect Coding	Tota Imp Non Cod
Ecoli_K12	4639675	106243	2.29	92911	13332	19088	6729	64381	11501	2926	1618	102532	89653	12879	19088	6451	62941	9704	2791	1557	3711	3258	453
Human_Atrophin1	4382	146	3.33	122	24	41	10	62	17	5	11	130	110	20	41	8	58	11	5	7	16	12	4
MTB_H37Rv	4411532	111583	2.53	102929	8654	6710	7194	80422	11618	3407	2232	105656	97441	8215	6710	6766	77670	9265	3143	2102	5927	5488	439
Plasmodium_Chr4	1204112	55693	4.63	30958	24735	18908	9241	18971	5311	2059	1203	45722	27168	18554	17377	5939	16967	3193	1412	834	9971	3790	618
Yeast_Chr4	1531918	40239	2.63	28159	12080	10387	2947	20767	4234	1278	626	38326	27011	11315	10289	2758	20086	3484	1152	557	1913	1148	765

FIGURE 8
Demonstration of EasySSR Reports from the batch comparison of perfect and imperfect SSR in five sequences with gene annotation: human atrophin1 gene, Plasmodium falciparum chromosome IV, yeast chromosome IV, *Mycobacterium tuberculosis* H37Rv, and *Escherichia coli* K12. **(A)** Comparison charts and **(B)** statistics table reports in print mode.

3.2.2 Benchmark testing against web servers and command-line tools

3.2.2.1 Intraspecific analysis for perfect SSR in prokaryotes, using only FASTA files as input

The benchmark results of this analysis are summarized in Table 2. Beier et al. (2017) did not include IMEX results in their comparison with Misa-Web because they reportedly could not execute the tool command-line mode due to operating system incompatibility. However, in the current analysis with EasySSR, a web tool that is IMEX based, the number of SSRs identified was greater than Misa-web, GMATo, Mreps, SciRoKo, ProGeRF, and TRF, and the analysis was conducted within the average time taken by the other programs, demonstrating that our algorithm has equal or higher sensibility with the same parameters, giving the user the outputs already processed in charts and tables in 5 s, as demonstrated through Figure 7, with interactive and detailed results.

Besides the raw amount of perfect SSR found, the EasySSR statistics table (Figure 7B) also gives the user categorized information about how many of the microsatellites found were Mono, Di, Tri, Tetra, Penta, and Hexanucleotide motifs. This information is also summarized visually into the graphs (Figure 7A). In Figure 7A, it is possible to notice that the graphs for imperfect SSRs are blank, due to the parameters set that searched

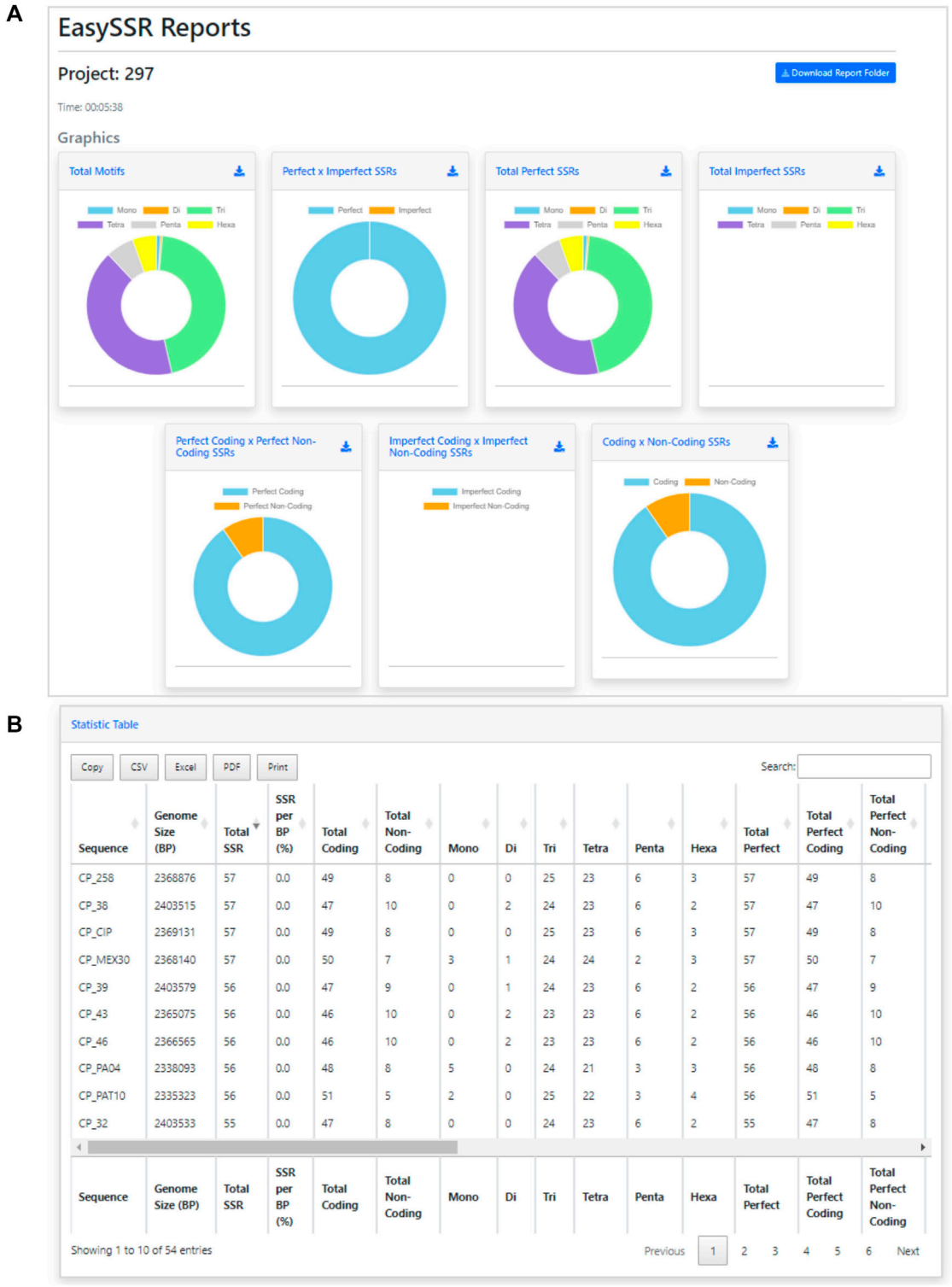


FIGURE 9
Easy SSR output screen from the large-scale analysis and comparison of perfect SSR in 54 complete genomes of *Corynebacterium pseudotuberculosis* with gene annotation. **(A)** Comparison charts and **(B)** statistics table reports ordered by total SSR.

for perfect SSR only. Moreover, in Figure 7A, the charts to compare the position of SSRs in coding/non-coding appear all in the same color, indicating that all SSRs were found in non-coding regions. This happens when no annotation file is uploaded by the user, in a way that the algorithm is set to consider everything in the FASTA file as non-coding by default.

TABLE 4 Perfect microsatellite identified for 54 complete genomes of *Corynebacterium pseudotuberculosis*.

Sequence	Accession	Biovar	Size (Mb)	Total PerfectSSR	Total coding	Total non-coding	Mono	Di	Tri	Tetra	Penta	Hexa
CP_04MAT	CP036469.1	Ovis	2.33801	53	49	4	1	0	24	22	3	3
CP_1002B	CP012837.1	Ovis	2.33831	54	49	5	2	0	24	22	3	3
CP_106A	CP003082.1	Equi	2.33835	54	48	6	0	0	24	21	6	3
CP_226	CP010889.1	Ovis	2.33783	53	50	3	0	0	25	21	3	4
CP_258	CP003540.3	Equi	2.33749	57	49	8	0	0	25	23	6	3
CP_262	CP012022.2	Equi	2.33757	48	44	4	0	0	22	23	1	2
CP_267	CP003407.1	Ovis	2.33790	54	50	4	1	0	25	21	3	4
CP_29156	CP010795.2	Ovis	2.33775	53	50	3	0	0	25	21	3	4
CP_31	CP003421.4	Equi	2.33727	53	47	6	0	0	24	23	4	2
CP_316	CP003077.2	Equi	2.33750	52	48	4	0	0	24	23	2	3
CP_32	CP015183.1	Equi	2.33730	55	47	8	0	0	24	23	6	2
CP_33	CP015184.1	Equi	2.33729	55	47	8	0	0	24	23	6	2
CP_34	CP015192.1	Equi	2.33733	55	47	8	0	0	24	23	6	2
CP_35	CP015185.1	Equi	2.33732	55	47	8	0	0	24	23	6	2
CP_36	CP015186.1	Equi	2.33734	54	46	8	0	0	23	23	6	2
CP_38	CP015187.1	Equi	2.33731	57	47	10	0	2	24	23	6	2
CP_38MAT	CP036457.1	Ovis	2.33771	53	48	5	2	0	24	21	3	3
CP_39	CP015188.1	Equi	2.33728	56	47	9	0	1	24	23	6	2
CP_43	CP015189.1	Equi	2.33756	56	46	10	0	2	23	23	6	2
CP_46	CP015190.1	Equi	2.33755	56	46	10	0	2	23	23	6	2
CP_48	CP015191.1	Equi	2.33735	55	46	9	0	1	23	23	6	2
CP_Cap1W	CP034411.1	Ovis	2.33817	53	49	4	1	0	24	22	3	3
CP_CAP3W	CP026500.1	Ovis	2.33818	52	49	3	0	0	24	22	3	3
CP_CAPJ4	CP026499.1	Ovis	2.33808	53	49	4	1	0	24	22	3	3
CP_CAPMI03	CP035717.1	Ovis	2.33812	51	48	3	0	0	23	22	3	3
CP_CIP	CP003061.3	Equi	2.33748	57	49	8	0	0	25	23	6	3
CP_Cp162	CP003652.3	Equi	2.33736	50	47	3	0	0	22	23	2	3
CP_E19	CP012136.1	Equi	2.33753	52	49	3	1	0	24	22	2	3
CP_E55	CP014341.1	Ovis	2.33829	55	51	4	2	0	25	23	2	3
CP_I19	CP002251.3	Ovis	2.33821	54	51	3	0	0	25	22	3	4
CP_I37	CP017384.1	Equi	2.33742	51	47	4	0	0	23	22	3	3
CP_MB11	CP013260.2	Equi	2.33741	52	48	4	0	0	24	23	2	3
CP_MB14	CP013261.1	Equi	2.33740	53	49	4	0	0	25	23	2	3
CP_MB20	CP016829.1	Equi	2.33739	54	50	4	1	0	24	24	2	3
CP_MB30	CP013262.2	Equi	2.33752	52	48	4	0	0	24	23	2	3
CP_MB66	CP013263.1	Equi	2.33737	53	49	4	0	0	24	24	2	3
CP_MEX1	CP017711.1	Ovis	2.33827	51	47	4	0	0	24	21	3	3
CP_MEX2	CP046644.1	Ovis	2.33809	51	47	4	0	0	24	21	3	3

(Continued on following page)

TABLE 4 (*Continued*) Perfect microsatellite identified for 54 complete genomes of *Corynebacterium pseudotuberculosis*.

Sequence	Accession	Biovar	Size (Mb)	Total PerfectSSR	Total coding	Total non-coding	Mono	Di	Tri	Tetra	Penta	Hexa
CP_MEX25	CP013697.1	Ovis	2.33813	55	50	5	1	0	26	21	3	4
CP_MEX29	CP016826.1	Ovis	2.33780	55	51	4	1	0	25	22	3	4
CP_MEX30	CP017291.1	Equi	2.33751	57	50	7	3	1	24	24	2	3
CP_MEX31	CP017292.1	Equi	2.33754	54	48	6	0	2	24	23	2	3
CP_OVID04	CP035640.1	Ovis	2.33810	51	48	3	0	0	24	21	3	3
CP_OVIOS02	CP035679.1	Ovis	2.33793	53	49	4	1	0	24	22	3	3
CP_OVIZ01	CP035678.1	Ovis	2.33781	52	48	4	1	0	24	21	3	3
CP_PA01	CP013327.1	Ovis	2.33777	53	49	4	1	0	25	21	3	3
CP_PA02	CP015309.1	Ovis	2.33834	51	48	3	0	0	23	22	3	3
CP_PA04	CP019587.1	Ovis	2.33773	56	48	8	5	0	24	21	3	3
CP_PA07	CP024457.1	Ovis	2.33820	51	48	3	0	0	24	21	3	3
CP_PAT10	CP002924.1	Ovis	2.33830	56	51	5	2	0	25	22	3	4
CP_PAT14	CP047603.1	Ovis	2.33825	54	51	3	0	0	25	22	3	4
CP_PAT16	CP046641.1	Ovis	2.33815	54	51	3	0	0	25	22	3	4
CP_PO22241	CP013698.1	Ovis	2.33816	53	49	4	1	0	25	21	3	3
CP_PO2695	CP012695.1	Ovis	2.33826	54	49	5	2	0	24	22	3	3
Total	54	Ovis = 28; Equi = 26	-	2,891	2,613	278	30	11	1,301	1,201	189	159

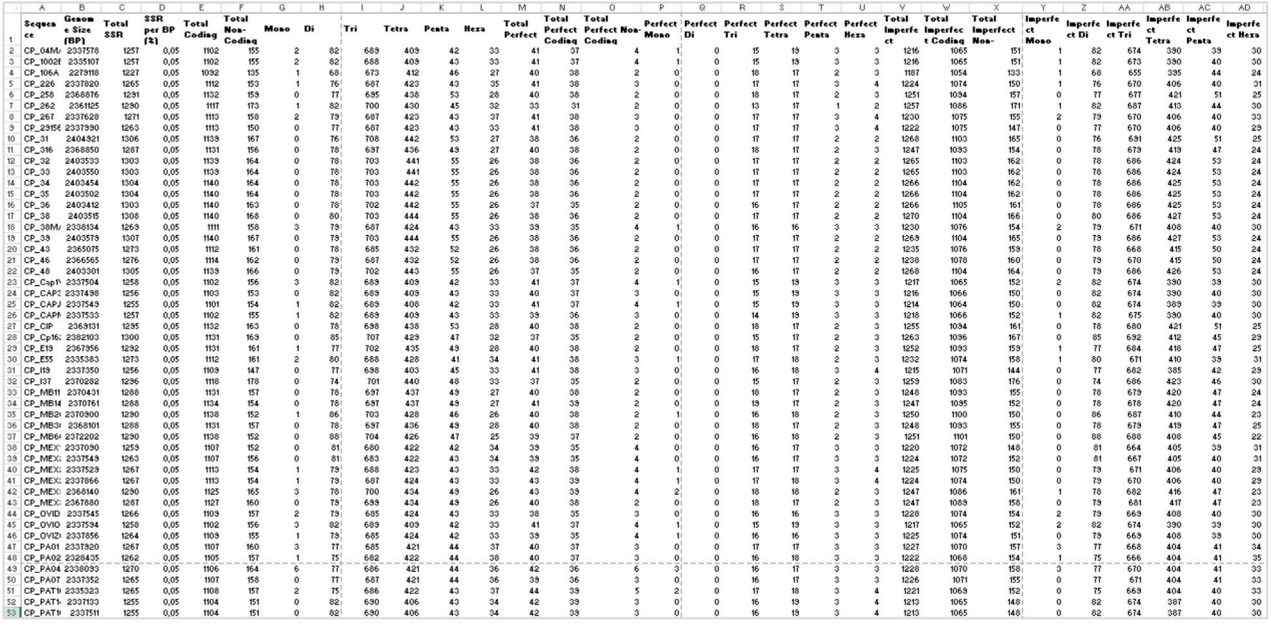


FIGURE 10
Demonstration of EasySSR Reports from the batch comparison of perfect and imperfect SSR in 54 sequences of *Corynebacterium pseudotuberculosis* with annotation. Statistics table reports in Excel mode optimized for visualization of the complete output with all columns and rows.

3.2.2.2 Interspecific analysis for imperfect SSR in prokaryotes and eukaryotes, using both FASTA and GenBank files as input

The benchmark test was carried out by running the "Dataset 2—IMEx" through the software tools EasySSR, TRF (Benson, 1999), Sputnik (Morgante et al., 2002), IMEx 1.0, and IMEx 2.1 (Mudunuri and Nagarajaram, 2007; Mudunuri et al., 2010a). We ran both versions of the IMEx program to compare the findings to version 1.0 tested in the article. Table 3 summarizes the findings, which were consistent with Mudunuri's original 2007 article.

IMEX 1.0 had already exceeded TRF and Sputnik in terms of sensibility and time since the 2007 article (Mudunuri and Nagarajaram, 2007). Many features were added to IMEX 2.1, which increased the analysis time slightly, although it is still less than the other tools evaluated. EasySSR is an online application that uses IMEx 2.1 for SSR mining; therefore, it has the same sensibility as this software and performs additional data analysis and output processing with friendly outputs on the web. Due to Internet speed and computational availability, EasySSR online analysis may be slightly slower than the standalone command-line IMEx 2.1; however, it still easily surpassed command-line TRF and Sputnik in terms of sensitivity and time benchmarks (Table 3). EasySSR compensates for any additional processing time spent by the automated results with post-processed information, saving the user time that would otherwise be spent during data tabulation and analysis.

As this analysis was conducted including imperfect and perfect SSRs and providing the GenBank annotation file as well, EasySSR outputs provided all the information in the graphics and tables regarding SSRs and their position in coding and non-coding regions, as demonstrated in Figure 8. In this way, besides the raw IMEx outputs, which are also available for download in the EasySSR outputs page for further analysis, the user can easily know the comparative proportion through the interactive charts for the whole sample of SSRs by coding/non-coding regions or motif classes, as perfect SSR, imperfect SSR, and in total (Figure 8A). The user can also run EasySSR with a single file per time in order to obtain individual charts for each genome.

Figure 8B depicts the "print" version of the statistics table, which is also available through a button on the EasySSR reports page alongside the "excel," "csv," "pdf," and "copy" alternative buttons that can be seen in Figure 7B. In this mode, the viewer can get a panoramic view, which includes extra columns that were previously hidden behind the scroll bar in the visualization. Because only perfect SSR were studied in the previous analysis, there was no need to split the total SSR into perfect and imperfect. However, because imperfection is now considered, more columns must be examined. The statistics table contains comprehensive information encompassing the total number of SSRs, along with subtotals for perfect and imperfect SSRs, coding and non-coding classifications, and the proportions of the motifs (Figure 8B).

3.2.3 Large-scale interspecific analysis for imperfect SSR, using both FASTA and GenBank files as input with default parameters

EasySSR was run two times for the dataset containing 54 complete genomes of *C. pseudotuberculosis (CP)*: i) With custom parameters, mining perfect SSR only, and ii) With default parameters, mining both perfect and imperfect SSR.

With EasySSR, which also runs IMEx as the microsatellite mining tool, it was possible to locate all SSR in coding and non-coding regions and to visualize the proportion through charts (Figures 4, 5) or generate new charts from the data available in the EasySSR statistic, motif frequency, and summary tables (Figure 6). The analysis for perfect SSR only was completed within 5 min and 38 s (Figure 9), while the analysis for perfect and imperfect took 8 min and 41 s (Figure 4). The complete output datasheets for perfect SSR and perfect/imperfect analysis of the 54 complete genomes of *C. pseudotuberculosis* are available in Supplementary Table S1.

The EasySSR quantitative results for perfect SSR were in concordance with those stated by Pinheiro et al. (2022), as demonstrated in Table 4, and the current analysis included further comparison of the motif classes proportions. In total, 2,891 perfect SSR, 2,613 in coding regions, and 278 in non-coding regions were found, with 30 mono, 11 di, 1,301 tri, 1,201 tetra, 189 penta, and 159 hexanucleotides as proportions demonstrated in Figure 9A and with data and accession numbers available in Table 4 ordered by sequence name. The genomes had an average incidence of 53,5 perfect SSRs. Most genomes have less than 57 SSRs, ranging from 48 (CP_262, *equi biovar*) to 57. CP_258, CP_38, CP_CIP and CP_MEX30 (*equi biovar*), were the only ones to have 57 perfect SSR, however the distribution of those microsatellites is not the same in all four sequences. As shown in Figure 9B, in CP_258 and CP_CIP, their distribution pattern (Simple Sequence Repeats Signature) is 49 SSR in coding to 8 SSR in non-coding regions, with 0 mono, 0 di, 25 tri, 23 tetra, 6 penta, and 3 hexanucleotides in both strains. Meanwhile, the distribution for CP_38 (2.33731 mb) is 47 coding/10 non-coding, with 0 mono, 2 di, 24 tri, 23 tetra, 6 penta, and 2 hexanucleotides, while the distribution for CP_MEX30 (2.33751 mb) was 50 coding/7 non-coding, 3 mono, 1 di, 24 tri, 24 tetra, 6 penta, and 3 hexanucleotides.

In the analysis where imperfect microsatellites were allowed, the Simple Sequence Repeats Signature changed. The total of the SSRs identified was 68,942 SSR, 60,390 in coding regions, and 8,552 in non-coding regions, with 50 mono, 4,268 di, 37,411 tri, 23,025 tetra, 2,524 penta, and 1,664 hexanucleotides, with a proportion of 2,146 perfect SSRs to 66,796 imperfect SSRs (Figure 4). The genomes had an average incidence of 40 perfect SSRs and 1,237 imperfect SSRs per genome, as shown in the data summarized in Figures 6, 10 through different visualization modes, with Figure 6B representing the output as shown in the EasySSR output page and Figure 10 showing the complete table ordered by sequence name for better comparison with Table 4 (Perfect SSRs output). The perfect SSRs found ranges from 33 (CP_262, *equi biovar*) to 44 (CP_PAT10, *ovis biovar*). CP_258, CP_CIP, CP_38, and CP_MEX30 had, respectively 40, 40, 38, and 43 perfect SSRs. The distribution of perfect SSRs was the same in CP_258 and CP_CIP with Mono: 0; Di: 0; Tri: 18; Tetra: 17; Penta: 2; and Hexa: 3. It is possible to notice that when mismatches were allowed in a tract, EasySSR through the IMEx algorithm could extend tracts that were previously interrupted by an imperfection and considered as perfect because it had passed the repetition cutoff when they were actually part of longer imperfect tracts; thus, the average amount of perfect SSRs per genome decreased from 53.5 to 40 in the analysis that included imperfections.

Laskar et al. (2021), (2022), Jilani and Ali (2022) used similar information about incidence, prevalence, composition, and localization in their studies of Simple Sequence Repeats Signature in viruses using IMEx. Those analyses might seem basic, but they require a lot of data tabulation before the tables are ready for analysis, a feature that is already automated by EasySSR. This is a small demonstration of the versatility of EasySSR output, which made this analysis possible in minutes due to the processed information given as a result, allowing the researcher to invest their time in further analysis that otherwise would be too time demanding.

EasySSR bar charts show the top 10 most-frequent motifs present in all the strains (Figure 5). They are interactive graphs that can be used to remove specific strains from visualization or verify how many times that specific motif was found in different loci in that genome. In this way, it is possible to verify that the GCT, TGC, and GCA were present in all the 54 genomes used by Pinheiro et al. (2022). The amount of GCT motifs present in a genome varied from 26 to 37 different loci, for example, (Figure 5A). It might present itself as a useful shortcut tool to marker development. Pinheiro et al. (2022) identified CAC and GGAA as putative markers based on their differential localization in the biovars. EasySSR did not reach the same results for those markers as it has a different approach, where the bar charts demonstrate quantitatively how many times the motif appears in each genome and ranks them based on how many genomes of the dataset are present, aiming to find motifs that are common to all sequences. However, EasySSR can also be used for analysis, such as the one conducted by Pinheiro et al. (2022), as their EasySSR summary table contains information about the motif, iteration, and position (start and end), and it is easily downloadable in friendly formats such as "xlsx" and ".csv" that can be imported for further analysis using others statistic tools present in the R programming language, for example,. In this way, EasySSR outputs are versatile and can be used as a guide for visual analysis through the interactive graphs or processed by other tools with any approach the user wants.

