Wei Peng
Zhipeng Cai
Pavel Skums (Eds.)

LNBI 14956

Bioinformatics Research and Applications

20th International Symposium, ISBRA 2024
Kunming, China, July 19–21, 2024
Proceedings, Part III

3 Part III

Springer

MOREMEDIA

Lecture Notes in Computer Science

Lecture Notes in Bioinformatics 14956

Series Editors

Sorin Istrail, *Brown University, Providence, USA*
Pavel Pevzner, *University of California, San Diego, USA*
Michael Waterman, *University of Southern California, Los Angeles, USA*

Editorial Board Members

Søren Brunak, *Technical University of Denmark, Kongens Lyngby, Denmark*
Mikhail S. Gelfand, *IITP, Research and Training Center on Bioinformatics, Moscow, Russia*
Thomas Lengauer, *Max Planck Institute for Informatics, Saarbrücken, Germany*
Satoru Miyano, *University of Tokyo, Tokyo, Japan*
Eugene Myers, *Max Planck Institute of Molecular Cell Biology and Genetics, Dresden, Germany*
Marie-France Sagot, *Université Lyon 1, Villeurbanne, France*
David Sankoff, *University of Ottawa, Ottawa, ON, Canada*
Ron Shamir, *Tel Aviv University, Ramat Aviv, Israel*
Terry Speed, *Walter and Eliza Hall Institute of Medical Research, Melbourne, Australia*
Martin Vingron, *Max Planck Institute for Molecular Genetics, Berlin, Germany*
W. Eric Wong, *University of Texas at Dallas, Richardson, USA*

The series Lecture Notes in Bioinformatics (LNBI) was established in 2003 as a topical subseries of LNCS devoted to bioinformatics and computational biology.

The series publishes state-of-the-art research results at a high level. As with the LNCS mother series, the mission of the series is to serve the international R & D community by providing an invaluable service, mainly focused on the publication of conference and workshop proceedings and postproceedings.

Wei Peng · Zhipeng Cai · Pavel Skums
Editors

Bioinformatics Research and Applications

20th International Symposium, ISBRA 2024 Kunming, China, July 19–21, 2024 Proceedings, Part III

Springer

Editors
Wei Peng
Kunming University of Science and Technology
Kunming, China

Zhipeng Cai
Georgia State University
Atlanta, GA, USA

Pavel Skums
University of Connecticut
Storrs, CT, USA

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Bioinformatics
ISBN 978-981-97-5086-3 ISBN 978-981-97-5087-0 (eBook)
https://doi.org/10.1007/978-981-97-5087-0

LNCS Sublibrary: SL8 – Bioinformatics

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2024

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

If disposing of this product, please recycle the paper.

Preface

On behalf of the Program Committee, we would like to welcome you to the proceedings of the 20th International Symposium on Bioinformatics Research and Applications (ISBRA 2024), held in Kunming, China, July 19–21, 2024. The symposium provides a forum for the exchange of ideas and results among researchers, developers, and practitioners working on all aspects of bioinformatics and computational biology and their applications.

This year, we received 236 submissions in response to the call for extended abstracts. Each submission was reviewed by at least three reviewers. The Program Committee decided to accept 93 of these for full publication in the proceedings; a list of these contributions can be found in this front matter.

The technical program also featured keynote talks delivered by four distinguished speakers: Can Yang from Hong Kong University of Science and Technology, China; Bin Ma from University of Waterloo, Canada; Hongbing Shen from the Shanghai Jiao Tong University, China; Xiaowo Wang from Tsinghua University, China; Jing Tang from University of Helsinki, Finland.

We would like to thank the Program Committee members and the additional reviewers for volunteering their time to review and discuss symposium papers. We would like to extend special thanks to the steering and general chairs of the symposium for their leadership, and to the finance, publicity, workshops, local organization, and publications chairs for their hard work in making ISBRA 2024 a successful event. Last but not least, we would like to thank all authors for presenting their work at the symposium.