4 Conclusion

Despite the versatility of the existing web tools for microsatellite analysis, EasySSR presents an innovative web technology that implements the popular IMEx 2.1 algorithm under novel settings, with a friendly interface suitable for experts and non-experienced scientists to realize online SSR analysis with the same accuracy and features as command-line tools. Easy SSR automatizes the SSR mining in batch analysis, for small or large datasets, from receiving many FASTA input files, converting, generating raw SSR outputs for each file, and processing those outputs in a comparative approach, with additional comprehensible results summarized into interactive charts and tables, giving the user the results ready for further analysis in minutes and reducing a significant amount of time worth of data tabulation.

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.

Author contributions

SA and RR conceived the idea of the program and together with VF developed the tool. SA, CD, and AS evaluated the biological and computational information and defined the functions to be inserted. All authors contributed to the article and approved the submitted version.

Funding

This work has been supported by the CNPq (National Council for Scientific and Technological Development) project #312316/2022-4, Secretary of State for Science, Technology, and Professional and Technological Education (SECTET), and Dean's Office for Research and Graduate Studies/Federal University of Pará–PROPESP/UFPA (PAPQ). PROCAD-AM (NATIONAL PROGRAM FOR ACADEMIC COOPERATION IN THE AMAZON) from CAPES, under project No. 88881.200563/2018-01.

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/fgene.2023.1228552/full#supplementary-material

References

Beier, S., Thiel, T., Münch, T., Scholz, U., and Mascher, M. (2017). MISA-Web: a web server for microsatellite prediction. *Bioinformatics* 33, 2583–2585. doi:10.1093/bioinformatics/btx198

Benson, G. (1999). Tandem repeats finder: a program to analyze DNA sequences. *Nucleic Acids Res.* 27, 573–580. doi:10.1093/nar/27.2.573

Biswas, M. K., Natarajan, S., Biswas, D., Nath, U. K., Park, J.-I., and Nou, I. (2018). Lsat: liliaceae simple sequences analysis tool, a web server. *Bioinformation* 14, 181–182. doi:10.6026/97320630014181

Boeva, V., Regnier, M., Papatsenko, D., and Makeev, V. (2006). Short fuzzy tandem repeats in genomic sequences, identification, and possible role in regulation of gene expression. *Bioinformatics* 22, 676–684. doi:10.1093/bioinformatics/btk032

da Maia, L. C., Palmieri, D. A., de Souza, V. Q., Kopp, M. M., de Carvalho, F. I. F., and Costa de Oliveira, A. (2008). SSR locator: tool for simple sequence repeat discovery integrated with primer design and PCR simulation. *Int. J. Plant Genomics* 2008, 412696–412699. doi:10.1155/2008/412696

Das, R., Arora, V., Jaiswal, S., Iquebal, M., Angadi, U., Fatma, S., et al. (2019). PolyMorphPredict: a universal web-tool for rapid polymorphic microsatellite marker discovery from whole genome and transcriptome data. *Front. Plant Sci.* 9, 1966. doi:10.3389/fpls.2018.01966

Delgrange, O., and Rivals, E. (2004). Star: an algorithm to search for tandem approximate repeats. *Bioinformatics* 20, 2812–2820. doi:10.1093/bioinformatics/bth335

Django Software Foundation (2023). Django makes it easier to build better web apps more quickly and with less code. Available at: https://www.djangoproject.com/.

Galasso, I., and Ponzoni, E. (2015). In Silico Exploration of Cannabis sativa L. Genome for Simple Sequence Repeats (SSRs). *Am. J. Plant Sci.* 06, 3244–3250. doi:10.4236/ajps.2015.619315

Jilani, M. G., and Ali, S. (2022). Assessment of simple sequence repeats signature in hepatitis E virus (HEV) genomes. *J. Genet. Eng. Biotechnol.* 20, 73. doi:10.1186/s43141-022-00365-w

Kim, J., Choi, J.-P., Ahmad, R., Oh, S.-K., Kwon, S.-Y., and Hur, C.-G. (2012). Risa: a new web-tool for rapid identification of SSRs and analysis of primers. *Genes Genomics* 34, 583–590. doi:10.1007/s13258-012-0032-x

Kofler, R., Schlötterer, C., and Lelley, T. (2007). SciRoKo: a new tool for whole genome microsatellite search and investigation. *Bioinformatics* 23, 1683–1685. doi:10.1093/bioinformatics/btm157

Kolpakov, R., Bana, G., and Kucherov, G. (2003). mreps: efficient and flexible detection of tandem repeats in DNA. *Nucleic Acids Res.* 31, 3672–3678. doi:10.1093/nar/gkg617

Laskar, R., Jilani, M. G., and Ali, S. (2021). Implications of genome simple sequence repeats signature in 98 Polyomaviridae species. *3 Biotech.* 11, 35. doi:10.1007/s13205-020-02583-w

Laskar, R., Jilani, M. G., Nasrin, T., and Ali, S. (2022). Microsatellite signature of reference genome sequence of SARS-CoV-2 and 32 species of coronaviridae family. *Int. J. Infect.* 9, e122019. doi:10.5812/iji-122019

Leclercq, S., Rivals, E., and Jarne, P. (2007). Detecting microsatellites within genomes: significant variation among algorithms. *BMC Bioinforma.* 8, 125. doi:10.1186/1471-2105-8-125

Lim, K. G., Kwoh, C. K., Hsu, L. Y., and Wirawan, A. (2013). Review of tandem repeat search tools: a systematic approach to evaluating algorithmic performance. *Brief. Bioinform.* 14, 67–81. doi:10.1093/bib/bbs023

Lopes, R. S., Moraes, W. J. L., Rodrigues, T. D. S., and Bartholomeu, D. C. (2015). ProGeRF: proteome and genome repeat finder utilizing a fast parallel hash function. *Biomed. Res. Int.* 2015, 394157–394159. doi:10.1155/2015/394157

Martins, W. S., Lucas, D. C. S., Neves, K. F. S., and Bertioli, D. J. (2009). WebSat - a web software for MicroSatellite marker development. *Bioinformation* 3, 282–283. doi:10.6026/97320630003282

Mathur, M., Tyagi, S., and Kataria, P. (2020). A comparative study of various simple sequence repeats identification tools using Aspergillus fumigatus genome. *J. Bioinfo Comp. Genom* 3, 1–13. doi:10.17303/jbcg.2020.3.102

Merkel, A., and Gemmell, N. (2008). Detecting short tandem repeats from genome data: opening the software black box. *Brief. Bioinform.* 9, 355–366. doi:10.1093/bib/bbn028

Morgante, M., Hanafey, M., and Powell, W. (2002). Microsatellites are preferentially associated with nonrepetitive DNA in plant genomes. *Nat. Genet.* 30, 194–200. doi:10.1038/ng822

Mudunuri, S. B., Kumar, P., Rao, A. A., Pallamsetty, S., and Nagarajaram, H. A. (2010a). G-IMEx: a comprehensive software tool for detection of microsatellites from genome sequences. *Bioinformation* 5, 221–223. doi:10.6026/97320630005221

Mudunuri, S. B., and Nagarajaram, H. A. (2007). IMEx: imperfect microsatellite extractor. *Bioinformatics* 23, 1181–1187. doi:10.1093/bioinformatics/btm097

Mudunuri, S. B., Rao, A. A., Pallamsetty, S., and Nagarajaram, H. A. (2010b). "Comparative analysis of microsatellite detecting software: a significant variation in results and influence of parameters," in *Proceedings of the international symposium on biocomputing*. Editor D. Tulpan (New York, NY, USA: ACM), 1–7. doi:10.1145/1722024.1722068

Oliveira, E. J. de, Dantas, J. L. L., Castellen, M. S., and Machado, M. D. (2008). Identificação de microssatélites para o mamoeiro por meio da exploração do banco de dados de DNA. *Rev. Bras. Frutic.* 30, 841–845. doi:10.1590/s0100-29452008000300049

Parisi, V., De Fonzo, V., and Aluffi-Pentini, F. (2003). String: finding tandem repeats in DNA sequences. *Bioinformatics* 19, 1733–1738. doi:10.1093/bioinformatics/btg268

Pinheiro, K. C., Gois, B. V. A., Nogueira, W. G., Araújo, F. A., Queiroz, A. L. C., Cardenas-Alegria, O., et al. (2022). *In silico* approach to identify microsatellite candidate biomarkers to differentiate the biovar of Corynebacterium pseudotuberculosis genomes. *Front. Bioinforma.* 2, 931583. doi:10.3389/fbinf.2022.931583

Sharma, P. C., Grover, A., and Kahl, G. (2007). Mining microsatellites in eukaryotic genomes. *Trends Biotechnol.* 25, 490–498. doi:10.1016/j.tibtech.2007.07.013

Sousa, A. L., Maués, D., Lobato, A., Franco, E. F., Pinheiro, K., Araújo, F., et al. (2018). PhageWeb – web interface for rapid identification and characterization of prophages in bacterial genomes. *Front. Genet.* 9, 1–7. doi:10.3389/fgene.2018.00644

Sreenu, V. B. (2003). MICdb: database of prokaryotic microsatellites. *Nucleic Acids Res.* 31, 106–108. doi:10.1093/nar/gkg002

Stieneke, D. L., and Eujayl, I. A. (2007). Imperfect SSR finder. Available at: http://ssr.nwisrl.ars.usda.gov/.

Tarailo-Graovac, M., and Chen, N. (2009). Using RepeatMasker to identify repetitive elements in genomic sequences. *Curr. Protoc. Bioinforma.* 25, 4.10.1–4.10.14. doi:10.1002/0471250953.bi0410s25

Thiel, T., Michalek, W., Varshney, R., and Graner, A. (2003). Exploiting EST databases for the development and characterization of gene-derived SSR-markers in barley (Hordeum vulgare L). *Theor. Appl. Genet.* 106, 411–422. doi:10.1007/s00122-002-1031-0

Thurston, M., and Field, D. (2005). Msatfinder: detection and characterisation of microsatellites. CEH oxford, mansf. Road, oxford OX1 3SR. Available at: http://www.genomics.ceh.ac.uk/msatfinder/.

Wang, X., Lu, P., and Luo, Z. (2013). GMATo: a novel tool for the identification and analysis of microsatellites in large genomes. *Bioinformation* 9, 541–544. doi:10.6026/97320630009541

Wexler, Y., Yakhini, Z., Kashi, Y., and Geiger, D. (2004). "Finding approximate tandem repeats in genomic sequences," in Proceedings of the eighth annual international conference on Computational molecular biology - RECOMB '04, San Diego California USA, March 27 - 31, 2004 (New York, New York, USA: ACM Press), 223–232. doi:10.1145/974614.974644

Yang, W., Zheng, J., Jia, B., Wei, H., Wang, G., and Yang, F. (2018). Isolation of novel microsatellite markers and their application for genetic diversity and parentage analyses in sika deer. *Gene* 643, 68–73. doi:10.1016/j.gene.2017.12.007

You, F. M., Huo, N., Gu, Y. Q., Luo, M., Ma, Y., Hane, D., et al. (2008). BatchPrimer3: a high throughput web application for pcr and sequencing primer design. *BMC Bioinforma.* 9, 253. doi:10.1186/1471-2105-9-253

frontiers | Frontiers in Genetics

TYPE Technology and Code
PUBLISHED 15 September 2023
DOI 10.3389/fgene.2023.1219297

OPEN ACCESS

EDITED BY
Lei Chen,
Shanghai Maritime University, China

REVIEWED BY
Juan P. Cardenas,
Major university, Chile
Abasiofiok Ibekwe,
United States Department of Agriculture (USDA), United States

*CORRESPONDENCE
Xiao Liang,
xliangvt@vt.edu

RECEIVED 08 May 2023
ACCEPTED 01 September 2023
PUBLISHED 15 September 2023

CITATION
Liang X, Zhang J, Kim Y, Ho J, Liu K, Keenum I, Gupta S, Davis B, Hepp SL, Zhang L, Xia K, Knowlton KF, Liao J, Vikesland PJ, Pruden A and Heath LS (2023), ARGem: a new metagenomics pipeline for antibiotic resistance genes: metadata, analysis, and visualization. *Front. Genet.* 14:1219297. doi: 10.3389/fgene.2023.1219297

ARGem: a new metagenomics pipeline for antibiotic resistance genes: metadata, analysis, and visualization

Xiao Liang[1]*, Jingyi Zhang[1], Yoonjin Kim[1], Josh Ho[1], Kevin Liu[1], Ishi Keenum[2], Suraj Gupta[3], Benjamin Davis[2], Shannon L. Hepp[2], Liqing Zhang[1], Kang Xia[4], Katharine F. Knowlton[5], Jingqiu Liao[2], Peter J. Vikesland[2], Amy Pruden[2] and Lenwood S. Heath[1]

[1]Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA, United States, [2]Department of Civil and Environmental Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA, United States, [3]Interdisciplinary PhD Program in Genetics, Bioinformatics, and Computational Biology, Virginia Polytechnic Institute and State University, Blacksburg, VA, United States, [4]School of Plant and Environmental Science, Virginia Polytechnic Institute and State University, Blacksburg, VA, United States, [5]Department of Dairy Science, Virginia Polytechnic Institute and State University, Blacksburg, VaA, United States

Antibiotic resistance is of crucial interest to both human and animal medicine. It has been recognized that increased environmental monitoring of antibiotic resistance is needed. Metagenomic DNA sequencing is becoming an attractive method to profile antibiotic resistance genes (ARGs), including a special focus on pathogens. A number of computational pipelines are available and under development to support environmental ARG monitoring; the pipeline we present here is promising for general adoption for the purpose of harmonized global monitoring. Specifically, ARGem is a user-friendly pipeline that provides full-service analysis, from the initial DNA short reads to the final visualization of results. The capture of extensive metadata is also facilitated to support comparability across projects and broader monitoring goals. The ARGem pipeline offers efficient analysis of a modest number of samples along with affordable computational components, though the throughput could be increased through cloud resources, based on the user's configuration. The pipeline components were carefully assessed and selected to satisfy tradeoffs, balancing efficiency and flexibility. It was essential to provide a step to perform short read assembly in a reasonable time frame to ensure accurate annotation of identified ARGs. Comprehensive ARG and mobile genetic element databases are included in ARGem for annotation support. ARGem further includes an expandable set of analysis tools that include statistical and network analysis and supports various useful visualization techniques, including Cytoscape visualization of co-occurrence and correlation networks. The performance and flexibility of the ARGem pipeline is demonstrated with analysis of aquatic metagenomes. The pipeline is freely available at https://github.com/xlxlxlx/ARGem.

KEYWORDS

antibiotic resistance genes, workflow, metagenomics, bioinformatics, genome annotation

1 Introduction

Antibiotic resistance poses a significant risk to human health. Antibiotic resistance genes (ARGs) encode resistance to antibiotics and can be carried in the bacterial chromosome or on mobile genetic elements (MGEs). ARGs are of greatest concern to human health, especially when they are found in known or emerging pathogens (Vikesland et al., 2019). The need for monitoring of ARGs in the environment, including water resources and agricultural production systems, is increasingly being recognized. Such environments play an important ecological role in propagation of ARGs. The ARGs can emanate from anthropogenic sources or from natural environments themselves, serving as facilitators of horizontal gene transfer (HGT) (Maiden, 1998; Barlow, 2009; Aminov, 2011; Lerminiaux and Cameron, 2019). HGT can contribute to expansion of the general reservoir of ARGs carried across environmental microbiomes, influencing human and animal pathogens, inducing new mechanisms of antimicrobial resistance. Metagenomics, the study of DNA extracted across the microbial community representing the environment of interest, has arisen as a promising approach to profiling ARGs and other microbial entities of concern, such as human pathogens (Koonin, 2018; Chiu and Miller, 2019). Environmental metagenomics has shown promise for tracking shifts in ARG and pathogen markers in the environment with time and in response to various disturbances and inputs (Berglund et al., 2019; de Abreu et al., 2021). Thus, metagenomics is being proposed as an efficient means of comprehensive surveillance of ARGs and pathogens across the One Health spectrum (Shen et al., 2021).

Contemporary environmental metagenomic data sets typically consist of a number of short read sequence files, typically generated by Illumina sequencing producing files ranging in size up to 100 Gbp (Gigabase pairs) (Davis et al., 2023), each representing either a Processing such datasets requires significant computational analysis. This typically needs to be organized in a bioinformatics pipeline that consists of selected software tools, which are mutually connected custom scripts. These scripts are usually written in programming languages such as Python 3 (Siegwald et al., 2017; Breitwieser et al., 2019), and composing such scripts to construct a bioinformatics pipeline can be challenging for non-expert users.

Many metagenomic analysis pipelines exist with much variation. However, the goal of a typical pipeline is to identify microbial taxa and genes of interest in a subset of samples, and to estimate their abundances. Further analysis of the annotation is often left to specific tools selected by the researcher. A pipeline may assemble the reads into contigs to allow identification of complete or nearly complete genes and to improve resolution for annotation. A classic metagenomics pipeline is the MG-RAST server, which is designed to process numerous samples on high-performance computing clusters (Meyer et al., 2008). A number of more recent pipelines (which we briefly review here) are available for a researcher to install and execute on their own computational resources (Uritskiy et al., 2018; Clarke et al., 2019; Dong and Strous, 2019; Tamames and Puente-Sanchez, 2019; Eng et al., 2020; Grieb et al., 2020). MetaWRAP employs binning and reassembly steps to obtain improved annotation (Uritskiy et al., 2018). SqueezeMeta concentrates on simultaneously assembling multiple metagenome data sets along with binning to enhance the identification of low-abundance taxa and genes (Tamames and Puente-Sanchez, 2019). MetaErg provides graphical summaries of the annotated contigs to support visual confirmation of contig quality (Dong and Strous, 2019). Sunbeam emphasizes a flexible pipeline framework that, in typical use, does not require the researcher to provide extensive run-time parameters (Clarke et al., 2019). Grieb et al. (Grieb et al., 2020) developed a pipeline explicitly tailored for research on marine plankton. Finally, MetaLAFFA is a flexible metagenomic analysis pipeline targeted to distributed computing environments (Eng et al., 2020).

A common limitation among the pipelines is a lack of integrated tools for additional analysis and visualization beyond basic annotation. Moreover, these pipelines do not provide flexible input, which results in a disincentive to data sharing and greatly detracts from the overall utility of the data. Metadata, which is the data describing properties (e.g., DNA extraction method and sample environment) of the sample, is nowadays commonly provided along with the sample sequences. Lack of extensive provision and sharing of metadata diminishes the ability to perform analyses that harness the power of metadata to support predictive modeling of environmental metagenomes. This deficiency in metadata sharing also detracts from encouraging reporting of comparable data, which is a critical need for the broader goal of large-scale environmental ARG monitoring. While researchers might recognize the importance of the extensive metadata that they collect for each sample, the actual types of metadata captured can vary greatly across research projects (Goncalves and Musen, 2019; Martinez-Romero et al., 2019). As one effort to remedy the situation, the National Center for Biotechnology Information (NCBI) (Sayers et al., 2019) collects a set of required metadata for each sample uploaded to resources, such as BioProject and BioSample (Federhen et al., 2014; Martinez-Romero et al., 2019), while still allowing for flexible column addition and following the minimum information about any (x) sequence (MIxS) guidelines (Yilmaz et al., 2011). However, comparing data across different projects remains a challenging task when using NCBI metadata.

Another notable framework, not limited to metagenomics analysis, is Galaxy (Jalili et al., 2020). Galaxy is a platform developed for flexible workflows that can be customized for bioinformatics tasks, with an open-source framework available for customization. Several pipelines have been developed using the Galaxy framework for various metagenomics tasks (Pilalis et al., 2012; Yang et al., 2016; Batut et al., 2018). Among them, only a few have aimed to develop an integrated pipeline that performs tasks beyond annotation. Additionally, most of these pipelines were not specifically designed for ARG detection tasks or for addressing the issue of customizing metadata in different environments.

Towards addressing the aforementioned issues, we present ARGem pipeline. This locally deployable pipeline supports ARG annotation as well as the capture of a flexible set of metadata, which will encourage comprehensive data sharing and be ultimately accessible to support more sophisticated future analysis after annotation is complete. To achieve this purpose, users are provided with a simple spreadsheet with required and recommended metadata fields and standardized units. Users complete the spreadsheet and submit it as input to create an ARGem project, in which the data are stored in a relational database that can be further cross-analyzed.

Key analytical tools and capabilities that are commonly applied for metagenomic-based ARG monitoring have been built into the ARGem pipeline, extending data analysis beyond the annotation of taxa and ARGs to include statistical analysis and ARG co-occurrence and correlation networks. The resulting outputs can culminate in a wide range of custom visualizations to support comparisons across samples and projects, as well as tables summarizing the results in tabulated format to support additional analysis. As detailed in Section *Assembly and Annotation*, we have extensively examined the bioinformatics components of the ARGem pipeline. In particular, we prioritized comprehensive databases for ARGs and MGEs annotation. One comparable pipeline is our own MetaStorm server (Arango-Argoty et al., 2016), which is only available as a Web service to execute on the computational resources of an individual research lab, which allows extendability of ARGem with new capabilities. PathoFact (de Nies et al., 2021) is a resource specialized in the prediction of ARGs and pathogens and make uses of our DeepARG resource (Arango-Argoty et al., 2018). However, PathoFact does not have the flexibility to incorporate or update reference databases other than the provided options, which were released prior to 2021. Also, PathoFact does not handle the assembly step and requires pre-assembled contigs as the input, prioritizing post-assembly analysis rather than a full sequence-to-analysis pipeline. PathoFact depends on Miniconda to guaranteee compatibility with specific versions of Snakemake and Python, making it convenient for users to install and use at the time of release, but may later lead to obsolescence compared to software with such dependency.

Overall, ARGem is a locally deployable pipeline which addresses many of the needs identified above through a user-friendly, full-service pipeline for ARG analysis of environmental metagenomic data with enhanced metadata capture and normalization to facilitate comparison within and across studies. In the *Method* section, we describe in detail the tools and methods employed in the ARGem pipeline. In Section *Results*, we describe the overall workflow of the pipeline and the general mechanism for each step, as well as demonstrate the value of our ARGem pipeline with a number of example runs utilizing metagenomic samples relevant to aquatic environments. Sections *Discussion* and *Conclusion* emphasize the strengths of our current implementation and identify potential paths for future extensions.

2 Methods

The ARGem pipeline integrates a number of tools implemented as individual modules that can be used within the pipeline or independently. Detailed descriptions are included for task scheduling, the Luigi workflow builder (Luigi Development Team, 2020), data retrieval, reference databases for annotations, assembly and annotation, analysis, visualization and the relational database.

2.1 Task scheduling

The ARGem pipeline consists of a sequence of tasks and employs a task scheduling mechanism that handles the distributed resources on multiple servers. This scheduling strategy is adequate for the computational resources of a typical lab. By maintaining a straightforward and concise task scheduling system, we intend to keep the system at lab scale and make it convenient for most researchers to use.

Specifically, we use the batch command in Linux. The batch command implements internal queues to manage and execute tasks in a manner that adapts execution demand to system capabilities, maintaining a ceiling on system load. If the job exits with an error, batch is used to catch the exception, and ARGem sends an email notification to the user email address stored in the database associated with the task. If the job completes successfully, the system also sends out an email notifying the user of the completion of the task.

2.2 Luigi workflow builder

Some of the tasks employed by ARGem are particularly time-demanding, such as sequence assembly and annotation. Such tasks can be especially demanding for analysis of environmental metagenomes, which tend to be particularly complex. In such cases, it is useful to incorporate a built-in workflow to handle the execution of tasks and deal with computational issues typically associated with long-running processes, such as error handling and status visualization. For ARGem, the Luigi package for Python (Luigi Development Team, 2020) is used by the back end to define tasks and chain them together to construct a workflow for the pipeline, as well as managing the scheduling of tasks, handling errors, and visualizing the status of the pipeline.

Luigi manages multiple tasks in the workflow by assigning them to different classes and drivers. Each class is designed to execute a particular task, such as short reads annotation or co-occurrence network analysis. Once the Luigi task classes are defined, they are aligned with each other in a workflow by indicating the dependencies between pairs of modules. Tasks without direct or indirect dependency on each other can be run in parallel, depending on how much resources the scheduler allocate for them. Figure 1A shows a generic Luigi workflow. In ARGem, all the Luigi modules are aligned linearly with a potential change on paralleling short read annotation with contig assembly and annotation, if needed.

2.3 Data retrieval from public websites

ARGem provides automatic raw sequence data retrieval from the public NCBI database (Sayers et al., 2022) through SRA toolkit (SRA Toolkit Development Team, 2022). The ARGem input spreadsheet contains an SRA number field in which the user can indicate the SRA or SRR number of the sample. The sample numbers then allow the SRA Toolkit to retrieve raw sequence data samples in `*.fasta` or `*.fastq` format. For the uploaded SRA numbers via the input spreadsheet, ARGem checks whether the SRA project numbers are accessible a suitable format through a pre-download. Invalid SRA numbers or those associated with incorrect formats are logged to a designated log file. Upon completion or interruption of the pipeline, these SRA numbers are then reported to the user in an email notification.

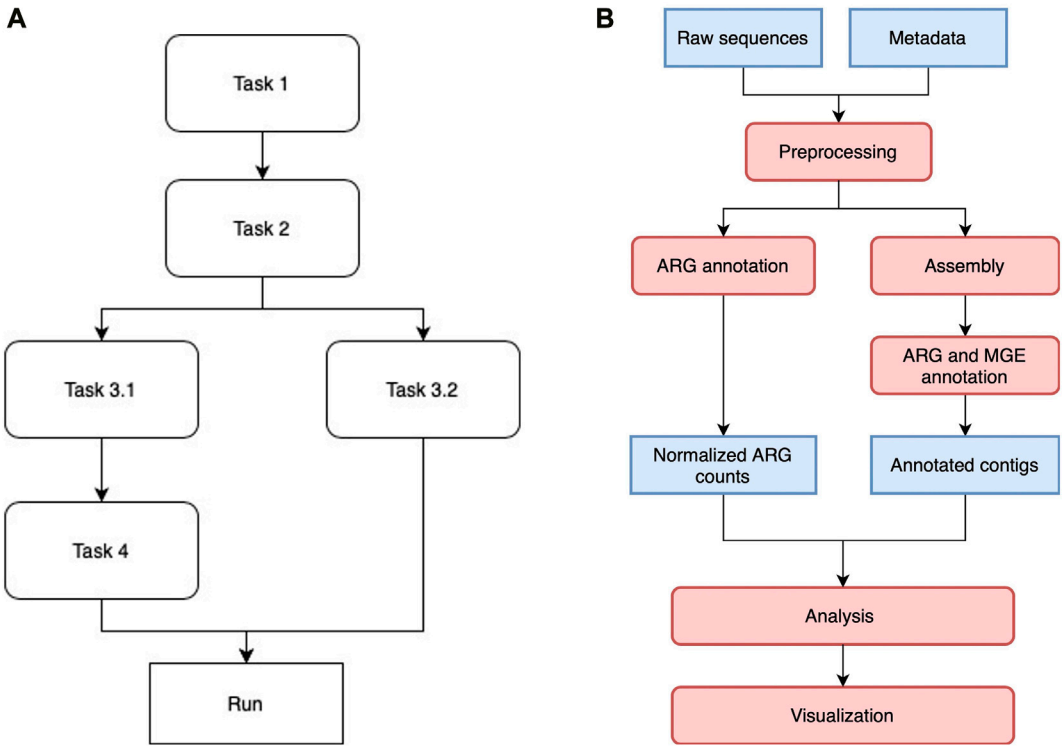


FIGURE 1
(A) A sample Luigi workflow. The workflow can consist of both linear and parallel tasks. A task that depends on a previous one will not be executed until all the dependencies have been completed. Tasks without direct or indirect dependency on each other can be executed in parallel if resource permits. **(B)** ARGem workflow. The ARGem pipeline automatically processes the raw sequences after a list of SRA accession numbers are submitted through a metadata spreadsheet. After preprocessing, the raw sequences go through two different branches: 1) short reads matching to generate normalized ARG counts, and 2) contig annotations against ARG and MGE reference databases. The results generated can then be passed to the integrated analysis and visualization tools. The default normalization methods built in the pipeline are 16S rRNA, TPM, and FPKM. 1) Blue rectangles indicate data and 2) red rounded rectangles indicate processing steps.

TABLE 1 An evaluation of assemblers on our server. In total one reclaimed waste water sample (water sample 1), one final treated biosolids sample (water sample 2), and two raw sewage and treated wastewater samples (water sample 3 and 4), were used to evaluate the performance of assemblers on our server. Note that the samples used here are different from those presented in Section *Results*. The size column shows the sizes of sample sequence files in gigabytes. Length indicates the sequence length of each sample sequence data. Time shows the total hours required to assemble the metagenomic data generated from a given sample. Percent of CPU, maximum resident set size and major page faults shows metrices reported by *time* command during the process.

Sample	Size (GB)	Length	Assembler	Time (hr)	Percent of CPU (%)	Maximum resident set size (KB)	Major page faults
Water1	5.91	108	MetaSPAdes	4 : 05: 43	1,147	46328252	31
Water1	5.91	108	IDBA-UD	2 : 47: 37	3130	32219196	1
Water1	5.91	108	MegaHIT	0 : 33: 47	3118	5369920	4
Water2	1.52	92	IDBA-UD	0 : 21: 20	2999	8617580	1
Water2	1.52	92	MegaHIT	0 : 05: 37	3109	1399316	1
Water2	1.52	92	MetaSPAdes	0 : 37: 23	1090	11654044	1
Water3	4.57	202	MegaHIT	0 : 43: 20	3402	4125444	0
Water3	4.57	202	MetaSPAdes	2 : 25: 05	1114	37923596	1
Water4	5.91	202	MegaHIT	0 : 54: 59	3,398	5014884	3
Water4	5.91	202	MetaSPAdes	3 : 13: 46	1,116	42655912	25

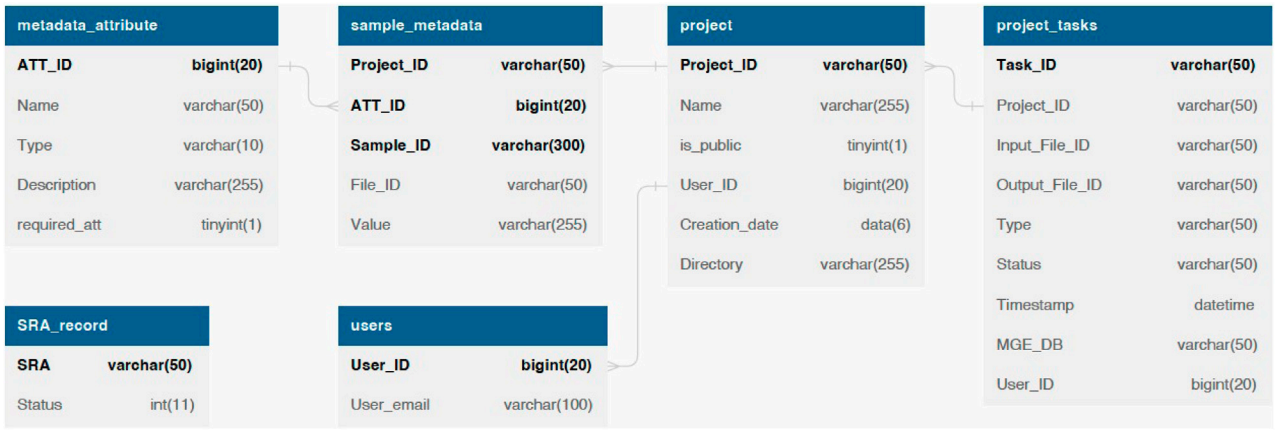


FIGURE 2
The database schema of ARGem. ARGem supports custom metadata attributes and various data processing parameters. Mandatory information including the SRA number and user information are reflected in database tables as *NOT NULL* fields. Optional fields are not required and can be set to a default value.

Once the accession and format verification is complete, ARGem begins the SRA sample retrieval process where the raw sequence files are downloaded individually for each SRA number. The retrieval of each sample is initialized with a query to the accession-size of the SRA project numbers through the SRA Toolkit to ensure that the size of a single sample is lower than the hard limit, which is set by default to be 70 gigabytes. In the case where the SRA sample is above the size limit, an error can be raised and logged accordingly by ARGem pipeline. Once the size verification is complete, the SRA sample is prefetched via the SRA Toolkit in `*.sra` format and then converted into `*.fastq` format. For paired end samples, the file format conversion process is split to convert each SRA sample into a paired files for assembly.

After all the raw sequence files are retrieved from NCBI website (Sayers et al., 2022), ARGem will initiate a post-download validation on the retrieved raw `*.fastq` sequence data files to validate their data integrity. Upon completion, annotation and assembly tasks for the valid samples will be scheduled according to the Luigi workflow.

2.4 ARG and MGE databases

The pipeline design of ARGem offers easy and flexible updates and interchanges for databases. Once a new ARG or MGE database is converted into a fasta file and a proper format for the annotation tool, ARGem redirects assembly and annotation tasks into the new databases. Default ARG and MGE databases were selected based on how widely they are used for metagenomic analysis, with a preference for databases that are frequently updated.

To annotate the raw sequences and assembly results into ARGs, ARGem integrates the current Comprehensive Antibiotic Resistance Database (CARD) (Alcock et al., 2020) as the default reference database, while the users have the option to use other databases at their choice.

ARGem utilizes three databases for MGEs: Mobile-OG (Brown et al., 2022), NanoARG (Arango-Argoty et al., 2019), and Parnanen et al. (Parnanen et al., 2018). The Mobile-OG database is a recently published database aiming to mitigate the high positive rates originated from accessory genes that are temporarily associated with the MGEs. The goal of the database is to provide high-quality annotations and annotations derived exclusively through bioinformatic evidence. NanoARG is a database that has been particularly insightful in identifyimg ARGs in sequences of varying lengths and a range of sequencing error rates. NanoARG is an integration of two data sets, NCBI and integron-integrase (I-VIP) database (Zhang et al., 2018). In the NanoARG database, MGE sequences have been extracted from NCBI using keywords such as “transposase,” “transposon,” “integrase,” “integron,” and “recombinase”, following the method described in (Forsberg et al., 2014). The I-VIP database focuses on comprehensive information on class 1 integrons. After extracting the MGE sequences from NCBI, the integrases of class 1 integrons have then been extracted from I-VIP database and added into the NanoARG database (Arango-Argoty et al., 2019). The Parnanen et al. MGE database (Parnanen et al., 2018) was created with a focus on mother-infant MGE sharing, providing a unique perspective and addition to the existing MGE research. This database was constructed by fetching coding sequences for genes that were annotated as IS*, ISCR*, intI1, int2, istA*, istB*, qacEdelta, tniA*, tniB*, tnpA* or Tn916 transposon open reading frames (ORFs). The genes were either sourced from the NCBI nucleotide database, or from the PlasmidFinder database (Carattoli and Hasman, 2020).

2.5 Assembly and annotation

The sequence data used in this study are available from the NCBI database (Sayers et al., 2022) and retrieved with the SRA Toolkit (SRA Toolkit Development Team, 2022) using the SRA accession numbers listed in the metadata table.

To select a suitable assembler for our short read metagenomic data, we carefully evaluated the performance of a set of assemblers on our server and on targeted data sets. The pre-selected set of assemblers was chosen based on evaluations in previous studies

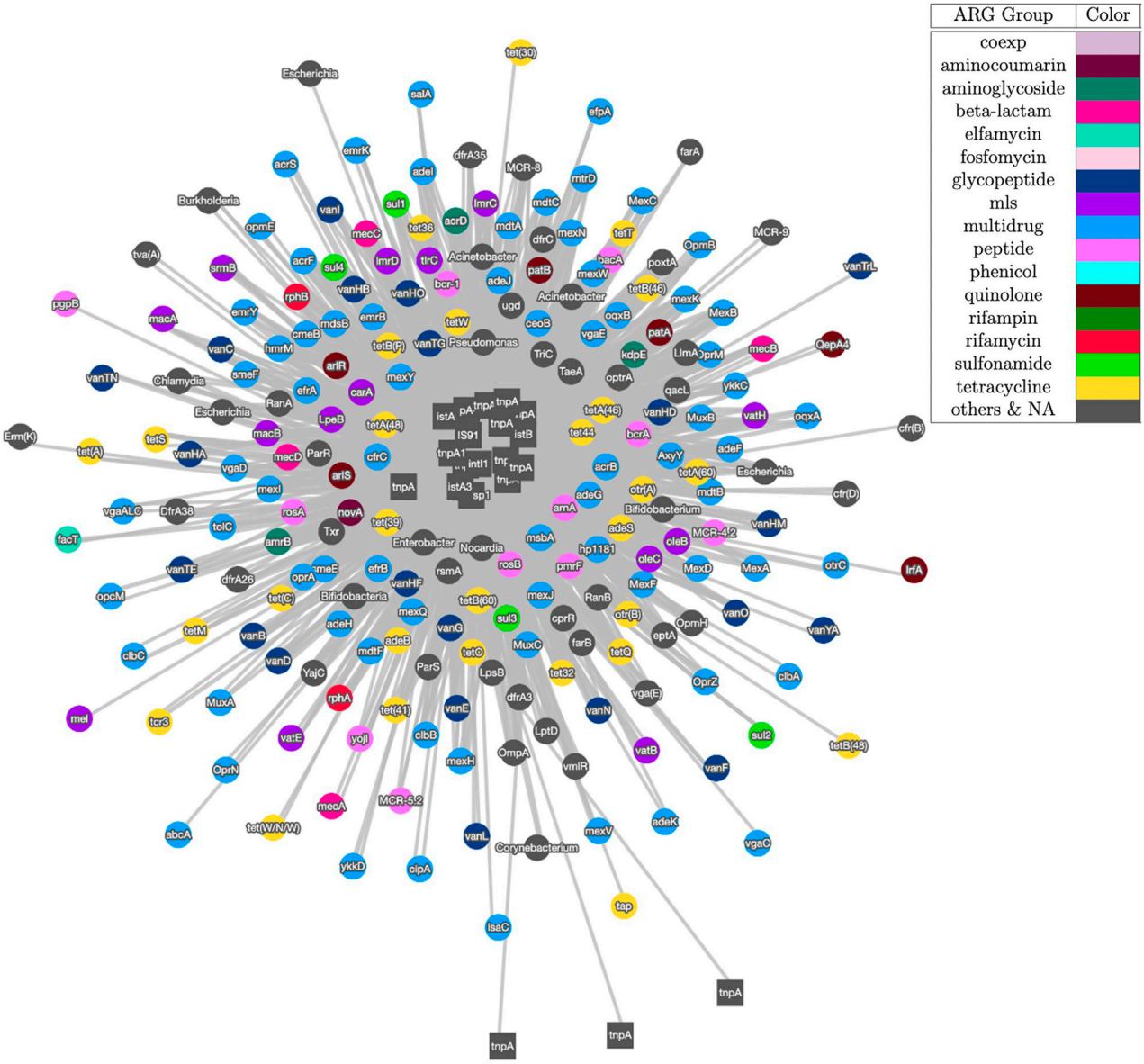


FIGURE 3
Co-occurrence graph generated using Cytoscape with threshold of 3 of samples SRR2088951, SRR2088982, SRR2088983, SRR2089011, SRR5571001, SRR5997542, SRR5997549, SRR9141345, SRR9141349, SRR9141356, SRR9141357, SRR9141362, SRR9141365, SRR9141380, and SRR9141383. MGEs are represented as square node and ARGs are shown as circle nodes. The colors of ARG nodes correspond to classification according to the corresponding class of antibiotic resistance assigned in CARD database (Alcock et al., 2020). The width of the edge between ARGs and MGEs in proportion to the number of common occurrences of each pair.

(Vollmers et al., 2017; Ayling et al., 2020; Zhang et al., 2020). Table 1 and Supplementary Table S1 summarizes the results of different analyses of these samples.

We evaluate the assemblers on the samples as follows: one reclaimed waste water sample (water sample 1), one final treated biosolids sample (water sample 2), and two raw sewage and treated wastewater samples (water sample 3 and 4) for the results depicted in Table 1. Note that the samples used here are different from those presented in Section *Results*. The first two samples were produced by our group, and the latter two samples were published in previous work (Lekunberri et al., 2018). For the first two wastewater samples we tested three assemblers: MetaSPAdes (Nurk et al., 2017), IDBA-UD (Peng et al., 2012) and MegaHIT (Li et al., 2015). While the annotation results of IDBA-UD and MegaHIT were similar, MegaHit showed a better performance in terms of time and memory usage in our test scenario. For the other two wastewater samples, we compared MetaSPAdes and MegaHIT. Overall, we found that on our data sets, MegaHIT generated reasonable results in a relatively short amount of time. Therefore we provide MegaHIT as the default assembler.

DIAMOND (Buchfink et al., 2015; 2021) was incorporated as the primary annotation tool across ARGem, both for short reads matching and contig annotation. DIAMOND is a open-source sensitive protein aligner used widely in the bioinformatics field.

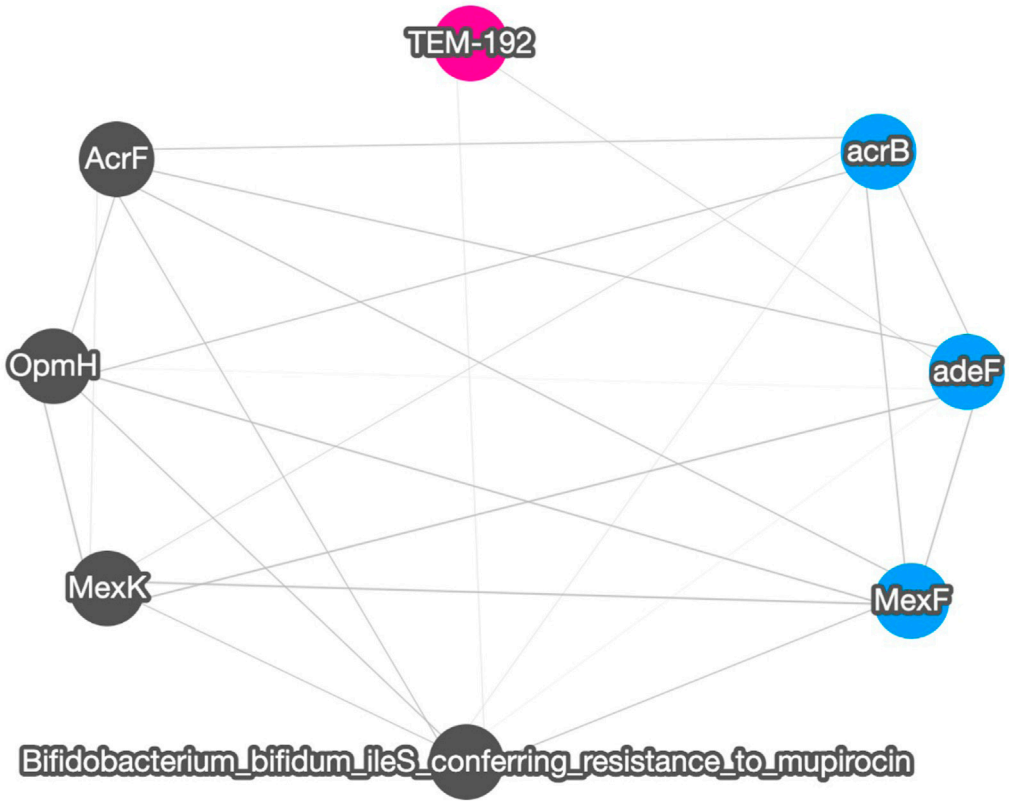


FIGURE 4
A correlation graph for 16S rRNA normalized short read matching result of samples SRR2088951, SRR2088982, SRR2088983, SRR2089011, SRR5571001, SRR5997542, SRR5997549, SRR9141345, SRR9141349, SRR9141356, SRR9141357, SRR9141362, SRR9141365, SRR9141380, and SRR9141383. The color codes are the same as in Figure 3. The width of the edge between ARGs and MGEs is in proportion to the absolute correlation value of each pair.

DIAMOND performs double-index alignment with a reduced alphabet and spaced seeds. DIAMOND has been reported to consume less amount of time for high-throughout scenarios compared to BLASTX (Camacho et al., 2009) and BLASTP in similar settings. We also use BLAST for our optional MGE Parnanen et al. (Parnanen et al., 2018) database for the nucleotide annotation, which is not available in DIAMOND.

2.6 Gene Co-occurrence and correlation analysis

Co-occurrence analysis is a widely applied technique in bioinformatics, and can infer important relationships among genes, such as their taxonomic host, their tendency to be co-expressed, and their ability to be co-mobilized via HGT (Faust and Raes, 2016). Sequencing depth is an important factor that influences the coverage and accuracy of assembly and thus the accuracy of co-occurrence analysis. This, in addition to inherent differences in microbiomes (diversity, representation in databases, etc.) creates difficulties for identifying a single method to accurately calculate gene correlations.

For co-occurrence analysis of ARGs and MGEs, the ARGem pipeline combines an ARG database and an MGE database to count the number of co-occurrence of contigs for each pair of one ARG and one MGE.

For correlation analysis, ARGem first imputes the missing values with zeros for the abundance data and then renormalizes it to be relative abundance data. This method is adapted from (Tao, 2014). We assume that the expression of each pair of genes is generated by an underlying bivariate normal distribution. Considering a gene pair denoted as (x_1, x_2), we calculate the mean values (μ_1, μ_2), the standard deviation (σ_1, σ_2), and the correlation ρ. To accomplish this, we need at least three complete gene pairs. Let N be the total number of experiments, and let $f(\cdot)$ represent the probability density function (pdf) of the underlying bivariate normal distribution. $F(\cdot)$ represents the combination of the pdf and the cumulative distribution function (cdf) of the normal distribution. The likelihood function L is defined as follows:

$$L\left(\hat{\theta} \mid x_1, x_2\right) = \prod_{i=1}^{N} f\left(x_{i1}, x_{i2}\right)^{\delta_{i1}\delta_{i2}} \cdot \frac{\partial}{\partial x_1} F\left(x_{i1}, c_2\right)^{\delta_{i1}(1-\delta_{i2})} \cdot \frac{\partial}{\partial x_2} F\left(c_1, x_{i2}\right)^{(1-\delta_{i1})\delta_{i2}} \cdot F\left(c_1, c_2\right)^{(1-\delta_{i1})(1-\delta_{i2})},$$

where c_1 and c_2 are the detection cut-offs for x_1 and x_2, and δ_{i1} and δ_{i2} are indicator variables indicating whether or not data is available for each x_{ij}.