June 2024

Zhipeng Cai
Pavel Skums
Wei Peng

Organization

Steering Committee

Dan Gusfield	UC Davis, USA
Ion Mandoiu	UConn, USA
Yi Pan (Chair)	SIAT, China
Marie-France Sagot	Inria, France
Zhirong Sun	Tsinghua University, China
Ying Xu	UGA, USA
Aidong Zhang	UVA, USA
Zhipeng Cai	GSU, USA

General Chairs

Yuyu Niu	KUST, China
Jianxin Wang	CSU, China
Alexander Zelikovsky	GSU, USA

Program Chairs

Zhipeng Cai	GSU, USA
Pavel Skums	UConn, USA
Wei Peng	KUST, China

Publicity Chairs

Min Li	CSU, China
Yanjie Wei	SIAT, China
Xuefeng Cui	SDU, China

Publication Chairs

Jin Liu	CSU, China
Xiujuan Lei	SNNU, China

Web Chairs

Hongdong Li	CSU, China
Wei Dai	KUST, China

Poster Chair

Wei Lan	GXU, China

Program Committee

Juanying Xie	Shanxi Normal University, China
Hisham Al-Mubaid	University of Houston - Clear Lake, USA
Ying An	Central South University, China
Mukul Bansal	University of Connecticut, USA
Mahua Bhattacharya	IIIT Gwalior, India
Xia-an Bi	Hunan Normal University, China
Dan Brown	University of Waterloo, Canada
Yunpeng Cai	Shenzhen Institutes of Advanced Technology, China
Rita Casadio	University of Bologna, Italy
Bolin Chen	Northwestern Polytechnical University, China
Hebing Chen	Institute of Health Service and Transfusion Medicine, China
Xiang Chen	Hunan University of Science and Technology, China
Jianhong Cheng	Guizhou Aerospace Institute of Measuring and Testing Technology, China
Young-Rae Cho	Yonsei University, South Korea
Xuefeng Cui	Shandong University, China
Xiaojun Ding	Yulin Normal University, China
Lei Du	Northwestern Polytechnical University, China
Oliver Eulenstein	Iowa State University, USA
Lívia Gama	University of São Paulo, Brazil
Jin Gu	Tsinghua University, China
Fei Guo	Central South University, China
Guosheng Han	Xiangtan University, China
Zengyou He	Dalian University of Technology, China
Steffen Heber	North Carolina State University, USA
Kai Hu	Xiangtan University, China

Bruno Iha	University of São Paulo, Brazil
Jicai Jiang	North Carolina State University, USA
Wooyoung Kim	University of Washington Bothell, USA
Xiangyong Kong	University of Shanghai for Science and Technology, China
Danny Krizanc	Wesleyan University, USA
Hulin Kuang	Central South University, China
Pavel Kuksa	University of Pennsylvania, USA
Kiril Kuzmin	Georgia State University, USA
Manuel Lafond	Université de Sherbrooke, Canada
Wei Lan	Guangxi University, China
Zhang Le	Sichuan University, China
Xiujuan Lei	Shanxi Normal University, China
Hong-Dong Li	Central South University, China
Min Li	Central South University, China
Xiaobo Li	Lishui University, China
Xingyi Li	Northwestern Polytechnical University, China
Yaohang Li	Old Dominion University, USA
Xingyu Liao	Northwestern Polytechnical University, China
Jin Liu	Central South University, China
Juan Liu	Wuhan University, China
Liangliang Liu	Henan Agricultural University, China
Weiguo Liu	Shandong University, China
Xiaowen Liu	Tulane University, USA
Zhendong Liu	University of Shanghai for Science and Technology, China
Zhi-Ping Liu	Shandong University, China
Chengqian Lu	Xiangtan University, China
Huimin Luo	Henan University, China
Junwei Luo	Henan Polytechnic University, China
Ion Mandoiu	University of Connecticut, USA
Xiangmao Meng	Xiangtan University, China
Wenwen Min	Yunnan University, China
Wancen Mu	University of North Carolina at Chapel Hill, USA
Rafael Nascimento	University of São Paulo, Brazil
Beifang Niu	Computer Network Information Center, Chinese Academy of Sciences, China
Le Ou-Yang	Shenzhen University, China
Murray Patterson	Georgia State University, USA
Wei Peng	Kunming University of Science and Technology, China
Xiaoqing Peng	Central South University, China