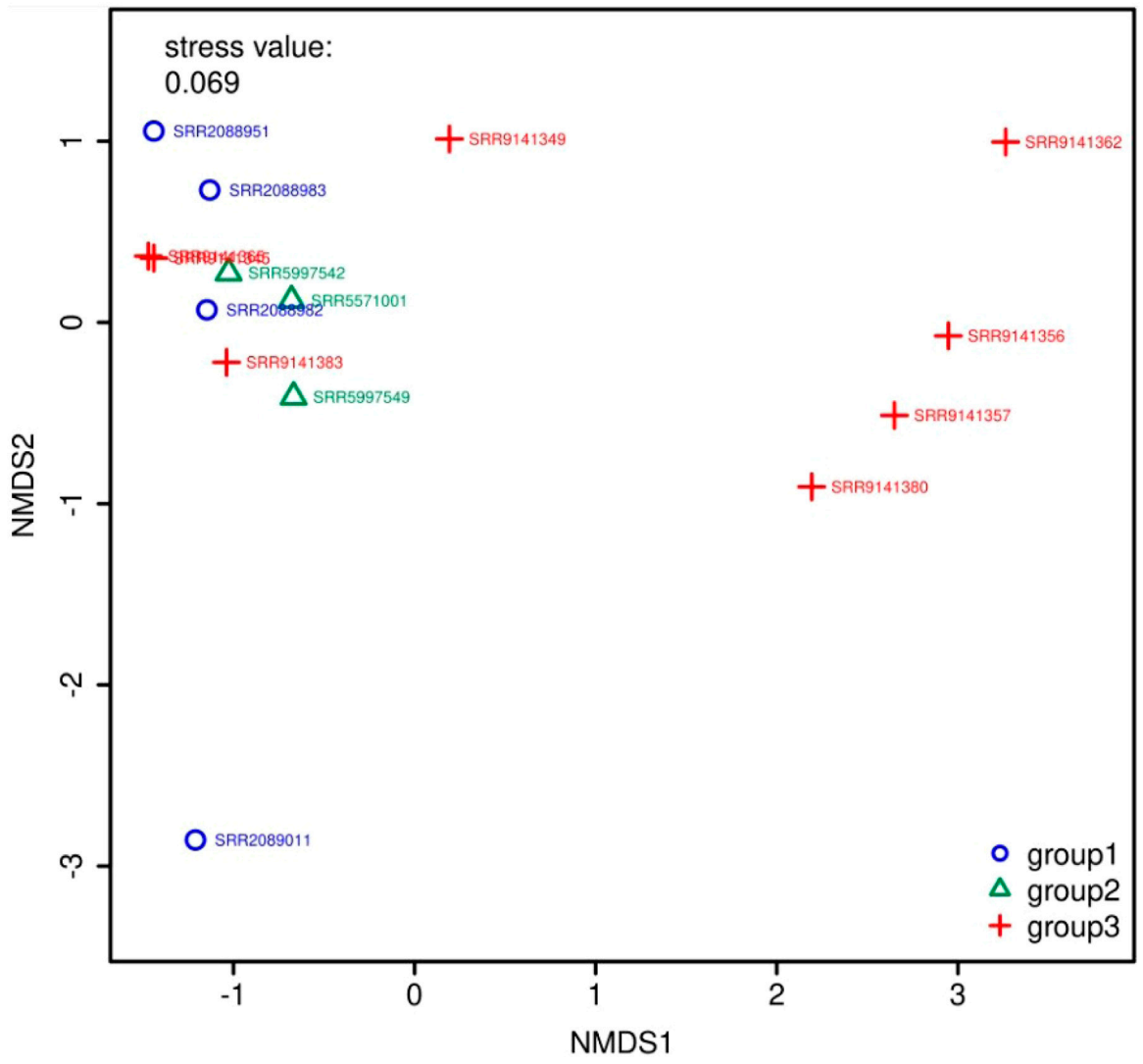


FIGURE 5
NMDS (Kruskal, 1964) plot for the 3 groups of samples. The axes of a NMDS plot are arbitrary units. Different colors and symbols distinguish samples in different groups. The stress value indicates the reliability of the ordination of the NMDS plot, while a stress value close to 0.05 indicates fair fit. In this plot, there are two data points that overlap almost entirely, which means they are similar to each other in the multidimensional space, compared to other data points.

In the above equation, we first calculate the regular likelihood term $f(\cdot)$ when data are available for both pairs and then the second term factorizes into the pdf of x_1 and the cdf of x_2 at the cutoff term in a normal distribution that is shifted up by the distance of the current x_1 observation from its mean multiplied by the correlation coefficient and scaled by the ratio of variances using $F(\cdot)$. If the correlation between the genes is strong, we expect the cdf of x_2 at the cutoff to be directly related to the distance of x_1 from its mean and *vice versa*. Then we calculate the joint cdf of the bivariate normal distribution at both cutoffs. The joint cdf term grows as the values of the cut-offs rise relative to their corresponding means. As this term increases, it tends to overshadow information from other terms.

Our approach involves maximizing the likelihood of observing a given expression pair while adjusting for a known cut-off threshold. In addition, we also capitalized on the data structure by introducing correlation bounds. To obtain sharper correlation estimates, we further utilize the partial correlation definition inequality to update our correlation estimates based on the correlation between other pairs. In this way, the proportional value of relative abundance can directly reflect the degree of correlation of the potential related gene pairs and we are able to produce correlation estimates even with severe missing data issues.

In the next step, our user can apply the desired threshold within the range [−1, 1] on the correlation matrix to filter out the relevant gene pairs for further analysis or visualization.

2.7 Visualization

Network analysis provides an intuitive means to visualize predicted relationships within bioinformatics fields, such as protein-protein interaction networks (Bharadwaj et al., 2017),

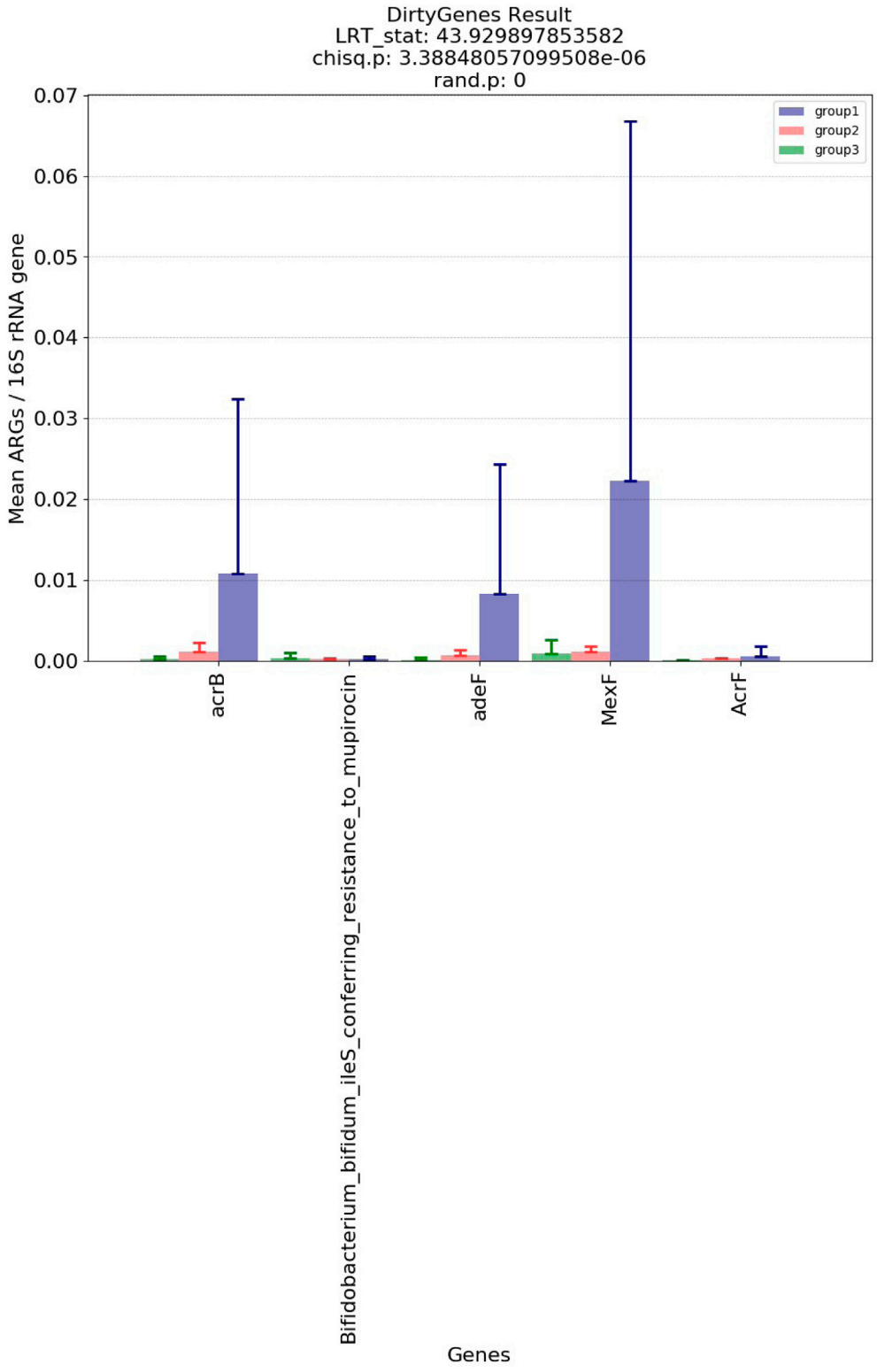


FIGURE 6
Bar plot of DirtyGenes (Shaw et al., 2019) test statistic result, divided into three sub-groups of the fifteen samples. This DirtyGenes statistic was generated based on 16S rRNA normalized ARG annotation result output by the pipeline, with columns only preserved where there were non-zero values for all three groups preserved.

gene-gene networks (Franz et al., 2016), and gene co-expression (Zhang and Horvath, 2005). ARGem approaches visualization from two perspectives: correlations and co-occurrences. Correlation graphs show relations among ARG annotation results without MGE using the method described in *Gene Co-occurrence and Correlation Analysis*. Co-occurrence graphs map ARGs and MGEs based on a number of co-occurrence pairs annotated on the same contigs assembled from raw sequences. For example, in three contigs *C*1, *C*2, and *C*3 in one sample, all contain the ARG-MGE pair ARG *A*1 and MGE *M*1, the occurrence of (*A*1, *M*1) is 3. The width of the edge between *A*1 and *M*1 will reflect the co-occurrence, in this case, which is 3. In correlation graphs, the width of the edges is based on the correlation score between two genes *Gene Co-occurrence and Correlation Analysis* and indicates the relative strength of the relationship in *Assembly and Annotation*. The size of each node is determined based on the sum of abundance in the metagenomic library. Co-occurrence networks, on the other hand, are an analysis of ARGs and MGE annotated on assembled contigs (1,000+ bps). Each edge that connects an ARG node and an MGE node represents the count of the given combination, where the width of the edges indicates the frequency that the combination is encountered (Arango-Argoty et al., 2019). Note that in co-occurrence networks, ARG nodes are only connected to MGE nodes.

ARGem by default builds correlation graphs and co-occurrence graphs using Cytoscape.js (Smoot et al., 2011; Franz et al., 2016), an open-source JavaScript-based graph library (Franz et al., 2016). Cytoscape provides interactive features so that users can select the target genes or filter the abundance rank from the network. Cytoscape library also enables changes in graphic scale, which can be adjusted to end users' preferable size of visualized images. Other tools such as Python NetworkX library (Hagberg et al., 2008) are also included or can be made available for visualization.

2.8 Relational database

We employ the MySQL database for data management and storage. The database schema is shown in Figure 2. Only general information such as the SRA number and email address are required for data retrieval and task status notification. As for optional fields, we provide default data processing and visualization parameters, such as the MGE database and the co-occurrence threshold. Users can customize these parameters to meet their specific needs.

By allowing users to upload customized metadata spreadsheets, our database design can expand to include arbitrary metadata attributes. We record user custom metadata entries in the *metadata_attribute* table, which are available for all projects. With custom metadata, users can compare and visualize data across different projects. For an example metadata spreadsheet, see Supplementary Material.

3 Results

3.1 Pipeline

The ARGem pipeline consists of multiple computational components arranged primarily in a linear sequence, with built-in detection of certain error cases that serve to halt the pipeline early and send out an email notification of the error. We integrated the ARGem pipeline as a key component in the web-based platform AgroSeek (Liang et al., 2021). ARGem can also be deployed in other systems that incorporates a relational database management system, as detailed in Section *Relational Database*. The overall workflow is depicted in Figure 1B. For a more detailed workflow diagram, see the Supplementary Figure S1.

The typical pipeline steps are summarized in the following sub-sections.

3.1.1 Input spreadsheet for a project

An ARGem Excel spreadsheet was designed through collaboration with environmental scientists to identify required *versus* recommended metadata for samples of various categories, along with specified reporting units. As an example, for aquatic environment samples, required metadata columns include the kind of experiment type from which the sample was collected (e.g., lab, field or pilot, selected from a drop-down menu), the DNA extraction method, the DNA sequencing platform, DNA sequencing output (e.g., single or paired reads), and the SRA accession number for each sample. The required columns are provided along with conditional columns depending on the type of aquatic environment matrix selected from the drop down menu.

Through an SRA number column, each sample is associated with a unique SRA number (Sayers et al., 2022) in the input spreadsheet. Therefore the raw data sequences can be conveniently retrieved from the online repository, if they have not yet been added to local data storage data. A complete, filled ARGem spreadsheet provides useful information on both the metadata and the raw data sequence, which can support richer analysis and visualization in later steps of the pipeline. In addition, a relational database associated with the pipeline is provided to store and manage the uploaded data, as well as the status of created projects.

Typically, the user selects one template from the ARGem library of spreadsheet templates that best represents the environment under study, customizes the template for their project, and enters the metadata into the spreadsheet with one row for each metagenomic sample.

There are in total six templates in the library, including five templates for different environment sample types and one user custom template. Through collaboration with environmental scientists, we designed specific templates for water, soil, treated or raw manure, pre- and post-harvest crop production system, and air samples.

3.1.2 Retrieve DNA sequence data from NCBI

In this step, sequence data are retrieved based on the input SRA numbers provided for each sample in the metadata spreadsheet. These data subsequently serve as raw sequence files for the samples required for subsequent analysis.

3.1.3 Assemble each DNA sample into contigs

In this step, the pipeline assembles the retrieved sequence files using the integrated assembly tool, namely, MEGAHIT (Li et al., 2015). This assembler was selected after evaluation on our server and targeted data sets. For details of the assembler evaluation, see Supplementary Table S1. The results of this step are a set of contigs for each sample.

3.1.4 Annotate known ARGs and MGEs in short reads and contigs

This step performs annotation on both the assembled contigs (long-contig annotation) and retrieved short reads (short reads matching) using the integrated annotation tools (BLAST (Altschul et al., 1990) and DIAMOND (Buchfink et al., 2015; 2021)). The reference databases used for this step include an ARG reference database CARD (Alcock et al., 2020) along with three optional MGE reference databases: MobileOG (Brown et al., 2022), NanoARG (Arango-Argoty et al., 2019) (which is the database also used in our MetaCompare (Oh et al., 2018) service), and Parnanen et al. (Parnanen et al., 2018). The annotated genes for each sample are sent to output text files along with their relative abundances.

3.1.5 Analysis

After obtaining the assembly and annotation results of each sample, the pipeline performs a set of analyses based on the results and the metadata attributes. Because it is not possible to discern ARGs imparted by mutations in housekeeping genes from true housekeeping genes, due to limitations in the resolution of sequencing technologies, ARGem excludes housekeeping genes from ARG analysis. A list of excluded genes is provided in the Supplementary Material. The results of the analysis are then made available to the users, usually in the form of tabular files. After this step, more optional analysis requiring user input parameters can be performed according to the desires of the user.

3.1.6 Visualization

For the gene co-occurrence and correlation analysis results, corresponding visualizations are generated and provided to the users. Some of the visualizations can be customized by user-selected parameter inputs.

3.1.7 Notification

After obtaining the results of each sample, or if the pipeline halts early, an email notification is sent to a designated e-mail address reporting the final status (success, partial success, or failure) of the pipeline. When the pipeline does not execute successfully, the notification will include specific information about the detected errors to help guide the user in addressing the problem.

3.2 Verification

The ARGem pipeline was tested using publicly-available data extracted from the NCBI database (Sayers et al., 2022). Results shown in this section are based on 15 fresh water samples obtained from BioProject PRJNA287840, collected monthly from 6 sites in 3 southwestern British Columbia streams over 14 months (Vlok et al., 2019). In the analysis results presented later, these 15 samples were arbitrarily divided into three groups to illustrate the functionality of the tools, rather than to reflect the inherent characteristics of the data. The results presented in this study have been annotated with one of the pipeline's default MGE databases. However, users have the option to choose a different database or integrate their preferred database into the pipeline.

The pipeline generated tables that summarize results for three analyses: 1) short read matching to profile ARGs and estimate their relative abundances, 2) assembly of contigs from short reads, and 3) annotation of ARGs and MGEs in assembled contigs. Short read matching results for these fifteen samples yielded 380 annotated ARGs found in at least one sample out of the fifteen, with 16S rRNA, TPM and FPKM normalization reported in three separate files. Contig assembly generated assembled contigs for all fifteen samples. The ARG and MGE annotation based on assembled contigs generated one table of annotated ARGs and one table for annotated MGEs, for each sample. A table was also generated to report ARGs and MGEs that were found to co-occur in the samples.

Figure 3 shows the visualization result based on contig assembly and annotation. This analysis and visualization is included in the ARGem pipeline. This is a co-occurrence network based on ARG and MGE annotation results on assembled contigs, using reference database CARD (Alcock et al., 2020) and Parnanen et al. (Parnanen et al., 2018), respectively. The co-occurrence graph is generated based on the number of co-occurrences in the sample. Once each combination of the MGE-ARG pair is counted, the pipeline filters the number of occurrences based on user input. Filtered pairs generate a co-occurrence graph, where nodes represent ARGs and MGEs detected and edges represent their occurrence together.

Figure 4 shows the correlation result based on short read matching. Given the 16S rRNA normalized ARG annotation generated by the pipeline, a correlation matrix was generated by the pipeline's correlation analysis module and visualized as a correlation graph. The correlation matrix calculated by our proposed method reports a range from −1 to 1 and excludes single paired combinations, where only two data points or less were found. See also Supplementary Figure S2 for the correlation visualization output using Python NetworkX library instead of the default option Cytoscape.

Figure 5 and Figure 6 show the visualization results based on short read matching. For the visualization on short read matching results, the 15 samples were divided into 3 groups: 1) SRR2088951, SRR2088982, SRR2088983, SRR2089011, 2) SRR5571001, SRR5997542, SRR5997549, and 3) SRR9141345, SRR9141349, SRR9141356, SRR9141357, SRR9141362, SRR9141365, SRR9141380, SRR9141383. Results based on the three relative abundance normalization methods are reported in the annotation table, which can then be processed by external analysis tools. Based on the 16S rRNA normalized ARG annotation generated by the pipeline, an NMDS (Kruskal, 1964) plot was generated for the three groups, as depicted in Figure 5. DirtyGenes (Shaw et al., 2019) was also used to process the 16S rRNA normalized ARG annotation result, where columns are preserved only if there were non-zero values for all 3 groups. The average and standard deviation values of DirtyGenes test statistic for each group depicted in Figure 6.

The visualizations shown here are examples of the analysis that can be performed based on ARGem outputs, but do not have to be limited to the tools and methods described above. Overall, the result tables generated by the ARGem pipeline are capable of further analysis and can be processed by different analysis and visualization tools.

4 Discussion

Antibiotic resistance is a significant public health concern that cannot be ignored (Vikesland et al., 2019). Metagenomics is a

promising approach for comprehensively monitoring ARGs and pathogens in healthcare settings, as demonstrated in recent studies (Berglund et al., 2019; de Abreu et al., 2021; Shen et al., 2021). The development of metagenomic data processing tools that can effectively aid in this detection is a beneficial but also challenging task. One of the challenges is that data from various studies can be collected in different environments and have varying characteristics, making it difficult to collate and organize the data. Additionally, there are multiple versions of the MGE reference database, each containing distinct lists of MGEs. This can be attributed to different research fields having varying perspectives on important MGEs, but also makes it challenging to develop an intergrated tool.

Here we integrated several essential aspects of metagenomic data processing into the ARGem pipeline, including short read matching, contig assembly, and annotation of ARGs and MGEs on assembled contigs. These steps are aligned and automated to provide an all-inclusive pipeline to support global ARG monitoring. The ARGem pipeline allows flexible metadata table inputs, including user-customizable metadata attributes, to be applied to data from different environmental sources and allows possible customized usage by users of this pipeline. A supporting SQL database structure has been developed to manage the flexible input and released along with the pipeline. In the ARG and MGE annotation step, this pipeline provides several different MGE databases for users to choose from. In the short read matching step, the normalization results of three different methods (16S rRNA, TPM, and FPKM) are provided to suit different research purposes. The data generated from this pipeline are capable of being further analyzed and visualized using various tools. Among those, two analysis tools, namely, the correlation analysis and co-occurrence network analysis tools, are included in the release of the pipeline.

Our intention is to offer the community an available, flexible and convenient pipeline designed specifically for metagenomics data to accommodate tincreasing needs in related fields, primarily focusing on the threats of ARGs posed to the agriculture chain and human health. The ARGem pipeline is constructed based on the discussion, suggestion, and testing by actual users who have conducted metagenomics studies and performed agriculture practices in related fields. By implementing flexible metadata input and relational database storage, user customizable reference databases, and an extendable analysis module, the ARGem pipeline intends to introduce flexibility and variety for data input and subsequent analysis, as well as automate the handling of such data. With the release of this pipeline, it is our intention for researchers to have a convenient pipeline to deploy and run on lab scale resources.

5 Conclusion

In this study, we present the ARGem pipeline as a tool for investigating features relevant to antibiotic resistance in environmental metagenomic data sets. As a significant impact on human health, antibiotic resistance has gained increasing attention from researchers and policymakers. As metagnenomics studies being an effective means of comprehensively monitoring ARGs and pathogens in healthy environments, we aim for the ARGem pipeline to contribute to this purpose as an integrated, flexible, and deployable tool.

We describe in this paper the overall workflow and mechanics of each step within the ARGem pipeline, including the methods and tools integrated into the pipeline. We demonstrate its applicability and flexibility through the analysis of metagenomic samples collected from aquatic environments. The ARGem pipeline is developed to be deployable on lab-scale resources, distinguished from other large, general and online pipelines.

Our intention is to make this pipeline readily accessible to a broad range of users, including governmental and academic researchers and policymakers, for tracking key drivers of antibiotic resistance in various environments using metagenomic data. The ARGem pipeline is available in the public domain for free use. In the future, more sequence process and analysis steps can be incorporated into the ARGem pipeline to accommodate the rapid pace of development in this field, which will be facilitated by the adaptable nature of ARGem.

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.

Author contributions

AP, KK, KX, LH, and LiZ initially conceived the concept of this project. PV contributed to later discussion and important changes on the concept. XL, JiZ, and YK carried out the core research effort, including designing and implementing the pipeline and associated database. Additionally, JiZ contributed to the correlation analysis and YK contributed to the co-occurrence network analysis work. JH and KL contributed to several important parts of the implementation of the pipeline, including the initial Luigi workflow builder and SRA retrieval module. IK and BD provided design input to the workflow design, metadata collection and analysis modules. SG contributed to the short reads annotation tool integrated in the pipeline. JL and SH contributed to the pilot testing of the pipeline. All authors contributed to the article and approved the submitted version.

Funding

This study was supported by USDA National Institute of Food and Agriculture competitive Grants 2015-68003-23050 and 2017-68003-26498, the U.S. National Science Foundation (NSF) Partnership in International Research and Education Award (PIRE) 1545756, NSF Research Traineeship (NRT) 2125798 and NSF CI4WARS Award 2004751.

Acknowledgments

We would like to thank the undergraduate students who participated in this project for their hard work and contributions: Shuqi Zhao, Ryan Stankiewicz, Mahira Sheikh, Hisham Juneidi, and Jarod Raedels. Also, Dr. A.~J.~Prussin provided consultation in the development of the metadata templates. Mohammed Salem and Xinyi Song from the Department of Statistics provided help on the correlation estimation.

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/fgene.2023.1219297/full#supplementary-material

References

Alcock, B. P., Raphenya, A. R., Lau, T. T. Y., Tsang, K. K., Bouchard, M., Edalatmand, A., et al. (2020). Card 2020: antibiotic resistome surveillance with the comprehensive antibiotic resistance database. *Nucleic Acids Res.* 48, D517–D525. doi:10.1093/nar/gkz935

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *J. Mol. Biol.* 215, 403–410. doi:10.1016/S0022-2836(05)80360-2

Aminov, R. I. (2011). Horizontal gene exchange in environmental microbiota. *Front. Microbiol.* 2, 158. doi:10.3389/fmicb.2011.00158

Arango-Argoty, G., Dai, D., Pruden, A., Vikesland, P., Heath, L. S., and Zhang, L. (2019). NanoARG: A web service for detecting and contextualizing antimicrobial resistance genes from nanopore-derived metagenomes. *Microbiome* 7, 88–18. doi:10.1186/s40168-019-0703-9

Arango-Argoty, G., Garner, E., Prudent, A., Heath, L. S., Vikesland, P., and Zhang, L. Q. (2018). DeepARG: A deep learning approach for predicting antibiotic resistance genes from metagenomic data. *Microbiome* 6, 23. doi:10.1186/s40168-018-0401-z

Arango-Argoty, G., Singh, G., Heath, L. S., Pruden, A., Xiao, W. D., and Zhang, L. Q. (2016). MetaStorm: A public resource for customizable metagenomics annotation. *PLOS One* 11, e0162442. doi:10.1371/journal.pone.0162442

Ayling, M., Clark, M. D., and Leggett, R. M. (2020). New approaches for metagenome assembly with short reads. *Briefings Bioinforma.* 21, 584–594. doi:10.1093/bib/bbz020

Barlow, M. (2009). What antimicrobial resistance has taught us about horizontal gene transfer. *Methods Mol. Biol.* 532, 397–411. doi:10.1007/978-1-60327-853-9_23

Batut, B., Gravouil, K., Defois, C., Hiltemann, S., Brugère, J.-F., Peyretaillade, E., et al. (2018). ASaiM: A galaxy-based framework to analyze microbiota data. *GigaScience* 7, giy057. doi:10.1093/gigascience/giy057

Berglund, F., Osterlund, T., Boulund, F., Marathe, N. P., Larsson, D. G. J., and Kristiansson, E. (2019). Identification and reconstruction of novel antibiotic resistance genes from metagenomes. *Microbiome* 7, 52. doi:10.1186/s40168-019-0670-1

Bharadwaj, A., Singh, D. P., Ritz, A., Tegge, A. N., Poirel, C. L., Kraikivski, P., et al. (2017). GraphSpace: stimulating interdisciplinary collaborations in network biology. *Bioinformatics* 33, 3134–3136. doi:10.1093/bioinformatics/btx382

Breitwieser, F. P., Lu, J., and Salzberg, S. L. (2019). A review of methods and databases for metagenomic classification and assembly. *Briefings Bioinforma.* 20, 1125–1136. doi:10.1093/bib/bbx120

Brown, C. L., Mullet, J., Hindi, F., Stoll, J. E., Gupta, S., Choi, M., et al. (2022). mobileOG-DB: A manually curated database of protein families mediating the life cycle of bacterial mobile genetic elements. *Appl. Environ. Microbiol.* 88, e0099122. doi:10.1128/aem.00991-22

Buchfink, B., Reuter, K., and Drost, H. G. (2021). Sensitive protein alignments at tree-of-life scale using DIAMOND. *Nat. Methods* 18, 366–368. doi:10.1038/s41592-021-01101-x

Buchfink, B., Xie, C., and Huson, D. H. (2015). Fast and sensitive protein alignment using DIAMOND. *Nat. Methods* 12, 59–60. doi:10.1038/nmeth.3176

Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K., et al. (2009). BLAST+: architecture and applications. *BMC Bioinforma.* 10, 421. doi:10.1186/1471-2105-10-421

Carattoli, A., and Hasman, H. (2020). Plasmidfinder and *in silico* pmlst: identification and typing of plasmid replicons in whole-genome sequencing (WGS). *Horiz. Gene Transf. Methods Protoc.* 2075, 285–294. doi:10.1007/978-1-4939-9877-7_20

Chiu, C. Y., and Miller, S. A. (2019). Clinical metagenomics. *Nat. Rev. Genet.* 20, 341–355. doi:10.1038/s41576-019-0113-7

Clarke, E. L., Taylor, L. J., Zhao, C. Y., Connell, A., Lee, J. J., Fett, B., et al. (2019). Sunbeam: an extensible pipeline for analyzing metagenomic sequencing experiments. *Microbiome* 7, 46. doi:10.1186/s40168-019-0658-x

Davis, B., Calarco, J., Liguori, K., Milligan, E., Brown, C., Gupta, S., et al. (2023). *Recommendations for the use of metagenomics for routine monitoring of antibiotic resistance in wastewater and impacted aquatic environments*. Press.

de Abreu, V. A. C., Perdigao, J., and Almeida, S. (2021). Metagenomic approaches to analyze antimicrobial resistance: an overview. *Front. Genet.* 11, 575592. doi:10.3389/fgene.2020.575592

de Nies, L., Lopes, S., Busi, S. B., Galata, V., Heintz-Buschart, A., Laczny, C. C., et al. (2021). PathoFact: A pipeline for the prediction of virulence factors and antimicrobial resistance genes in metagenomic data. *Microbiome* 9, 49. doi:10.1186/s40168-020-00993-9

Dong, X. L., and Strous, M. (2019). An integrated pipeline for annotation and visualization of metagenomic contigs. *Front. Genet.* 10, 999. doi:10.3389/fgene.2019.00999

Eng, A., Verster, A. J., and Borenstein, E. (2020). MetaLAFFA: A flexible, end-to-end, distributed computing-compatible metagenomic functional annotation pipeline. *BMC Bioinforma.* 21, 471–479. doi:10.1186/s12859-020-03815-9

Faust, K., and Raes, J. (2016). CoNet app: inference of biological association networks using cytoscape. *F1000Res.*, 5, 1519. doi:10.12688/f1000research.9050.1

Federhen, S., Clark, K., Barrett, T., Parkinson, H., Ostell, J., Kodama, Y., et al. (2014). Toward richer metadata for microbial sequences: replacing strain-level ncbi taxonomy taxids with bioproject, biosample and assembly records. *Stand. Genomic Sci.* 9, 1275–1277. doi:10.4056/sigs.4851102

Forsberg, K. J., Patel, S., Gibson, M. K., Lauber, C. L., Knight, R., Fierer, N., et al. (2014). Bacterial phylogeny structures soil resistomes across habitats. *Nature* 509, 612–616. doi:10.1038/nature13377

Franz, M., Lopes, C. T., Huck, G., Dong, Y., Sumer, O., and Bader, G. D. (2016). Cytoscape.Js: A graph theory library for visualisation and analysis. *Bioinformatics* 32, 309–311. doi:10.1093/bioinformatics/btv557

Goncalves, R. S., and Musen, M. A. (2019). The variable quality of metadata about biological samples used in biomedical experiments. *Sci. Data* 6 (1), 1–15. doi:10.1038/sdata.2019.21

Grieb, A., Bowers, R. M., Oggerin, M., Goudeau, D., Lee, J. N., Malmstrom, R. R., et al. (2020). A pipeline for targeted metagenomics of environmental bacteria. *Microbiome* 8, 21. doi:10.1186/s40168-020-0790-7

Hagberg, A., Swart, P., and S Chult, D. (2008). *Tech. Rep., los alamos national lab.(LANL)*. Los Alamos, NM (United States).Exploring network structure, dynamics, and function using NetworkX

Jalili, V., Afgan, E., Gu, Q., Clements, D., Blankenberg, D., Goecks, J., et al. (2020). The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2020 update. *Nucleic acids Res.* 48, W395–W402. doi:10.1093/nar/gkaa434

Koonin, E. V. (2018). Environmental microbiology and metagenomics: the brave new world is here, what's next? *Environ. Microbiol.* 20, 4210–4212. doi:10.1111/1462-2920.14403

Kruskal, J. B. (1964). Nonmetric multidimensional-scaling — a numerical-method. *Psychometrika* 29, 115–129. doi:10.1007/bf02289694

Lekunberri, I., Balcazar, J. L., and Borrego, C. M. (2018). Metagenomic exploration reveals a marked change in the river resistome and mobilome after treated wastewater discharges. *Environ. Pollut.* 234, 538–542. doi:10.1016/j.envpol.2017.12.001

Lerminiaux, N. A., and Cameron, A. D. S. (2019). Horizontal transfer of antibiotic resistance genes in clinical environments. *Can. J. Microbiol.* 65, 34–44. doi:10.1139/cjm-2018-0275

Li, D. H., Liu, C. M., Luo, R. B., Sadakane, K., and Lam, T. W. (2015). Megahit: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. *Bioinformatics* 31, 1674–1676. doi:10.1093/bioinformatics/btv033

Liang, X., Akers, K., Keenum, I., Wind, L., Gupta, S., Chen, C. Q., et al. (2021). AgroSeek: A system for computational analysis of environmental metagenomic data and associated metadata. *BMC Bioinforma.* 22, 117. doi:10.1186/s12859-021-04035-5

Luigi Development Team (2020). *Luigi 3.0.3*. Available at: Https://github.com/spotify/luigi

Maiden, M. C. J. (1998). Horizontal genetic exchange, evolution, and spread of antibiotic resistance in bacteria. *Clin. Infect. Dis.* 27, S12–S20. doi:10.1086/514917

Martinez-Romero, M., O'Connor, M. J., Egyedi, A. L., Willrett, D., Hardi, J., Graybeal, J., et al. (2019). Using association rule mining and ontologies to generate metadata recommendations from multiple biomedical databases. *Database-the J. Biol. Databases Curation* 2019, baz059 pagesdoi:10.1093/database/baz059

Meyer, F., Paarmann, D., D'Souza, M., Olson, R., Glass, E. M., Kubal, M., et al. (2008). The metagenomics rast server — a public resource for the automatic phylogenetic and functional analysis of metagenomes. *BMC Bioinforma.* 9, 386. doi:10.1186/1471-2105-9-386

Nurk, S., Meleshko, D., Korobeynikov, A., and Pevzner, P. A. (2017). metaSPAdes: A new versatile metagenomic assembler. *Genome Res.* 27, 824–834. doi:10.1101/gr.213959.116

Oh, M., Pruden, A., Chen, C. Q., Heath, L. S., Xia, K., and Zhang, L. Q. (2018). MetaCompare: A computational pipeline for prioritizing environmental resistome risk. *FEMS Microbiol. Ecol.* 94, fiy079. doi:10.1093/femsec/fiy079

Parnanen, K., Karkman, A., Hultman, J., Lyra, C., Bengtsson-Palme, J., Larsson, D. G. J., et al. (2018). Maternal gut and breast milk microbiota affect infant gut antibiotic resistome and mobile genetic elements. *Nat. Commun.* 9, 3891. doi:10.1038/s41467-018-06393-w

Peng, Y., Leung, H. C. M., Yiu, S. M., and Chin, F. Y. L. (2012). IDBA-UD: A *de novo* assembler for single-cell and metagenomic sequencing data with highly uneven depth. *Bioinformatics* 28, 1420–1428. doi:10.1093/bioinformatics/bts174

Pilalis, E., Ladoukakis, E., Kolisis, F. N., and Chatziioannou, A. (2012). "A Galaxy workflow for the functional annotation of metagenomic samples," in *Artificial intelligence: Theories and applications: 7th hellenic conference on AI* (Lamia, Greece: Springer), 7, 247–253. SETN 2012, May 28-31, 2012. Proceedings.

Sayers, E. W., Agarwala, R., Bolton, E. E., Brister, J. R., Canese, K., Clark, K., et al. (2019). Database resources of the national center for biotechnology information. *Nucleic Acids Res.* 47, D23–D28. doi:10.1093/nar/gky1069

Sayers, E. W., Bolton, E. E., Brister, J. R., Canese, K., Chan, J., Comeau, D. C., et al. (2022). Database resources of the national center for biotechnology information. *Nucleic Acids Res.* 50 (D1)–D20. doi:10.1093/nar/gkab1112

Shaw, L. M., Blanchard, A., Chen, Q. L., An, X. L., Davies, P., Totemeyer, S., et al. (2019). DirtyGenes: testing for significant changes in gene or bacterial population compositions from a small number of samples. *Sci. Rep.* 9, 2373. doi:10.1038/s41598-019-38873-4

Shen, J. X., McFarland, A. G., Young, V. B., Hayden, M. K., and Hartmann, E. M. (2021). Toward accurate and robust environmental surveillance using metagenomics. *Front. Genet.* 12, 600111. doi:10.3389/fgene.2021.600111

Siegwald, L., Touzet, H., Lemoine, Y., Hot, D., Audebert, C., and Caboche, S. (2017). Assessment of common and emerging bioinformatics pipelines for targeted metagenomics. *PLOS One* 12, e0169563. doi:10.1371/journal.pone.0169563

Smoot, M. E., Ono, K., Ruscheinski, J., Wang, P. L., and Ideker, T. (2011). Cytoscape 2.8: new features for data integration and network visualization. *Bioinformatics* 27, 431–432. doi:10.1093/bioinformatics/btq675

SRA Toolkit Development Team (2022). *The NCBI SRA (sequence read archive) toolkit*. Available at: Http://ncbi.github.io/sra-tools/

Tamames, J., and Puente-Sanchez, F. (2019). SqueezeMeta, a highly portable, fully automatic metagenomic analysis pipeline. *Front. Microbiol.* 9, 3349. doi:10.3389/fmicb.2018.03349

Tao, T. (2014). When is correlation transitive? Available at: https://terrytao.wordpress.com/2014/06/05/when-is-correlation-transitive/ (Accessed 06 December, 2023).

Uritskiy, G. V., DiRuggiero, J., and Taylor, J. (2018). MetaWRAP-A flexible pipeline for genome-resolved metagenomic data analysis. *Microbiome* 6, 158. doi:10.1186/s40168-018-0541-1

Vikesland, P., Garner, E., Gupta, S., Kang, S., Maile-Moskowitz, A., and Zhu, N. (2019). Differential drivers of antimicrobial resistance across the world. *Accounts Chem. Res.* 52, 916–924. doi:10.1021/acs.accounts.8b00643

Vlok, M., Gibbs, A. J., and Suttle, C. A. (2019). Metagenomes of a freshwater charavirus from British Columbia provide a window into ancient lineages of viruses. *Viruses* 11, 299. doi:10.3390/v11030299

Vollmers, J., Wiegand, S., and Kaster, A. K. (2017). Comparing and evaluating metagenome assembly tools from a microbiologist's perspective — not only size matters. *PLOS One* 12, e0169662. doi:10.1371/journal.pone.0169662

Yang, Y., Jiang, X., Chai, B., Ma, L., Li, B., Zhang, A., et al. (2016). ARGs-OAP: online analysis pipeline for antibiotic resistance genes detection from metagenomic data using an integrated structured ARG-database. *Bioinformatics* 32, 2346–2351. doi:10.1093/bioinformatics/btw136

Yilmaz, P., Kottmann, R., Field, D., Knight, R., Cole, J. R., Amaral-Zettler, L., et al. (2011). Minimum information about a marker gene sequence (MIMARKS) and minimum information about any (x) sequence (MIxS) specifications. *Nat. Biotechnol.* 29, 415–420. doi:10.1038/nbt.1823

Zhang, A. N., Li, L.-G., Ma, L., Gillings, M. R., Tiedje, J. M., and Zhang, T. (2018). Conserved phylogenetic distribution and limited antibiotic resistance of class 1 integrons revealed by assessing the bacterial genome and plasmid collection. *Microbiome* 6, 1–14. doi:10.1186/s40168-018-0516-2

Zhang, B., and Horvath, S. (2005). A general framework for weighted gene co-expression network analysis. *Stat. Appl. Genet. Mol. Biol.* 4, Article17. doi:10.2202/1544-6115.1128

Zhang, L., Fang, X. D., Liao, H. R., Zhang, Z. M., Zhou, X., Han, L. J., et al. (2020). A comprehensive investigation of metagenome assembly by linked-read sequencing. *Microbiome* 8, 1–11. doi:10.1186/s40168-020-00929-3

frontiers | Frontiers in Genetics

TYPE Original Research
PUBLISHED 24 November 2023
DOI 10.3389/fgene.2023.1291043

OPEN ACCESS

EDITED BY
Nathan Olson,
National Institute of Standards and Technology (NIST), United States

REVIEWED BY
Yasir Hameed,
Islamia University of Bahawalpur, Pakistan
Shuheng Bai,
The First Affiliated Hospital of Xi'an Jiaotong University, China

*CORRESPONDENCE
Hocine Bendou,
hocine.bendou@uct.ac.za

RECEIVED 08 September 2023
ACCEPTED 13 November 2023
PUBLISHED 24 November 2023

CITATION
Livesey M, Eshibona N and Bendou H (2023), Assessment of the progression of kidney renal clear cell carcinoma using transcriptional profiles revealed new cancer subtypes with variable prognosis. *Front. Genet.* 14:1291043.
doi: 10.3389/fgene.2023.1291043

Assessment of the progression of kidney renal clear cell carcinoma using transcriptional profiles revealed new cancer subtypes with variable prognosis

Michelle Livesey[1], Nasr Eshibona[1] and Hocine Bendou[1,2]*

[1]SAMRC Bioinformatics Unit, South African National Bioinformatics Institute, University of the Western Cape, Cape Town, South Africa, [2]Computational Biology Division, Department of Integrative Biomedical Sciences, Faculty of Health Sciences, University of Cape Town, Cape Town, South Africa

Background: Kidney renal clear cell carcinoma is the most prevalent subtype of renal cell carcinoma encompassing a heterogeneous group of malignancies. Accurate subtype identification and an understanding of the variables influencing prognosis are critical for personalized treatment, but currently limited. To facilitate the sub-classification of KIRC patients and improve prognosis, this study implemented a normalization method to track cancer progression by detecting the accumulation of genetic changes that occur throughout the multi-stage of cancer development.

Objective: To reveal KIRC patients with different progression based on gene expression profiles using a normalization method. The aim is to refine molecular subtyping of KIRC patients associated with survival outcomes.

Methods: RNA-sequenced gene expression of eighty-two KIRC patients were downloaded from UCSC Xena database. Advanced-stage samples were normalized with early-stage to account for differences in the multi-stage cancer progression's heterogeneity. Hierarchical clustering was performed to reveal clusters that progress differently. Two techniques were applied to screen for significant genes within the clusters. First, differentially expressed genes (DEGs) were discovered by Limma, thereafter, an optimal gene subset was selected using Recursive Feature Elimination (RFE). The gene subset was subjected to Random Forest Classifier to evaluate the cluster prediction performance. Genes strongly associated with survival were identified utilizing Cox regression analysis. The model's accuracy was assessed with Kaplan-Meier (K-M). Finally, a Gene ontology and Kyoto Encyclopedia of Genes and Genomes enrichment analyses were performed.

Results: Three clusters were revealed and categorized based on patients' overall survival into short, intermediate, and long. A total of 231 DEGs were discovered of which RFE selected 48 genes. Random Forest Classifier revealed a 100% cluster prediction performance of the genes. Five genes were identified with significant diagnostic capacity. The downregulation of genes *SALL4* and *KRT15* were associated with favorable prognosis, while the upregulation of genes *OSBPL11*, *SPATA18*, and *TAL2* were associated with favorable prognosis.

Conclusion: The normalization method based on tumour progression from early to late stages of cancer development revealed the heterogeneity of KIRC and identified three potential new subtypes with different prognoses. This could be of great importance for the development of new targeted therapies for each subtype.

KEYWORDS

kidney renal clear cell carcinoma, normalization, cancer progression, subtypes, prognosis, gene signature

1 Introduction

Multiple different forms of kidney tumors make up the complex disease known as kidney cancer (Hu et al., 2019). Renal cell carcinoma (RCC) is a heterogeneous group of kidney parenchyma tumors that can be further divided into histologically defined subtypes (Znaor et al., 2015; Casuscelli et al., 2017; Xiong et al., 2022). The different subtypes have undergone multiple revisions in the past two decades, due to advancements in the morphological as well as molecular characterization of renal tumors (Kovacs et al., 1997; Lopez-Beltran et al., 2006; Srigley et al., 2013; Moch et al., 2016; Udager and Mehra, 2016).

The recent discoveries in renal tumor transcriptome profiling studies have had a substantial influence in the field of genomics as a category for "molecularly defined renal carcinomas" has been introduced by the World Health Organization 2022 classification of urinary and male genital tumors (5th edition) (Trpkov et al., 2021a; 2021b; Mohanty et al., 2023). These studies have significantly improved our understanding of RCC, however, effective diagnostic and therapeutic approaches have yet to be achieved (Caliskan et al., 2020). Additionally, these studies revealed the high molecular heterogeneity of these tumors, necessitating further sub-classification.

In this study, the most prevalent and aggressive subtype Kidney renal clear cell carcinoma (KIRC) was investigated as it accounts for 80%–90% of the total number of RCC patients (Wang Q. et al., 2019). Patients with KIRC are associated with a high mortality rate and poor clinical outcomes (Gray and Harris, 2019; Puzanov, 2022). Also, there are limited therapeutic options available; surgery is the primary option since KIRC is resistant to radiotherapy and chemotherapy (Yin et al., 2019). The resistance to treatment may be due to the heterogeneity of these tumors. Therefore, an accurate assessment of the heterogeneity of these tumors is crucial to identify subtypes of patients that can benefit from targeted therapy. This can be achieved by investigating the underlying molecular mechanisms and progression of KIRC, which are currently not fully understood (You et al., 2021).

To track cancer progression we implemented a recently established normalization method, which also has the potential to facilitate the sub-classification of KIRC (Livesey et al., 2023). The normalized gene expression reveals how cancer progresses by detecting the accumulated genetic changes that emerge from early-stages of cancer development to advanced-stages. The application of the normalization method and hierarchical clustering will allow for the identification of clusters (subtypes) that progress differently.