Wu Qiu	Huazhong University of Science and Technology, China
Bikram Sahoo	Georgia State University, USA
João Setubal	University of São Paulo, Brazil
Junliang Shang	Qufu Normal University, China
Jian-yu Shi	Northwestern Polytechnical University, China
Xinghua Shi	Temple University, USA
Gianluca Silva	University of São Paulo, Brazil
Arthur Solano	University of São Paulo, Brazil
Mingzhou Song	New Mexico State University, USA
Huiyan Sun	Jilin University, China
Jiarui Sun	Southeast University, China
Shiwei Sun	Institute of Computing & Technology, CAS, China
Weitian Tong	Georgia Southern University, USA
Tomas Vinar	Comenius University in Bratislava, Slovakia
Han Wang	Northeast Normal University, China
Hong-Qiang Wang	University of Science and Technology of China, China
Jianxin Wang	Central South University, China
Jiayin Wang	Xi'an Jiaotong University, China
Juan Wang	Inner Mongolia University, China
Kaili Wang	Donghua University, China
Shunfang Wang	Yunnan University, China
Xinyue Wang	Rutgers University, USA
Ying Wang	Xiamen University, China
Yanjie Wei	Shenzhen Institute of Advanced Technology, China
Ka-Chun Wong	City University of Hong Kong, China
Fang-Xiang Wu	University of Saskatchewan, Canada
Hongyan Wu	Shenzhen Institutes of Advanced Technology, China
Jingli Wu	Guangxi Normal University, China
Ju Xiang	Changsha University of Science and Technology, China
Minzhu Xie	Hunan Normal University, China
Yuying Xie	Michigan State University, USA
Guangzhi Xiong	University of Virginia, USA
Cheng Yan	Hunan University of Chinese Medicine, China
Yang Yang	Shanghai Jiao Tong University, China
Yuedong Yang	Sun Yat-sen University, China
Yusen Ye	Xidian University, China

Liang Yu	Xidian University, China
Feng Zeng	Xiamen University, China
Min Zeng	Central South University, China
Cheng Zhang	Peking University, China
Eric Lu Zhang	Hong Kong Baptist University, China
Fa Zhang	Beijing Institute of Technology, China
Fuhao Zhang	Northwest A&F University, China
Han Zhang	Nankai University, China
Houwang Zhang	City University of Hong Kong, China
Wen Zhang	Huazhong Agricultural University, China
Yiming Zhang	University of Connecticut, USA
Yongqing Zhang	Chengdu University of Information Technology, China
Ruiqing Zheng	Central South University, China
Jiancheng Zhong	Hunan Normal University, China

Contents – Part III

Feddaw: Dual Adaptive Weighted Federated Learning for Non-IID Medical Data

Linan Ren[1], Kaixin Li[1], Ying An[1], Yuan Liu[2], and Xianlai Chen[1,3(✉)]

[1] Big Data Institute, Central South University, Changsha 410083, China
chenxianlai@csu.edu.cn
[2] School of Computer Science and Engineering, Central South University, Changsha 410083, China
[3] Key Laboratory of Medical Information Research (Central South University), College of Hunan Province, Changsha, China