This study aims to reveal KIRC patients with different progression (subtypes) and establish a genotype-phenotype link to the identified clusters. In this study, the genotype-phenotype relationship to the distinct clusters was defined by the average overall survival (OS) of the KIRC patient samples. Prognostic gene signatures were identified that differentiate between the different survival clusters and have the potential to function as prognostic biomarkers that can facilitate the prognosis and monitoring of KIRC. Therefore, the study advances knowledge of the transcriptional landscape of KIRC patients with an emphasis on cancer progression.

2 Materials and methods

2.1 Data acquisition and processing

The RNA-Sequencing (RNA-Seq) gene expression profiles of KIRC were downloaded from the UCSC Xena database using cancer-specific data from The Cancer Genome Atlas cohort, from the Genomic Data Commons (GDC-TCGA) (Goldman et al., 2020). A total of eighty-two advanced-stage cancer samples, along with a matched number of randomly selected early-stage samples were extracted. The accompanying metadata included the corresponding patient phenotypic and survival profiles.

The gene expression profile of each patient was organized in a gene-by-sample genomic matrix. The cancer datasets consisted of 60,483 unique Ensembl identifiers (ENSG) (Aken et al., 2016), quantified as $\log_2(x+1)$, where x represents the count of reads mapped to a specific genomic location in the human reference genome (GRCh38.p2, gencode release 22). Ensembl BioMart (GRCh38.p13, Ensembl 104 May 2021) (Smedley et al., 2015) was utilized to retrieve a total of 19,556 ENSG identifiers that were annotated with a protein-coding biotype. Hence, 40,927 (67, 7%) non-coding entries were eliminated. For further analysis, the 19,556 protein-coding gene expressions were converted to counts.

2.2 Data normalization

The normalization method that tracked cancer progression and corrected for multiple cancers (Livesey et al., 2023) was modified to investigate a cancer type. The normalization method involves calculating the quotient of advanced-stage gene expression and early-stage gene expression.

2.2.1 Tracking cancer progression

A normalization method was implemented to capture the heterogeneity between cancerous tumors by detecting their

molecular differences in progression from early to late-stages of tumor development using gene expression by RNA-Seq. As a result, the method exposes the accumulated genetic changes that occur throughout the multi-stage of cancer development. To track the development of cancer, the gene expression profiles of both early-stage and late-stage cancer samples were required. Thus, the gene-by-sample matrix of KIRC was used to create two distinct matrices; early-stage (E) and advanced-stage (A) gene expression as follows:

E, s x r matrix for early-stage gene expression and,

A, s x q matrix for advanced-stage gene expression.

The early-stage and advanced-stage gene expression matrices are represented by E and A, respectively. Where r and q corresponds to the number of cancer samples in early-stage and advanced-stage, and s the number of protein-coding genes represented with raw count gene expression value.

The early-stage patient profiles do not match the same patient profiles in the late-stages. Thus, the initial approach to calculating the normalized dataset involves generating a mean normalized expression, or "m_i", for gene *i* in the early-stage dataset. The sum of early-stage gene *i* for all early-stage cancer *k* samples was calculated, as shown in Eq 1. The average early-stage expression vector of gene *i* produced by this equation offers a more accurate representation of the early-stage expression of a particular gene.

$$m_i = \frac{1}{r}\sum_{k=1}^{r} E_{i,k} \quad \text{(eq 1)}$$

$$\mathrm{L}_i = ln\left(\frac{A}{m_i}\right) \quad \text{(eq 2)}$$

Finally, the gene expression matrix that represents cancer progression, L was calculated as demonstrated in Eq 2. Matrix L contains normalized counts of the quotients of advanced-stage (dividend) and the mean gene expression of early-stage cancer samples (divisor). Therefore, the normalized gene expression represents the continuously changing cellular transcriptome, allowing for an efficient and comprehensive description of gene expression profiles.

2.3 Hierarchical clustering

The clustering of cancer samples is the most fundamental strategy to identify groups of samples that progressed differently in gene expression patterns. This approach may result in the identification of novel cancer clusters (subtypes) within a cancer type. Therefore, the normalized gene expression profiles of the KIRC cancer samples were subjected to hierarchical clustering analysis, to reveal the grouping of cancer samples.

The clusters of cancer samples were created by hierarchical clustering, using the cosine distance between the gene expression profiles and Ward's method for agglomeration (Ward, 1963; Jaskowiak et al., 2014). The optimal number of clusters was determined using the *find_k* function as part of the dendextend R package (version 1.17.1), which calculates k using maximal average silhouette widths (Rousseeuw, 1987). Finally, the dendrograms were split into k groups to assign samples to a cluster.

2.4 Feature analysis

2.4.1 Differential gene expression

Limma package in R (version 3.54.2) (Ritchie et al., 2015) was used to screen for differentially expressed genes (DEGs), by applying an empirical Bayesian approach to evaluate for differences in gene expression profiles between the identified clusters. The *decideTests* (Law et al., 2016) function assigned binary values (0: not detected, 1: upregulated, and −1: downregulated) to the genes, to identify and extract genes that differentiate between the altered (up or down) gene expression. Significant DEGs were defined as those with a Benjamini–Hochberg (BH) adjusted *p*-value <0.05 and log2-fold change (LFC) ≥ 0.5 or ≤ −0.5.

2.4.2 Marker gene selection using machine learning

Recursive Feature Elimination (RFE) algorithm was implemented to identify key genes playing a role in the classification of the identified KIRC clusters (subtypes), using the Scikit-learn python package (Pedregosa et al., 2011). RFE with a linear kernel support vector machine (SVM) was utilized to find optimal genes that predict the cancer clusters. The k-fold cross-validation procedure, with a value of K set to 10, was repeated 3 times.

The model was built with all identified DEGs and in several iterations eliminates a single gene deemed least important for segregating the identified clusters (Guyon et al., 2002). The model is rebuilt, and the new gene subset are evaluated based on their classification performance. Hence, the genes are ranked according to their relevance. In this study, the final gene subset was selected based on the highest classification accuracy by linear SVM with *C* set to 5. The final gene subset was further subjected to principal component analysis (PCA) using the R packages FactoMineR (version 2.8) (Lê et al., 2008) and factoextra (version 1.0.7) (Kassambara and Mundt, 2020).

2.5 Predictive and validation of marker genes

The performance of the RFE selected gene subset was validated using Random Forest (RF) classifier with a "test-train split ()" class to split the data into train and test sets with a ratio of 75: 25. The performance of the RF classifier was measured using accuracy, precision, and recall score as the performance metrics. All machine learning implementations were run in Anaconda environment based on python programming language and Scikit-learn package (Pedregosa et al., 2011).

2.6 Survival analysis

The gene subset selected by RFE was subjected to a Cox regression model based on the Lasso algorithm of the glmnet R package (version 4.1-7), to further understand the relative importance of the gene subset (Friedman et al., 2010; Simon et al., 2011; Tibshirani et al., 2012). The model reduces the total number of the gene subset and identifies the genes with the most significant impact on a patient's survival. This step assigned a regression coefficient value to the given gene that is multiplied by the corresponding gene's expression and results in a prognostic risk

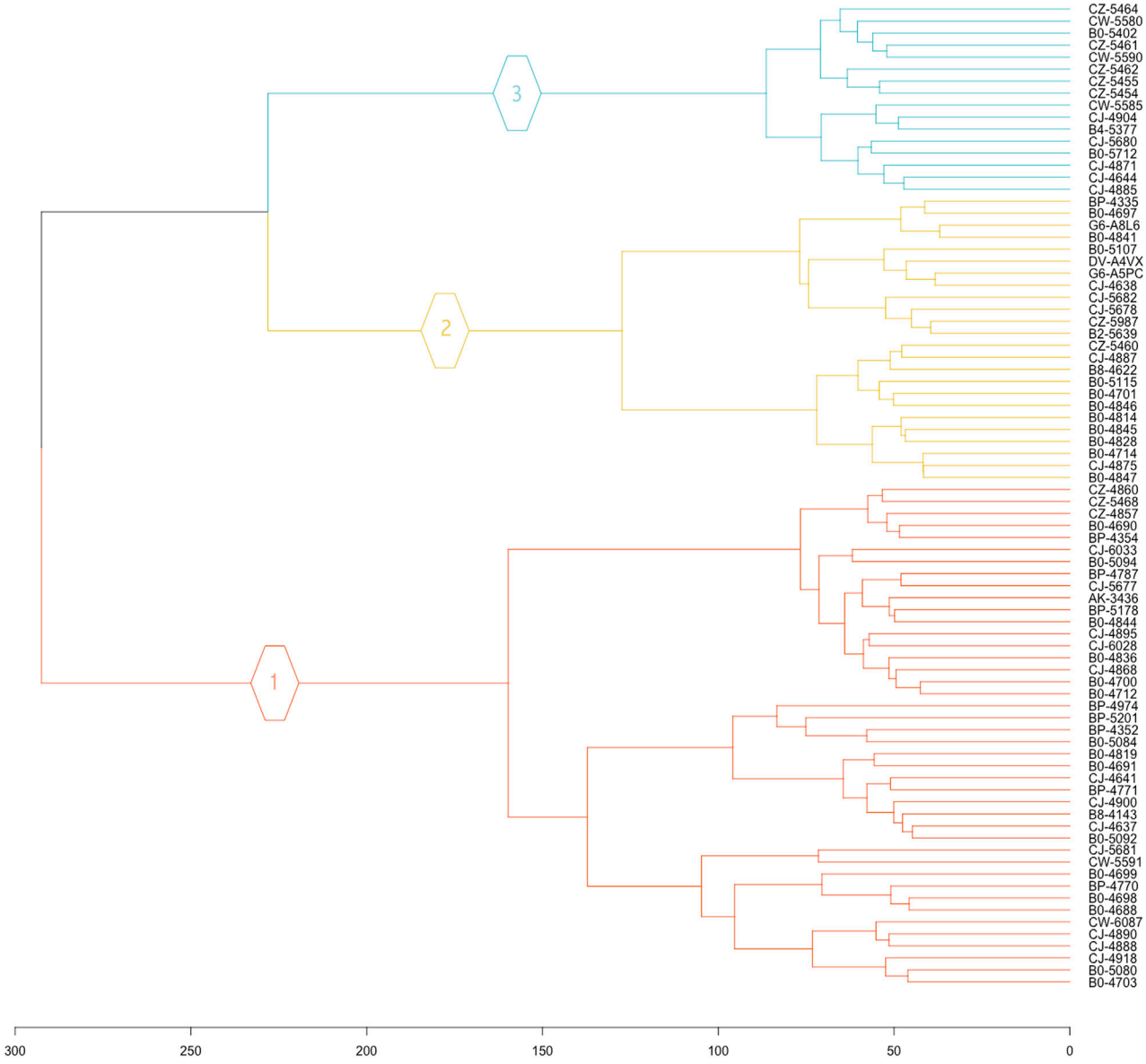


FIGURE 1
Hierarchical clustering dendrogram of KIRC patient. The 19,556 normalized gene expression profiles of the eighty-two KIRC cancer samples were subjected to clustering analysis, to reveal the grouping of cancer samples.

score for each patient. The patient scores were used to calculate a median risk score. Each patient was assigned a status value of 0 or 1 based on whether the patient's score was higher or lower than the median risk score. The patient status information was used to generate Kaplan-Meier (K-M) estimates for OS. The K–M curves were constructed using the *ggsurvplot* function from the survminer R package (version 0.4.9).

2.7 One-way ANOVA

A one-way analysis of variance (ANOVA) was performed to compare the mean gene expression of the prognostic genes discovered by Cox regression analysis between the identified clusters. Statistical analysis was conducted with the stats R package (version 4.2.2). Following the application of ANOVA, Tukey's *post hoc* test for pairwise comparisons was applied (Tukey, 1949). The null hypothesis (H0) of equal mean between the clusters was rejected if the p-value < 0.05; H1: the cluster means are significantly different from one another.

2.8 Enrichment

The list of DEGs were subjected to functional annotations of Gene ontology (GO) (Ashburner et al., 2000), with an adjusted p-value < 0.05 determined as a cut-off criterion for significant enrichment. Additionally, the 48 RFE gene subset were subjected to Kyoto Encyclopedia of Genes and Genomes (KEGG) pathways enrichment,

TABLE 1 The number of patient samples stratified by hierarchical clustering. The average overall survival of all patients within a cluster was calculated and further categorized into Short (SS), Intermediate (IS), and Long Survival (LS).

Cluster	Average survival (days)	Survival time	Risk subcategory	Samples
1	864.43	Short	SS	42
2	1076.38	Intermediate	IS	24
3	1522.31	Long	LS	16
Total				82

with the threshold for significant enrichment established as p-value <0.05. The enrichment analysis was performed utilizing the clusterProfiler R package (version 4.6.2) (Yu et al., 2012).

3 Results

3.1 Cancer clusters detection with normalized expression

The gene expression profiles of eighty-two advanced-stage KIRC samples were normalized with early-stage cancer samples to consider the heterogeneity differences that occur in the multi-stage cancer progression.

In this study, all 19,556 normalized protein-coding genes were subjected to clustering. The clusters are visually represented in a hierarchical tree called a dendrogram. The clustering of all eighty-two KIRC samples revealed three unique KIRC progression patterns based on gene expression profiles (Figure 1).

Three unique cancer clusters (subtypes) as Clusters 1, 2, and 3 were identified and encompass a total of 42, 24, and 16 KIRC patient samples, respectively. These three molecularly identified clusters were further correlated with the patients' average overall survival to reflect its genotype-phenotype relationship. Cluster 1 showed the lowest average OS of 864.43 days, Cluster 2 displayed an average OS of 1076.38, and Cluster 3 had the highest average OS of 1522.31 days. Therefore, these Clusters were categorized as Short (SS), Intermediate (IS), and Long Survival (LS) (Table 1).

3.2 Differential gene expression analysis

In the differential gene expression (DGE) analysis, a total of 19,556 protein-coding genes were evaluated for DEGs to distinguish between SS, IS, and LS. A pairwise comparison approach between the gene expression profiles of IS and SS, LS and SS, and LS and IS were used, and only the genes with an adjusted p-value <0.05 and LFC ≥0.5 or ≤ −0.5 between all three pairwise comparisons were used for further analysis. Thus, a total of 231 DEGs were discovered.

Considering only the DEGs that were significant between all three pairwise comparisons, a total of 47 genes were identified as upregulated, when IS was compared to SS, whereas 184 genes were found to be downregulated. While 159 genes were upregulated, and 72 genes were downregulated in the comparison of LS and SS. Finally, the comparison of LS and IS, identified 221 and 10 genes as upregulated and downregulated, respectively.

3.3 Selection of optimal gene subset

All 231 DEGs identified between SS, IS, and LS KIRC patients were screened by the RFE algorithm. The optimal gene subset is defined by the best combination of genes that has candidate characteristics of classification and prognosis. This also refers to the performance of the RFE and is quantified by the feature importance score. In this study, the optimal gene subset of 48 genes (Supplementary Table S1) with the highest performance score of 0.963 was selected for further analysis (Figure 2A).

3.3.1 Validation of optimal RFE gene subset

An RF classifier model was constructed to evaluate the classification power of the 48 RFE gene subset for SS, IS, and LS. A tenfold cross-validation on a forest model in the training phase (75% of the samples) and testing phase (25% of the samples) was computed. The Random Forest classification yielded an accuracy score of 100%, a precision of 100%, and a recall of 100%.

A confusion matrix that defines the performance of the classification algorithm is presented in Figure 2B. The importance of each gene for risk subcategory prediction to the RF classifier model is presented in Figure 2C.

A PCA model was built to determine the heterogeneity in gene expression between the SS, IS, and LS risk subcategories. The PCA assessed and identified the key sources of variance, allowing samples to be grouped based on similar and different gene expression profiles.

Dim 1 represented 29.8% of the overall variance, whereas Dim 2 represented 23.6% (Figure 3). A clear segregation between KIRC patient samples can be observed to distinguish between the three risk subcategories.

To further compare the initial clustering analysis of protein-coding genes to the clustering of the selected 48 RFE gene subset, a hierarchical clustering was performed with the normalized gene expression of the 48 RFE gene subset of the eighty-two KIRC cancer samples. The correspondence between the two hierarchical clusters is represented by a tanglegram (Figure 4). It can be observed that only four samples were assigned to a different cluster (risk subcategory) with the reduced gene subset (Figure 4).

3.4 Identification of prognostic genes

Five prognostic genes were identified and linked with KIRC patient survival by univariate Cox regression analysis between the 48 RFE gene subset and patient survival data. The prognostic genes were detected utilizing the LASSO algorithm, which assigns non-zero, positive, or negative coefficients. Two of the five genes had

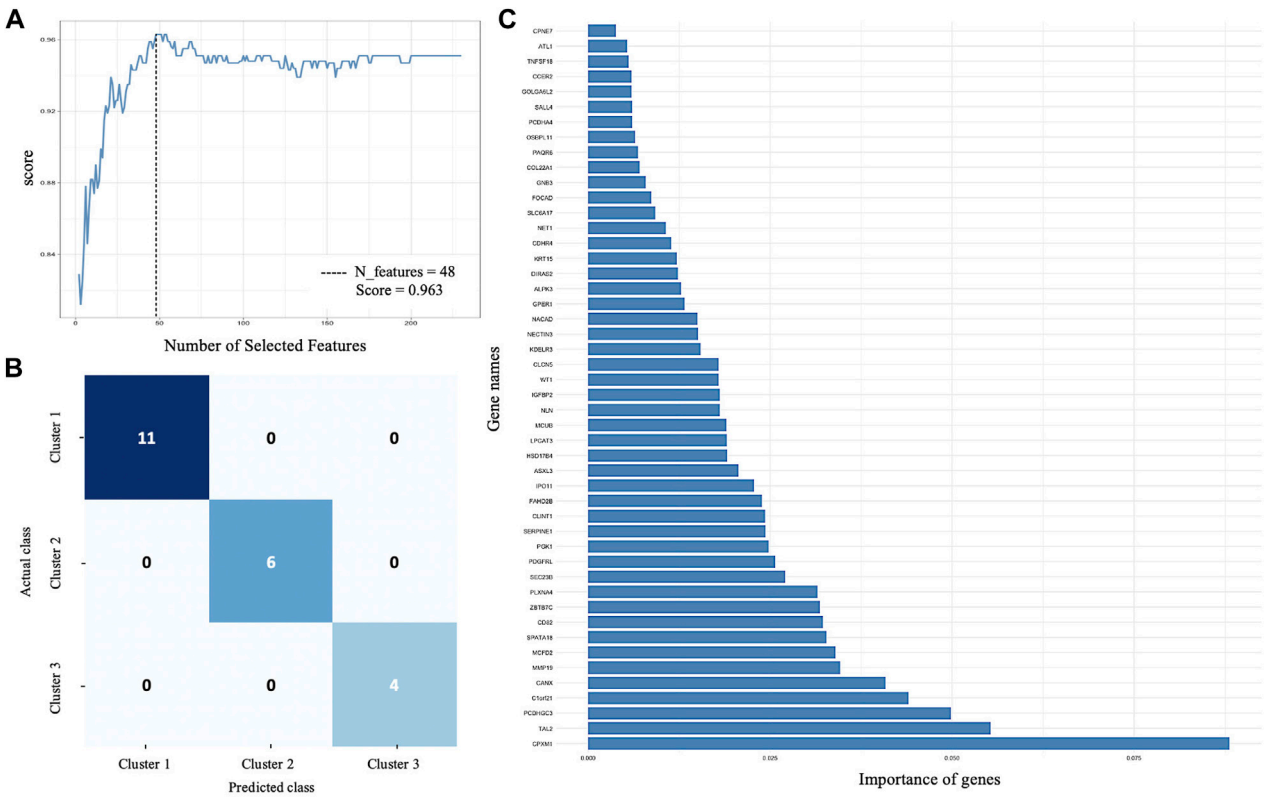


FIGURE 2
Supervised machine learning. **(A)** Recursive feature elimination selected 48 genes with the highest performance score of 0.963. **(B)** Confusion matrix that defines the performance of RF classifier. Each row and columns represent the instances in an actual and predicted class, respectively. **(C)** The importance of each gene for RF classifier prediction.

positive coefficients, while three genes had negative coefficients (Table 2).

Based on patient statuses, the K-M estimations for overall survival were derived and presented below. The K-M curves illustrate low, intermediated, and high gene expression in blue, green, and red colors, respectively. The K-M curves of genes *SALL4* and *KRT15* with positive coefficient values are presented in Figure 5.

The K-M curves for the three genes *OSBPL11, SPATA18*, and *TAL2* with negative coefficient values are presented in Figure 6.

The five prognostic genes' estimations and *p*-values in the Cox regression model were all significant, which demonstrates that the altered expression of these genes affects KIRC survival.

3.5 Gene expression patterns between risk subcategories

One-way ANOVA was performed to assess for differences in the mean normalized gene expression profiles of each of the prognostic genes detected between the risk subcategories. This evaluation included the differences between SS and IS, IS and LS, and SS and LS. Each survival group consisted of a set of samples that make up that risk subcategory, from which a boxplot was created using the normalized gene expression profile of a specific prognostic gene (Figure 7).

All prognostic genes showed a statistically significant difference between SS and LS (p-value ≤ 0.015). It is further noteworthy that ANOVA resulted in a statistical difference in the normalized gene expression between IS and LS (p-value ≤ 0.0032) as well as between survival IS and SS (p-value ≤ 0.018) (Figure 7).

3.6 Enrichment analysis

The GO enrichment analysis illustrated that KIRC DEGs were significantly enriched in biological processes (BP), including extracellular matrix (ECM) organization, extracellular structure organization, and external encapsulating structure organization (Figure 8). In terms of cellular component (CC), collagen-containing ECM, cell leading edge, and cell projection membrane, among other terms were significantly enriched in KIRC DEGs (Figure 8). Lastly, the molecular function (MF), were significantly enriched in ECM structural constituent, growth factor binding, and hormone binding (Figure 8). The KEGG analysis revealed that the 48 gene subset significantly enriched for the p53 signaling pathway, HIF-1 signaling pathway, and estrogen signaling pathway (Figure 9).

4 Discussion

The high molecular heterogeneity of RCC necessitates further sub-classification to establish a successful treatment strategy and

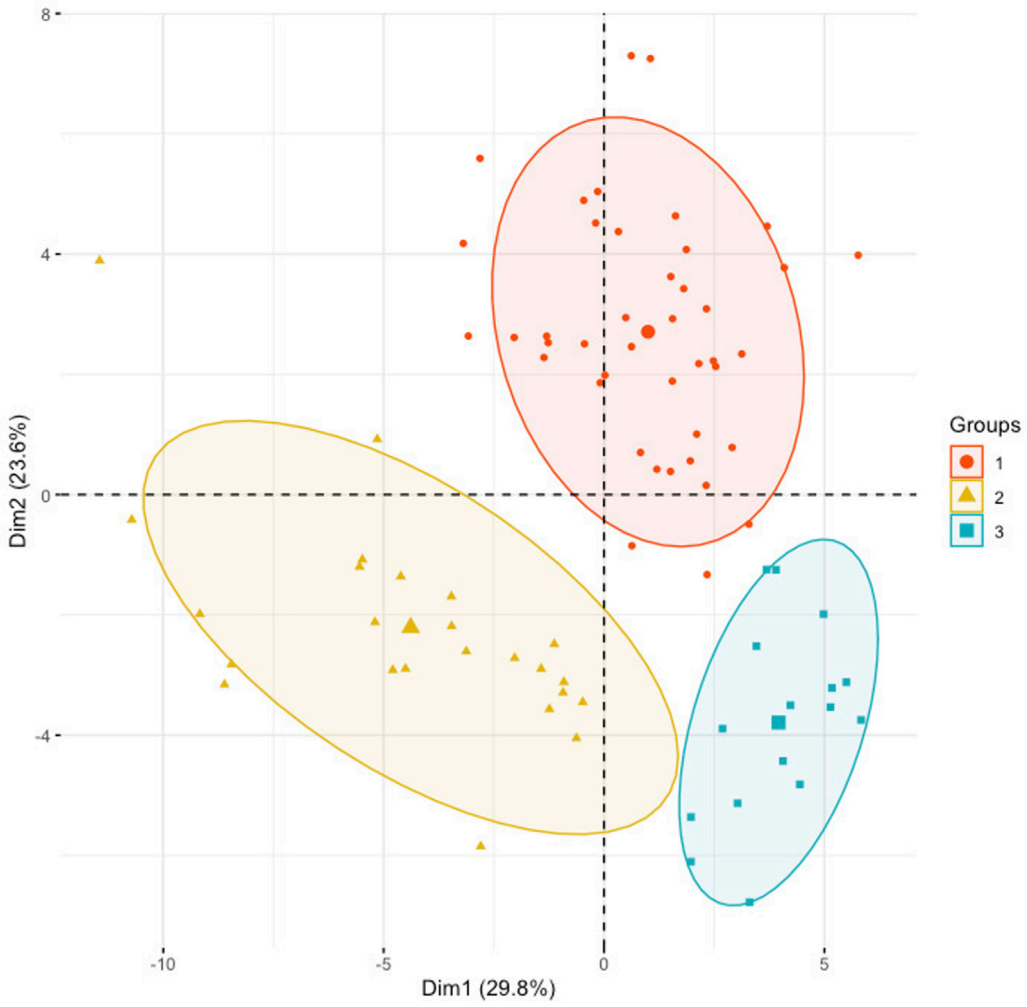


FIGURE 3
Principal component analysis using the normalized gene expression profiles of the 48 RFE gene subset. KIRC samples were stratified according to the initial hierarchical clustering analysis.

medical care. Therefore, this study focussed on KIRC as it represents the majority of RCC diagnoses. The study aims to identify subtypes that reflect a genotype-phenotype relationship for KIRC patients that provide a more accurate prognosis, with an emphasis on cancer progression.

This study implemented a normalization method in which the gene expression profiles of eighty-two advanced-stage KIRC samples were normalized with early-stage cancer samples to consider heterogeneity differences in the multi-stage cancer progression. The normalization method corrects for genes that present with high expression variability in early-stage samples but less expression variability in advanced-stage cancer samples. This leads to the availability of more meaningful information to track the cancer progression from early-to advanced-stage, based on the differences in the gene expression profiles.

The normalized gene expression was subjected to a hierarchical clustering method, to detect cancer samples that progress differently in gene expression patterns. The approach allows for the grouping, alternatively, clustering of cancer samples to identify samples within a group/cluster that are similar to each other and different from samples in other groups. This popular method revealed three cancer clusters (subtypes) for KIRC cancer. The three molecularly defined clusters were correlated with the patients' average OS. It can be noted that patients in Cluster 3 lived on average 657.88 days longer than patients in Cluster 1. Meanwhile patients in Cluster 2 and Cluster 3 live on average 211.95 days and 445.93 days longer than patients in Cluster 1 and Cluster 2, respectively. Thus, the obtained three clusters by the use of our normalization method illustrate different KIRC tumors that progressed differently from early-stage to late-stage cancer development (Figure 3). Consequently, these clusters have different prognoses and can be considered as different subtypes. The results of the hierarchical clustering analysis were subjected to a validation step using an independent GEO dataset (Supplementary Material S1). This test dataset includes sixty-five KIRC samples, and the normalization method also identified three clusters in the GEO KIRC dataset (Supplementary Material S1).

The 48 genes identified through the Machine Learning analysis have the capacity to accurately classify and predict the KIRC subtypes to an extent similar to the use of the 19,556 protein-coding genes. This demonstrates the existence of genetic heterogeneity within KIRC

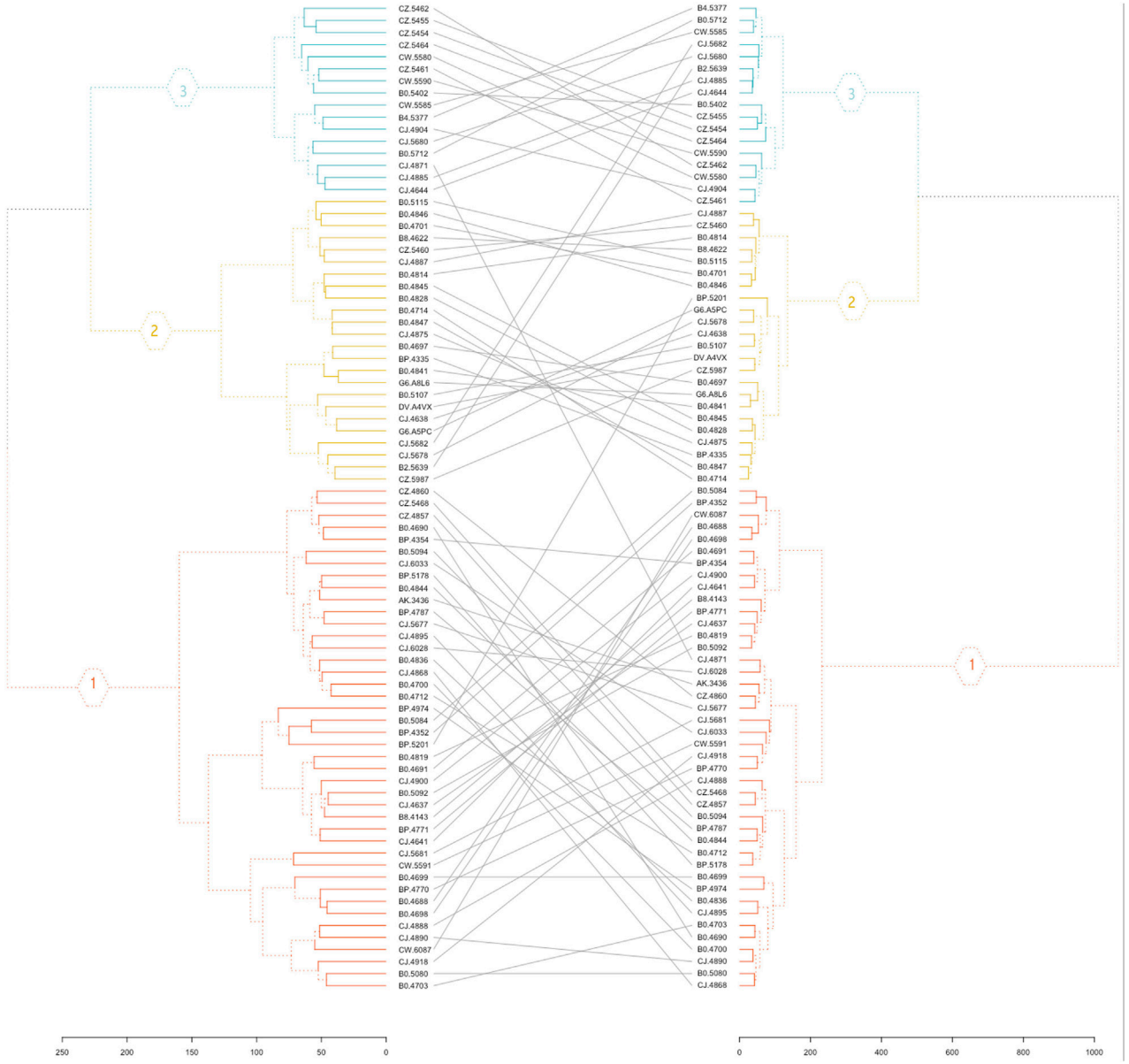


FIGURE 4
Tanglegram. The initial hierarchical clustering of 19,556 protein-coding genes (left) and clustering analysis of the 48 RFE gene subset (right).

TABLE 2 Five prognostic genes. The coefficient value obtained by LASSO algorithm.

Gene name	Coefficient value
SALL4	0.06613418699953
KRT15	0.0296694189909953
OSBPL11	−0.121246995833747
SPATA18	−0.0770127595245775
TAL2	−0.18919349247905

tumors and the ability of our normalization method to recognize this heterogeneity and associate it with prognosis and OS. The gene set contains genes that were reported to play a critical role in the aggressiveness of renal tumors, and our study revealed their involvement in the heterogeneity of the most prevalent and aggressive subtype in renal cancer, KIRC.

Analysis of GO enrichment illustrates the involvement of DEGs in the biological processes that promote tumor aggressiveness. It has been reported that ECM regulates fundamental properties of tumors, such as growth and invasion. The most prevalent genetic mutations in KIRC inactivate the *VHL* gene, which plays a direct role in ECM organization. Therefore, therapeutic approaches to control ECM are currently being investigated and an advanced understanding of KIRC ECM will determine if ECM-modifying drugs are appropriate for KIRC (Oxburgh, 2022). An additional BP enrichment was macrophages that are highly enriched in RCC, and the RCC survival rate is strongly correlated with the inflammatory cytokines secreted by macrophages (Xie et al., 2022).

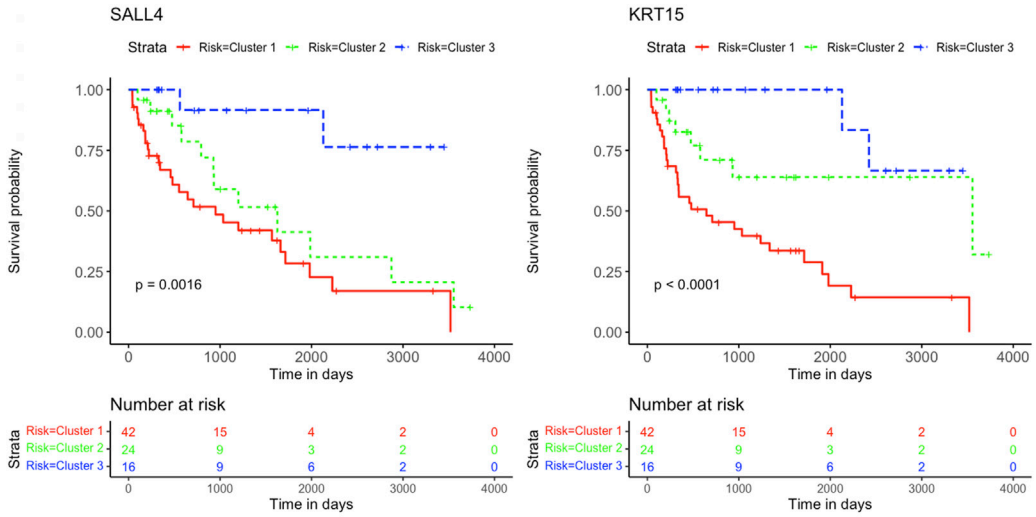


FIGURE 5
Kaplan-Meier survival curves. Analysis revealed the survival prediction associated with high and low gene expression profiles of *SALL4* and *KRT15* prognostic genes in KIRC patients.

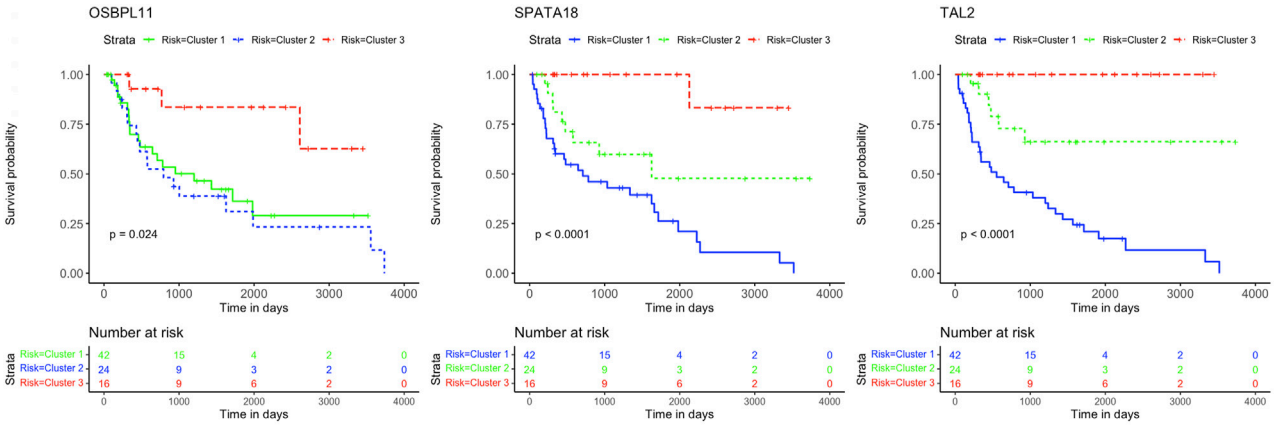


FIGURE 6
Kaplan-Meier survival curves. Analysis revealed the survival prediction associated with high and low gene expression profiles of *OSBPL11, SPATA18*, and *TAL2* prognostic genes in KIRC patients.

In terms of the cellular component (CC), KIRC DEGs were significantly enriched in functional elements such as basement membrane (BM). According to a recent study, KIRC is associated with unique basement membrane gene expression patterns, and the characterization of the BM has the potential to guide clinical therapy (Xiong et al., 2022). Cellular component, collagen trimer has been similarly found in studies focused on renal cancer progression (Wang A. et al., 2019), along with molecular function enriched extracellular matrix structural constituent and platelet-derived growth factor binding (Wang A. et al., 2019; van Roeyen et al., 2019). Lastly, MF is significantly enriched for hormone binding, and hormones plays a role in RCC etiology. Hormone receptor expression in RCC cells has been demonstrated to be aberrant (Czarnecka et al., 2016).

Analysis of KEGG pathways revealed signalling pathways that promote cancer progression and resistance to therapies. The *SERPINE1* gene was enriched in the p53 signaling pathway, HIF-1 signaling pathway, and apelin signaling pathway. The interaction between P53 and HIF signaling can promote cancer progression (Zhang et al., 2021). While apelin signaling has also been linked to the development of cancer and its progression (Liu et al., 2021). It is thus noteworthy, that the survival analysis of *SERPINE1* expression in TCGA found a correlation between shorter survival, and the increased tumor grade, lymph node metastasis, and tumor stage (Guo et al., 2023). Therefore, *SERPINE1* plays a crucial role in the progression of KIRC. KIRC patients categorized as SS revealed high levels of *SERPINE1* gene expression, whereas LS displayed low levels of gene expression. Hence, the method tracked the progression of KIRC

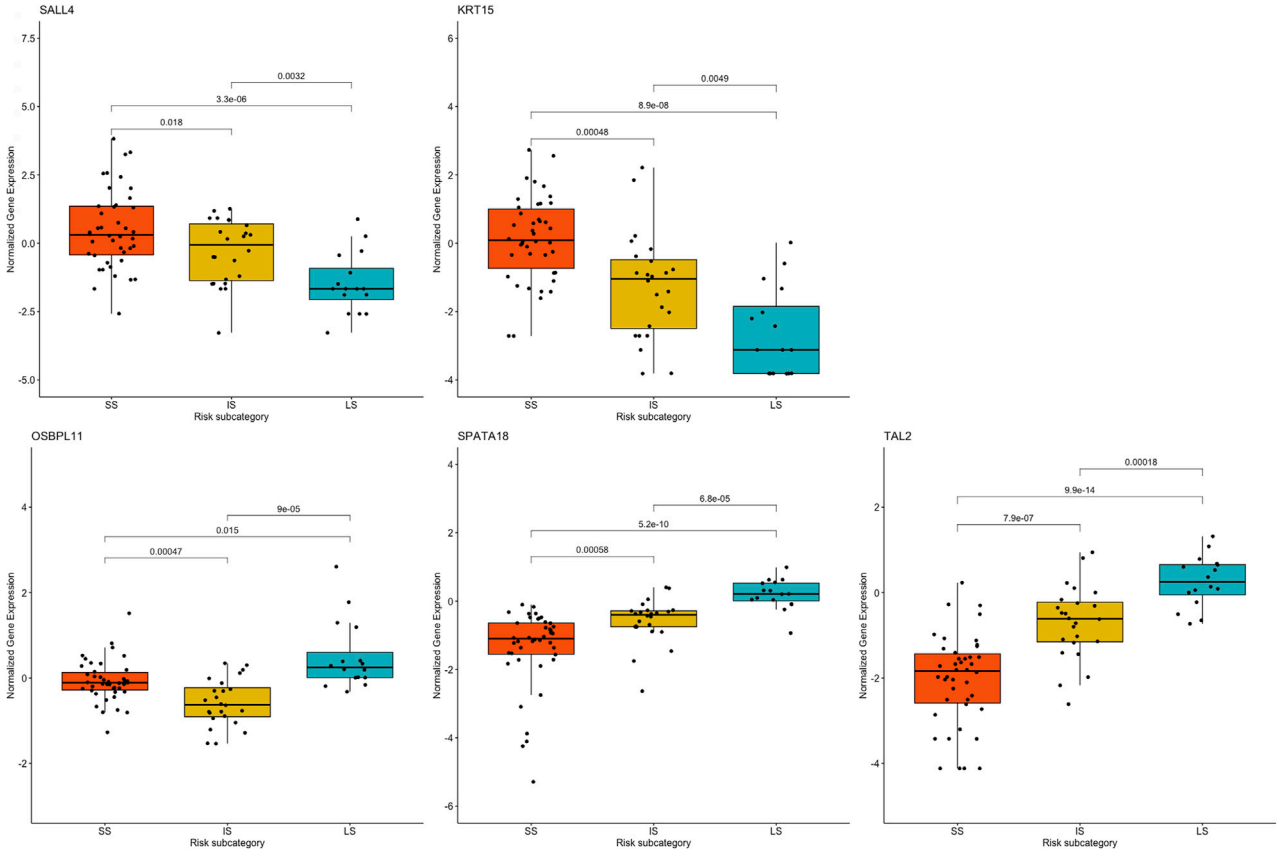


FIGURE 7
Boxplots based on risk subcategories of the five prognostic genes in KIRC patients. A boxplot was constructed with the normalized gene expression profile of each prognostic gene in all the samples that were categorized into the SS, IS, and LS categories.

and further indicated the potential of *SERPINE1* as a therapeutic target for KIRC patients.

Together with *SERPINE1,* the *PGK1* gene was also enriched for HIF-1 signaling pathway. HIF-1 is known to modulate a number of signaling pathways, having a significant impact on the cancer's response to radiotherapy (Huang and Zhou, 2020). Therefore, a viable approach for sensitization of KIRC to radiotherapy is to target *SERPINE1* and *PGK1*. Also, *PGK1* has been linked to several roles in the development of cancer, tumor progression, and drug resistance. The gene is known to promote sorafenib resistance, which is a first-line treatment for KIRC patients as a tyrosine kinase inhibitor. However, resistance to sorafenib significantly reduces the effectiveness of therapy (He et al., 2022). Therefore, the large patient group (n = 42), accounting for about half of the KIRC patients investigated in this study encompassed in SS, may be affected by this resistance to therapy.

Genes *KRT15* and *GPER1* enriched for estrogen signaling pathways can also serve as treatment targets for KIRC patients. Estrogen is known to inhibit the proliferation, migration, and infiltration of RCC cells as well as increase RCC apoptosis (Yu et al., 2013). This study illustrated that the downregulation of *KRT15* had favorable prognostic outcomes for KIRC patients for Cluster 2 and 3 (Figures 5, 7), whereas the downregulation of *GPER1* was linked to unfavorable prognosis in Cluster 1. Therefore, the two genes may serve as valuable prognostic markers for KIRC and a novel developmental approach for enhancing KIRC therapeutics.

This study further identified five prognostic genes as promising prognostic biomarkers and treatment targets for KIRC patients (Table 2). Cox regression together with Kaplan-Meier analyses confirmed the prognostic biomarkers and showed that patients with high levels of *SALL4* and *KRT15* gene expression have a poor survival outcome than patients with low levels of gene expression (Figure 5). While the high gene expression level of *OSBPL11, SPATA18*, and *TAL2* has a favorable survival outcome than patients with a low level of gene expression (Figure 6). Therefore, K-M confirmed that the five genes are effective at diagnosing KIRC patients and predicting prognosis.

The results are supported by previous research, which indicated that the high gene expression level of *SALL4* has a poor survival outcome in comparison to KIRC patients with a low gene expression level (Che et al., 2020). Also, data from Sun et al. (2020) showed that the downregulation of *SALL4* reduces KIRC tumor growth, metastasis, and angiogenesis. Therefore, it is noteworthy that Cluster 2 with intermediate survival followed a similar trend in cumulative survival probabilities as Cluster 1 with short survival (Figure 5). Furthermore, the high gene expression of *KRT15* has also been reported to correlate with a poor prognosis for RCC (Zhang et al., 2023). This study was able to detect *KRT15* as a prognostic gene in the KIRC subtype. The levels of gene expression correspond with the SS, IS, and LS (Figure 7). Previous studies have also reported higher

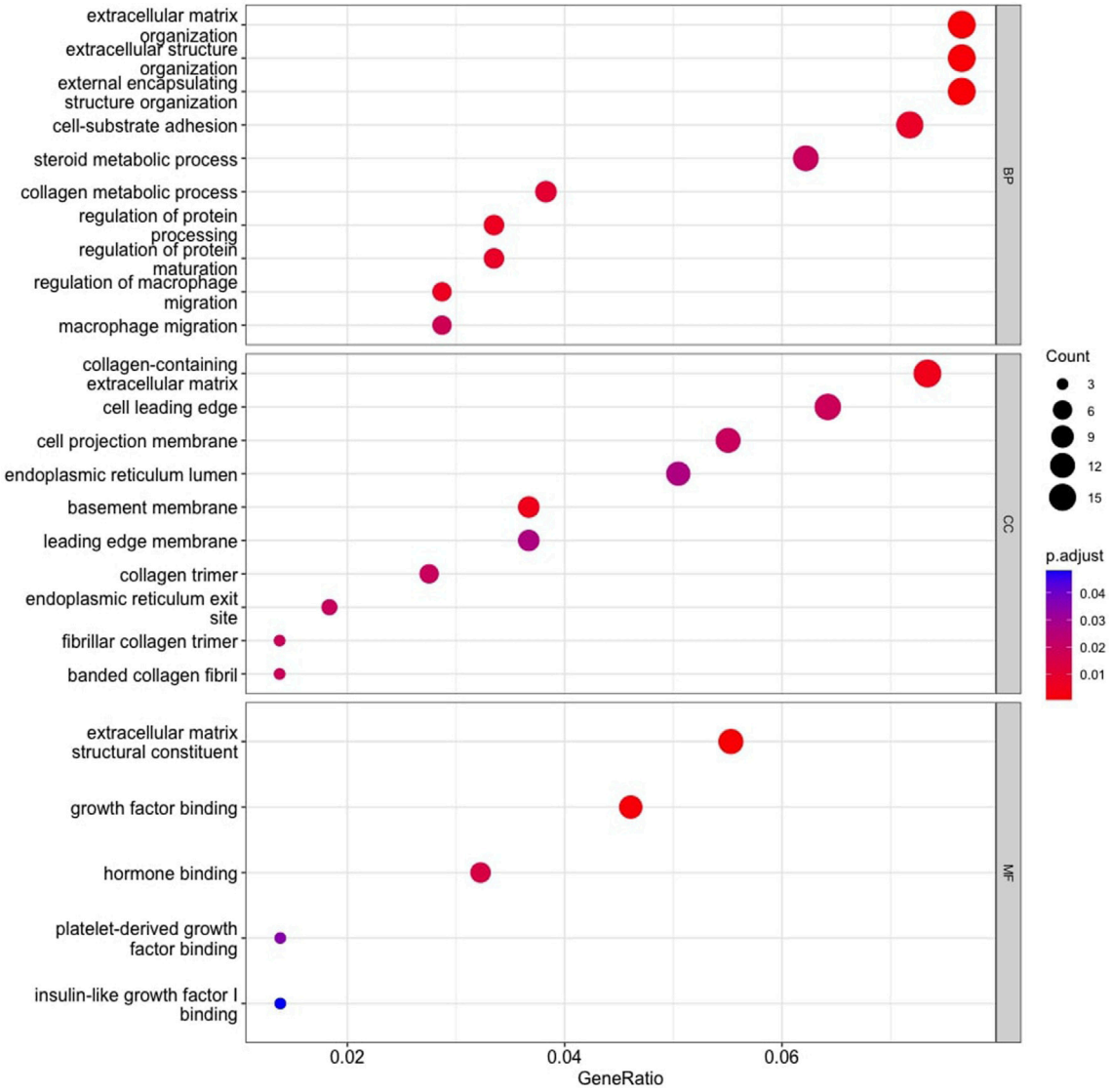


FIGURE 8
Gene Ontology enrichment analysis. Top 10 functional items of KIRC DEGs based on clusterProfiler. *Functional databases: BP, Biological process; CC, Cellular component; and MF, Molecular function.

levels of *SPATA18* gene expression associated with favorable OS in the KIRC subtype (Lingui et al., 2023) as well as in RCC (The human protein atlas, 2023a). High expression of *TAL2* has been reported with a favorable OS in RCC (The human protein atlas, 2023b). This is the first article to our knowledge to report *OSBPL11* as a prognostic biomarker. A similar observation as with the *SALL4* K-M curve is observed with the *OSBPL11* gene. The K-M curve of Cluster 2 followed a similar trend in cumulative survival probabilities as Cluster 1 (Figure 6). Therefore, the upregulation of *OSBPL11* could reduce KIRC progression.