Abstract. The use of deep learning methods in disease diagnosis holds great promise with the development of medical big data. However, the scale of parameters in deep learning models, which can often reach millions, requires learning from large and diverse medical datasets to achieve the accuracy required for clinical applications. The challenges of cross-domain, decentralization, and data privacy in medical data have constrained the development of this field. Federated learning (FL) addresses these challenges by exchanging model parameters between clients and servers to share the model. However, in the case of medical data, there may be significant disparities in data quality among medical institutions, leading to imbalances in data volume and labeling, which may significantly affect model performance. Traditional FL approaches typically use simple methods such as averaging or weighted averaging during the parameter aggregation process, ignoring the Non-IID (Non-Independent and Identically Distributed) problem among clients. In this paper, a novel FL approach, Feddaw, is proposed based on the characteristics of non-IID medical data distribution. Feddaw aims to reduce the negative impact of label distribution shift in medical data by limiting the probability weighting factor of the CNN classification layer during client-side local training. Additionally, it verifies the accuracy of the client-side model in each round at the server-side, using accuracy-based weight aggregation to balance the negative impact of different data sample shifts. The experimental results show that the proposed Feddaw outperforms traditional FL methods in medical disease diagnosis.

Keywords: Federated Learning · Non-IID · Medical Data · Disease Diagnosis

1 Introduction

The sharing economy has been propelled by the explosive growth of technologies such as the internet, big data, mobile edge computing, and artificial intelligence. The medical industry is gradually becoming the new focus of the sharing economy, as huge amounts of medical data are generated and the demands of social progress evolve. The use of machine

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2024
W. Peng et al. (Eds.): ISBRA 2024, LNBI 14956, pp. 1–13, 2024.
https://doi.org/10.1007/978-981-97-5087-0_1

learning and deep learning methods for disease diagnosis shows potential in the medical industry, which is beneficial for improving medical practice and optimizing the allocation of limited medical resources, thereby advancing the construction of precision medicine. However, in order to learn parameters for training disease diagnosis models using deep learning, the disease diagnosis task requires a large amount of medical data as a training base. The dataset from a single medical institution is often insufficient to support the training of a high-performance disease diagnosis model due to its limited scale and diversity. Federated learning (FL) [1] is a novel machine learning method that aims to address data management and privacy protection issues by collaboratively training models from multiple endpoints without sharing the data itself. However, applying FL to disease diagnosis faces a pressing problem: the Non-Independent and Identically Distributed (Non-IID) data [2] in medical data. The complexity of the Non-IID problem in the medical field arises from the significant variation in data patterns, dimensions, and general features among different medical institutions. Furthermore, differences in detection methods, patient characteristics, indicator ranges, medical equipment, and patient populations result in significant differences in data distributions. As a result, the overall performance of FL decreases in the medical field compared to other fields [3].

Recently, there has been a significant focus on addressing the negative impact of Non-IID data on FL in both domestic and foreign research. Currently, there are three main approaches [4] to address the non-IID problem: reducing transmission parameters to optimize transmission processes, assimilating data distribution, and evaluation and selection. Kim et al. [5] proposed federated distillation and federated augmentation methods, which enable clients and servers to exchange model outputs without exchanging model parameters. This reduces the size of the model outputs compared to the original parameters. Li et al. [6] attempted to reduce the number of machine learning model parameters using the Lottery Ticket hypothesis. These methods reduce the number of model parameters that need to be transferred between the server and clients. However, these methods may result in a loss of accuracy as some abandoned parameters may still contribute to the FL process. Li et al. [7] proposed the FedProx algorithm, which executes a variable number of SGD algorithms based on the available system resources of client devices. This method shortens the convergence time and compresses the model update data, making it more suitable for FL scenarios with varying client data quality and computational resources. In their approach to assimilating data distributions, Zhao et al. [8] provided all clients with a small amount of identical data, thereby making the data distributions of clients more similar. However, this approach poses potential privacy risks, increasing the likelihood of data leakage. In evaluating and selecting approaches to differentiate client contributions, Wang et al. [9] used reinforcement learning to evaluate and select approaches for differentiating client contributions. They selected clients that made significant contributions and allowed them to continue iterations, while excluding others from participating in subsequent rounds of iteration. Kopparapu et al. [10] proposed the FedCD algorithm, which selects high-contributing clients and assigns weights based on the accuracy of client tests. However, this approach results in each client receiving numerous model parameters, leading to higher communication costs during the FL process.

Based on existing research, the Non-IID issues in medical data resulting from client-side data heterogeneity in FL are commonly of two types: label distribution shift and data volume shift. This is due to the disparities in data sizes across disparate medical institutions, as well as the varying water quality, living conditions, and medical conditions in distinct regions, which give rise to pronounced regionalisation of some diseases. CNN (Convolutional Neural Network) plays a significant role in disease diagnosis [11], particularly in the field of disease diagnosis through image classification using medical imaging data, demonstrating superior performance. Therefore, based on CNN, we propose a Dual Adaptive Weighted Federated Learning (*Feddaw*) in the application of disease diagnosis to address the problems of label distribution shift and data volume shift in medical datasets from different medical institutions. Specifically, this method addresses the issue of label distribution shift by limiting the scaling operation of the softmax layer [12] during the local training of the CNN. Additionally, it verifies the accuracy of the client models during the global model fusion process and dynamically adjusts weights to solve the data volume shift problem. Finally, we conducted multiple sets of comparative experiments with other FL methods on benchmark and medical image datasets, and evaluated and analyzed the experimental results.

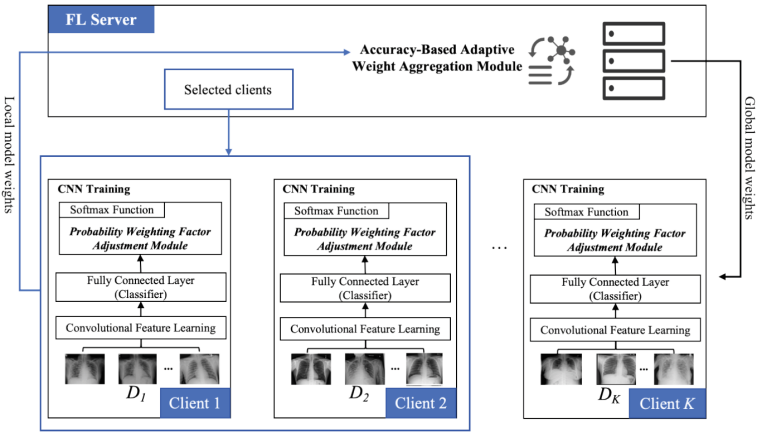

Fig. 1. An overview of Feddaw

2 Method

As shown in Fig. 1, the overall architecture of Feddaw consists of two components. First, on the client side, during local training, set a probability weighting factor in the softmax layer in the given training rounds, and train a more accurate model to the server to solve the problem of label distribution shift. Then, on the server-side, during the global model fusion process, validate the client models and dynamically adjust the global model aggregation weights for each client model by validating the accuracy obtained to resolve the data volume shift problem. The detailed structure of each module is described below.

2.1 Client-Side Classification Layer Probability Weighting Factor Adjustment Module

Consider a scenario where there are K medical institution clients. Each client possesses its own local dataset $D_i^k = \left(x_i^k, y_i^k\right)_{i=1}^{N^k}$, where x_i^k represents the input sample of the k-th client, $y_i^k \in \{1,2,\ldots,C\}$, C is the number of sample categories. The CNN contains a feature extractor $F_\theta(\cdot)$ and weights $w_{c\,c=1}^{k\,C}$ of the final classification layer. And use $h_i^k = F_\theta\left(x_i^k\right) \in \mathcal{R}^d$ as the extracted feature vector for the i-th sample, Then, the softmax function can be used to normalize each classification (i.e., $h_i^k w_c^{kT}$). For the classification weight w_c^k of the c^k-th category, the c^k-th category is considered as the positive feature, while the features of the remaining categories are considered as negative features. FL in medical scenarios for complex Non-IID datasets from different medical institutions, there is the problem of missing sample categories or rare samples for certain categories. Therefore, the sample category C can be partitioned into existence class $\mathcal{O}$ and the missing class $\mathcal{M}$, where $\mathcal{O} \cup \mathcal{M} = C$ and $\mathcal{O} \cap \mathcal{M} = \varnothing$, with $\varnothing$ as the empty set. In the label distribution shift FL scenario, the local dataset distribution of the k-th client can be expressed as follows:

$$\mathcal{D}^k = \mathcal{P}(x, y) \tag{1}$$

In the case of a label distribution shift, there may be differences in $\mathcal{P}^k(y)$ among clients, while $\mathcal{P}^k(x|y)$ may be similar. The goal of the FL algorithm is essentially to minimize the local loss, as Eq. (2):

$$min_{\ell=\left(\theta,\{w_c\}_{c=1}^C\right)} = \sum_{k=1}^{K} p_k \mathcal{L}^k\left(\ell, \mathcal{D}^k\right) \tag{2}$$

where p_k is the weight of client k, and $\sum_{k=1}^K p_k = 1$, $\mathcal{L}^k\left(\ell, \mathcal{D}^k\right)$ denotes the local loss. and assume that the label y_i^k is only from the existence class $\mathcal{O}^k$ of the k-th client, and the missing classes are denoted as $\mathcal{M}^k$, and the classification weight of the missing class is denoted as $w_{c\,c\in\mathcal{M}^k}^k$, Then, a certain percentage of scaling is performed in the softmax layer to limit the update, as Eq. (3):

$$p_{i,c}^k = \frac{exp\left(\alpha_c^k w_c^{kT} h_i^k\right)}{\sum_{c=1}^C exp\left(\alpha_c^k w_c^{kT} h_i^k\right)} \tag{3}$$

where α_c^k is set to Eq. (4), and $\alpha \in [0, 1]$ is a hyperparameter in the algorithm.

$$\alpha_c^k = \mathbb{1}\left\{c \in \mathcal{O}^k\right\} + \alpha \;\; \mathbb{1}\left\{c \in \mathcal{M}^k\right\} \tag{4}$$

For the existence class $\mathcal{O}^k$, $\alpha_c^k = 1$. For the missing class $\mathcal{M}^k$, $\alpha_c^k = \alpha$. The cross-entropy loss function is expressed as follows:

$$\mathcal{L}^k = -\sum_{i=1}^{N^k}\sum_{c=1}^{C} \mathbb{1}\left\{y_i^k = c\right\} log p_{i,c}^k \tag{5}$$

And the gradient of w_c^k can be represented as follows:

$$\frac{\partial \mathcal{L}^k}{\partial w_c^k} = -\alpha_c^k \sum_{i=1}^{N^k} \left(\mathbb{1}\left\{ y_i^k = c \right\} - p_{i,c}^k \right) h_i^k \tag{6}$$

Updating the classification weights w_c^k using a gradient descent algorithm with a learning rate of η. For the missing class $c \in \mathcal{M}^k$ where $\alpha_c^k = \alpha$, then w_c^k is updated as Eq. (7):

$$w_c^k = w_c^k - \alpha \cdot \eta \sum_{i=1}^{N^k} p_{i,c}^k h_i^k \tag{7}$$

when the sample category $c \in \mathcal{O}^k$, w_c^k is expressed as Eq. (8):

$$w_c^k = w_c^k + \underbrace{\eta \sum_{i=1, y_i = c}^{N^k} \left(1 - p_{i,c}^k\right) h_i}_{increment} - \underbrace{\eta \sum_{i=1, y_i \neq c}^{N^k} p_{i,c}^k h_i^k}_{existence\ class |\mathcal{O}|-1} \tag{8}$$