ANOVA was used to assess the heterogeneity in the prognostic genes' mean gene expression profiles, to establish whether SS, IS, and LS samples' gene expression profiles differ from one another. The prognostic value of the five prognostic genes found was confirmed by ANOVA, which also indicated a statistically significant difference in gene expression between short- and long-term survival. A crucial discovery was made between the gene expression profiles in the intermediate- and long survival as well as intermediate- and short survival. ANOVA showed statistically significant differences between the gene expression profiles of both IS and LS, and IS and SS. This further validates the finding of an intermediate-survival group. The unique gene expression pattern of each of the five prognostic genes were further subjected to a validation step using the independent GEO dataset (Supplementary Material S1). This test dataset verified prognostic genes *OSBPL11* and *TAL2* in the GEO dataset illustrated a similar gene expression pattern for cluster 1 (short survival) and cluster 3 (long survival). The remaining three prognostic genes, *SALL4*, *KRT15*, and *SPATA18* showed similar gene expression patterns for all three clusters (Supplementary Material S1). The five prognostic genes are therefore essential as they may enable an improved KIRC patient prognosis based on the gene expression level of the five genes. Hence, this discovery is important as it is directly correlated with survival and could aid in predicting the outcome of KIRC patients.

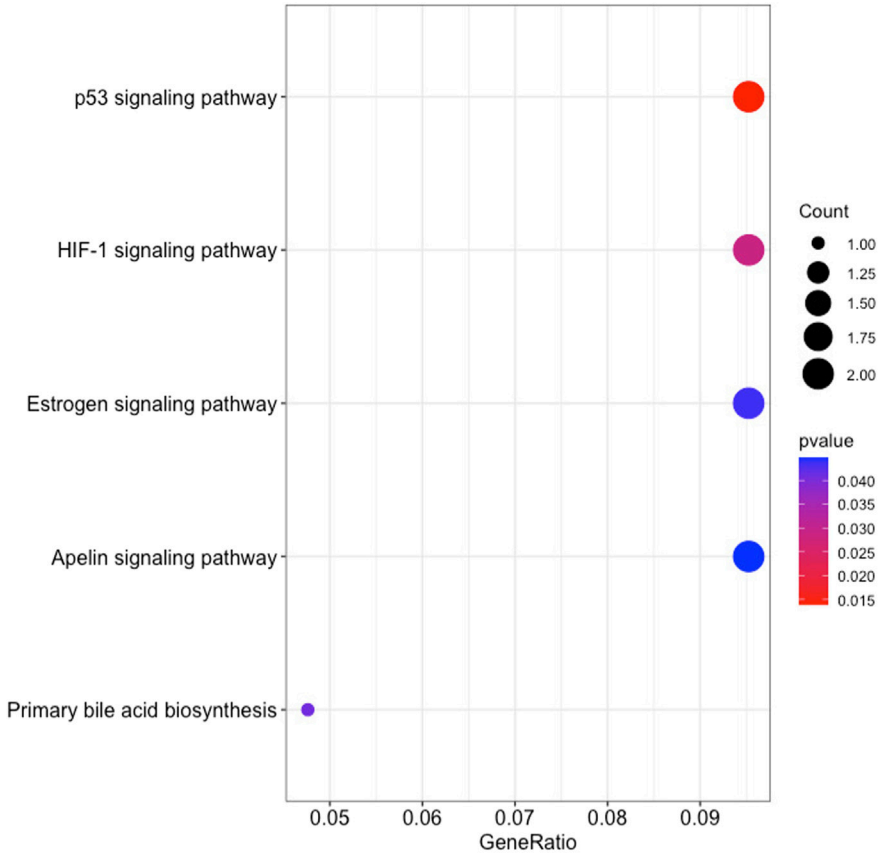


FIGURE 9
The results of KEGG pathways enrichment analysis of the 48 RFE gene subset based on clusterProfiler.

The investigation detected molecular mechanisms that allowed for the segregation of three unique cancer clusters (subtypes) that progress differently in gene expression profiles and correlate with KIRC patient survival. Therefore, the normalization method was successfully implemented in this study and hierarchical clustering was able to provide an accurate assessment of the heterogeneity of KIRC. The cellular functions detected by GO enrichment along with the pathogenic genes detected by KEGG pathway analysis further confirmed the contribution to the progression of the disease. Additionally, the heterogeneity of KIRC served as a fuel for therapy resistance and emphasized the urgent need to expand the clinical subtypes for KIRC patients. As a result, this investigation facilitated and contributed to the current KIRC cancer classification with in-depth patient subtyping. The discovery of the five prognostic genes, combined with the biomarkers detected in pathway analysis, can provide a more accurate prognosis, and serve as targets to provide a more effective therapeutic approach for KIRC patients.

5 Conclusion

The implemented normalization method has the potential to reveal cancer patients that progress differently (subtypes) and establish a genotype-phenotype relationship between the identified subtypes and the patient's OS. In this study, correlations between the risk subcategories and gene signatures differentiated short, intermediate, and long survival in KIRC patients. The prognostic capacity of the prognostic genes can successfully classify and predict the prognosis of KIRC patients. Moreover, the prognostic genes were able to segregate patients into additional survival subcategories and thus provide targets that can enhance patient prognosis and aid in the development of individualized treatment approaches.

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. The source code for the implementation of reproducibility of the analyses for the study is available in GitHub: https://github.com/LiveseyM/KIRC_Subtyping.git.

Author contributions

ML: Conceptualization, Data curation, Formal Analysis, Investigation, Methodology, Software, Validation, Visualization, Writing–original draft, Writing–review and editing. NE: Data

curation, Software, Validation, Visualization, Writing–review and editing. HB: Conceptualization, Data curation, Formal Analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Writing–review and editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. The work reported herein was made possible through funding by the Cancer Research Trust, Faculty of Health Sciences, Start-up Emerging Researcher Award from the University of Cape Town, and the South African National Research Foundation (NRF Grant ID 121787) for ML bursary.

Acknowledgments

We extend our appreciation to the University of Cape Town and National Research Foundation of South Africa (NRF) for funding.

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/fgene.2023.1291043/full#supplementary-material

References

Aken, B. L., Ayling, S., Barrell, D., Clarke, L., Curwen, V., Fairley, S., et al. (2016). The Ensembl gene annotation system. *Database* 2016, baw093. doi:10.1093/database/baw093

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., et al. (2000). Gene ontology: tool for the unification of biology. The gene ontology consortium. *Nat. Genet.* 25 (1), 25–29. doi:10.1038/75556

Caliskan, A., Andac, A. C., and Arga, K. Y. (2020). Novel molecular signatures and potential therapeutics in renal cell carcinomas: insights from a comparative analysis of subtypes. *Genomics* 112 (5), 3166–3178. doi:10.1016/j.ygeno.2020.06.003

Casuscelli, J., Vano, Y. A., Fridman, W. H., and Hsieh, J. J. (2017). Molecular classification of renal cell carcinoma and its implication in future clinical practice. *Kidney cancerClift. Va.)* 1 (1), 3–13. doi:10.3233/KCA-170008

Che, J., Wu, P., Wang, G., Yao, X., Zheng, J., and Guo, C. (2020). Expression and clinical value of SALL4 in renal cell carcinomas. *Mol. Med. Rep.* 22 (2), 819–827. doi:10.3892/mmr.2020.11170

Czarnecka, A. M., Niedzwiedzka, M., Porta, C., and Szczylik, C. (2016). Hormone signaling pathways as treatment targets in renal cell cancer (Review). *Int. J. Oncol.* 48 (6), 2221–2235. doi:10.3892/ijo.2016.3460

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *J. Stat. Softw.* 33 (1), 1–22. doi:10.18637/jss.v033.i01

Goldman, M. J., Craft, B., Hastie, M., Repečka, K., McDade, F., Kamath, A., et al. (2020). Visualizing and interpreting cancer genomics data via the Xena platform. *Nat. Biotechnol.* 38 (6), 675–678. doi:10.1038/s41587-020-0546-8

Gray, R. E., and Harris, G. T. (2019). Renal cell carcinoma: diagnosis and management. *Am. Fam. physician* 99 (3), 179–184.

Guo, L., An, T., Wan, Z., Huang, Z., and Chong, T. (2023). SERPINE1 and its co-expressed genes are associated with the progression of clear cell renal cell carcinoma. *BMC Urol.* 23 (1), 43. doi:10.1186/s12894-023-01217-6

Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Mach. Learn.* 46 (1–3), 389–422. doi:10.1023/a:1012487302797

He, Y., Wang, X., Lu, W., Zhang, D., Huang, L., Luo, Y., et al. (2022). PGK1 contributes to tumorigenesis and sorafenib resistance of renal clear cell carcinoma via activating CXCR4/ERK signaling pathway and accelerating glycolysis. *Cell. death Dis.* 13 (2), 118. doi:10.1038/s41419-022-04576-4

Hu, F., Zeng, W., and Liu, X. (2019). A gene signature of survival prediction for kidney renal cell carcinoma by multi-omic data analysis. *Int. J. Mol. Sci.* 20 (22), 5720. doi:10.3390/ijms20225720

Huang, R., and Zhou, P.-K. (2020). HIF-1 signaling: a key orchestrator of cancer radioresistance. *Radiat. Med. Prot.* 1 (1), 7–14. doi:10.1016/j.radmp.2020.01.006

Jaskowiak, P. A., Campello, R. J., and Costa, I. G. (2014). On the selection of appropriate distances for gene expression data clustering. *BMC Bioinforma.* 15 (2), S2. doi:10.1186/1471-2105-15-S2-S2

Kassambara, A., and Mundt, F. (2020). Factoextra: extract and visualize the results of multivariate data analyses. R Package Version. 1.0.7 https://CRAN.R-project.org/package=factoextra.

Kovacs, G., Akhtar, M., Beckwith, B. J., Bugert, P., Cooper, C. S., Delahunt, B., et al. (1997). The Heidelberg classification of renal cell tumours. *J. pathology* 183 (2), 131–133. doi:10.1002/(SICI)1096-9896(199710)183:2<131::AID-PATH931>3.0.CO;2-G

Law, C. W., Alhamdoosh, M., Su, S., Dong, X., Tian, L., Smyth, G. K., et al. (2016). RNA-seq analysis is easy as 1-2-3 with limma, Glimma and edgeR. *F1000Research* 5, 1408. ISCB Comm J-1408. doi:10.12688/f1000research.9005.1

Lê, S., Josse, J., and Husson, F. (2008). FactoMineR: an R package for multivariate analysis. *J. Stat. Softw.* 25 (1), 1–18. doi:10.18637/jss.v025.i01

Lingui, X., Weifeng, L., Yufei, W., and Yibin, Z. (2023). High SPATA18 expression and its diagnostic and prognostic value in clear cell renal cell carcinoma. *Med. Sci. Monit.* 29, e938474. doi:10.12659/MSM.938474

Liu, L., Yi, X., Lu, C., Wang, Y., Xiao, Q., Zhang, L., et al. (2021). Study progression of apelin/APJ signaling and apela in different types of cancer. *Front. Oncol.* 11, 658253. doi:10.3389/fonc.2021.658253

Livesey, M., Rossouw, S. C., Blignaut, R., Christoffels, A., and Bendou, H. (2023). Transforming RNA-Seq gene expression to track cancer progression in the multi-stage early to advanced-stage cancer development. *PloS one* 18 (4), e0284458. doi:10.1371/journal.pone.0284458

Lopez-Beltran, A., Scarpelli, M., Montironi, R., and Kirkali, Z. (2006). 2004 WHO classification of the renal tumors of the adults. *Eur. Urol.* 49 (5), 798–805. doi:10.1016/j.eururo.2005.11.035

Moch, H., Humphrey, P. A., Ulbright, T. M., and Reuter, V. E. (2016). *WHO classification of tumours of the urinary system and male genital organs*. 4th ed. Lyon (France): International Agency for Research on Cancer.

Mohanty, S. K., Lobo, A., and Cheng, L. (2023). The 2022 revision of the World Health Organization classification of tumors of the urinary system and male genital organs: advances and challenges. *Hum. Pathol.* 136, 123–143. doi:10.1016/j.humpath.2022.08.006

Oxburgh, L. (2022). The extracellular matrix environment of clear cell renal cell carcinoma. *Cancers* 14 (17), 4072. doi:10.3390/cancers14174072

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn Machine learning in python. *J. Mach. Learn. Res.* 12, 2825–2830.

Puzanov, G. A. (2022). Identification of key genes of the ccRCC subtype with poor prognosis. *Sci. Rep.* 12 (1), 14588. doi:10.1038/s41598-022-18620-y

Ritchie, M. E., Phipson, B., Wu, D., Hu, Y., Law, C. W., Shi, W., et al. (2015). Limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic acids Res.* 43 (7), e47. doi:10.1093/nar/gkv007

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* 20, 53–65. doi:10.1016/0377-0427(87)90125-7

Simon, N., Friedman, J., Hastie, T., and Tibshirani, R. (2011). Regularization paths for cox's proportional hazards model via coordinate descent. *J. Stat. Softw.* 39 (5), 1–13. doi:10.18637/jss.v039.i05

Smedley, D., Haider, S., Durinck, S., Pandini, L., Provero, P., Allen, J., et al. (2015). The BioMart community portal: an innovative alternative to large, centralized data repositories. *Nucleic acids Res.* 43 (W1), W589–W598. doi:10.1093/nar/gkv350

Srigley, J. R., Delahunt, B., Eble, J. N., Egevad, L., Epstein, J. I., Grignon, D., et al. (2013). The international society of urological pathology (ISUP) vancouver classification of renal neoplasia. *Am. J. Surg. pathology* 37 (10), 1469–1489. doi:10.1097/PAS.0b013e318299f2d1

Sun, J., Tang, Q., Gao, Y., Zhang, W., Zhao, Z., Yang, F., et al. (2020). VHL mutation-mediated SALL4 overexpression promotes tumorigenesis and vascularization of clear cell renal cell carcinoma via Akt/GSK-3β signaling. *J. Exp. Clin. cancer Res.* 39 (1), 104. doi:10.1186/s13046-020-01609-8

The human protein atlas (2023a). *Human pathology atlas. SPATA18 gene*. Available From: https://www.proteinatlas.org/ENSG00000186051-TAL2/pathology/renal+cancer (Accessed August 20, 2023).

The human protein atlas (2023b). *Human pathology atlas. TAL2 gene*. Available From: https://www.proteinatlas.org/ENSG00000163071-SPATA18/pathology/renal+cancer (Accessed August 20, 2023).

Tibshirani, R., Bien, J., Friedman, J., Hastie, T., Simon, N., Taylor, J., et al. (2012). Strong rules for discarding predictors in lasso-type problems. *J. R. Stat. Soc. Ser. B, Stat. Methodol.* 74 (2), 245–266. doi:10.1111/j.1467-9868.2011.01004.x

Trpkov, K., Williamson, S. R., Gill, A. J., Adeniran, A. J., Agaimy, A., Alaghehbandan, R., et al. (2021a). Novel, emerging and provisional renal entities: the Genitourinary Pathology Society (GUPS) update on renal neoplasia. *Mod. Pathol.* 34 (6), 1167–1184. doi:10.1038/s41379-021-00737-6

Trpkov, K., Hes, O., Williamson, S. R., Adeniran, A. J., Agaimy, A., Alaghehbandan, R., et al. (2021b). New developments in existing WHO entities and evolving molecular concepts: the Genitourinary Pathology Society (GUPS) update on renal neoplasia. *Mod. pathology official J. U. S. Can. Acad. Pathology, Inc* 34 (7), 1392–1424. doi:10.1038/s41379-021-00779-w

Tukey, J. W. (1949). Comparing individual means in the analysis of variance. *Int. Biom. Soc.* 5 (2), 99–114. doi:10.2307/3001913

Udager, A. M., and Mehra, R. (2016). Morphologic, molecular, and taxonomic evolution of renal cell carcinoma: a conceptual perspective with emphasis on updates to the 2016 World Health organization classification. *Archives pathology laboratory Med.* 140 (10), 1026–1037. doi:10.5858/arpa.2016-0218-RA

van Roeyen, C. R. C., Martin, I. V., Drescher, A., Schuett, K. A., Hermert, D., Raffetseder, U., et al. (2019). Identification of platelet-derived growth factor C as a mediator of both renal fibrosis and hypertension. *Kidney Int.* 95 (5), 1103–1119. doi:10.1016/j.kint.2018.11.031

Wang, Q., Zhang, H., Chen, Q., Wan, Z., Gao, X., and Qian, W. (2019). Identification of METTL14 in kidney renal clear cell carcinoma using bioinformatics analysis. *Dis. markers* 2019, 5648783. doi:10.1155/2019/5648783

Wang, A., Chen, M., Wang, H., Huang, J., Bao, Y., Gan, X., et al. (2019). Cell adhesion-related molecules play a key role in renal cancer progression by multinetwork analysis. *BioMed Res. Int.* 2019, 2325765. doi:10.1155/2019/2325765

Ward, J. H. (1963). Hierarchical grouping to optimize an objective function. Journal of American *Statistical Association. J. Am. Stat. Assoc.* 58, 236–244. doi:10.1080/01621459.1963.10500845

Xie, D., Mao, Y., Du, N., Ji, H., and Li, J. (2022). Macrophages promote growth, migration and epithelial-mesenchymal transition of renal cell carcinoma by regulating GSDMD/IL-1β axis. *Cytokine* 159, 156021. doi:10.1016/j.cyto.2022.156021

Xiong, X., Chen, C., Yang, J., Ma, L., Wang, X., Zhang, W., et al. (2022). Characterization of the basement membrane in kidney renal clear cell carcinoma to guide clinical therapy. *Front. Oncol.* 12, 1024956. doi:10.3389/fonc.2022.1024956

Yin, L., Li, W., Wang, G., Shi, H., Wang, K., Yang, H., et al. (2019). NR1B2 suppress kidney renal clear cell carcinoma (KIRC) progression by regulation of LATS 1/2-YAP signaling. *J. Exp. Clin. Cancer Res.* 38, 343. doi:10.1186/s13046-019-1344-3

You, Y., Ren, Y., Liu, J., and Qu, J. (2021). Promising epigenetic biomarkers associated with cancer-associated-fibroblasts for progression of kidney renal clear cell carcinoma. *Front. Genet.* 12, 736156. doi:10.3389/fgene.2021.736156

Yu, G., Wang, L. G., Han, Y., and He, Q. Y. (2012). clusterProfiler: an R package for comparing biological themes among gene clusters. *Omics a J. Integr. Biol.* 16 (5), 284–287. doi:10.1089/omi.2011.0118

Yu, C. P., Ho, J. Y., Huang, Y. T., Cha, T. L., Sun, G. H., Yu, D. S., et al. (2013). Estrogen inhibits renal cell carcinoma cell progression through estrogen receptor-β activation. *PloS one* 8 (2), e56667. doi:10.1371/journal.pone.0056667

Zhang, C., Liu, J., Wang, J., Zhang, T., Xu, D., Hu, W., et al. (2021). The interplay between tumor suppressor p53 and hypoxia signaling pathways in cancer. *Front. Cell. Dev. Biol.* 9, 648808. doi:10.3389/fcell.2021.648808

Zhang, W., Chen, P., Li, Z., Zhang, R., and Zhang, J. (2023). Clinical implication of keratin-15 quantification for renal cell carcinoma management: its dysregulation and association with clinicopathologic characteristics and prognostication. *Tohoku J. Exp. Med.* 260 (2), 99–107. doi:10.1620/tjem.2023.J017

Znaor, A., Lortet-Tieulent, J., Laversanne, M., Jemal, A., and Bray, F. (2015). International variations and trends in renal cell carcinoma incidence and mortality. *Eur. Urol.* 67 (3), 519–530. doi:10.1016/j.eururo.2014.10.002

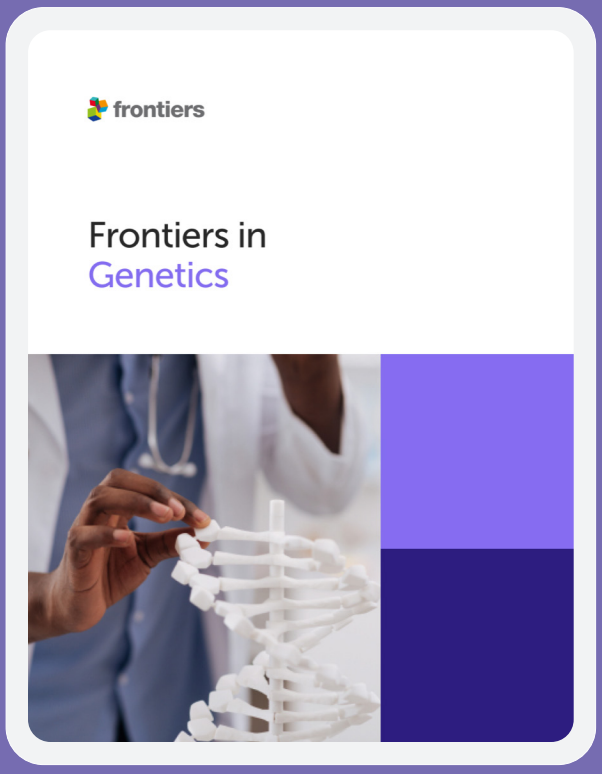
Frontiers in Genetics

Highlights genetic and genomic inquiry relating to all domains of life

The most cited genetics and heredity journal, which advances our understanding of genes from humans to plants and other model organisms. It highlights developments in the function and variability of the genome, and the use of genomic tools.

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