From Eq. (7), the update of w_c^k can be limited by adjusting the value of α. When α = 0, w_c^k remains unchanged; when α = 1, it is a standard softmax function. For the i-th sample, there is Eq. (9):

$$\alpha_{y_i^k} = 1, \alpha_{c \in \mathcal{O}^k} = 1, \alpha_{c \in \mathcal{M}} = \alpha \tag{9}$$

Then From Eq. (7), Eq. (8) and Eq. (9), The feature update process can be obtained as fllows:

$$h_i^k = h_i^k + \underbrace{\eta \left(1 - p_{i,y_i}^k\right) w_{i,y_i}^k}_{increment} - \underbrace{\eta \sum_{c \in \mathcal{O}^k, c \neq y_i^k}^{C} p_{i,y_i}^k w_{i,y_i}^k}_{existence\ class |\mathcal{O}|-1} - \underbrace{\alpha \cdot \eta \sum_{c \in \mathcal{M}^k}^{C} \alpha p_{i,c}^k w_c^k}_{missing\ class\ \mathcal{M}} \tag{10}$$

From Eq. (10), the update of the feature vector is decomposed into a superposition of one increment and two decrements. Compared with the standard softmax, without the limitation of the probability weighting factor α for the client classification layer, the effective decrement is only from the existence class $|\mathcal{O}| - 1$, and the decrement of the missing class is inaccurate or even zero values. For the error caused by the missing class, one solution is to directly discard the classification probability of the missing class $\mathcal{M}$. However, this solution is not suitable for the label distribution shift in medical scenarios. Because the datasets of each medical institution differ in the presence of missing classes, e.g., sample categories are missing or a very small amount of data for a sample category, directly discarding the classification probability of a client $\mathcal{M}$ will seriously affect the probability prediction of the global model for that disease category. So the client-side limit on the probability weighting factor of the classification layer solves this problem well.

2.2 Server-Side Accuracy-Based Adaptive Weight Aggregation Module

Based on the FedAvg method [13], This module uses the central server to verify the local models uploaded by clients, calculates the accuracy of the local models, and then dynamically adjusts the global model aggregation weights. The specific algorithm flow is as follows:

Step 1: In communication rounds t − 1, the server randomly selects $max(\rho \cdot K, 1)$ sets of clients C_t, where client $k \in C_t, k = 1,2, \ldots, \rho \cdot K$, and these selected clients download the global model parameters ω_{t-1}.

Step 2: The selected client k trains the global model ω_{t-1} using the local dataset and updates the global model parameters ω_{t-1} to its local model parameters ω_t.

Step 3: All selected clients upload the local model $\omega_t^k (k = 1,2, \ldots, \rho \cdot K)$ to the server after completing the local training.

Step 4: The server verifies the accuracy of the client local models ω_t^k respectively and calculates the local model accuracy a_t^k.

Step 5: Based on the accuracy a_t^k of the client local models, the server calculates the weights of each client according to Eq. (11).

$$p_k = \beta \frac{a_t^k}{\sum_{i=1}^{\rho \cdot K} a_t^i} \tag{11}$$

Step 6: After calculating the weights of all local clients, the server completes the global model aggregation according to Eq. (12) and calculates the global model parameter ω_t.

$$\omega_t = \sum_{k=1}^{\rho \cdot K} p_k \omega_{t-1}^k \tag{12}$$

Finally repeat Steps 1 to 5 until the global model has reached convergence.

Based on the two modules above, the detailed flow of the Feddaw method is shown in Algorithm 1.

3 Experiments and Results

3.1 Data Description

The Feddaw uses two benchmark datasets, CIFAR-10 [14] and CIFAR-100 [15], as well as two medical image datasets, COVIDx [16] and KVASIR [17].

CIFAR-10 is a small dataset containing ten categories of RGB color images, including 50,000 training images and 10,000 testing images.

CIFAR-100 dataset is similar to CIFAR-10, but contains 100 classes, each each containing 500 training images and 100 testing images.

COVIDx consists of six publicly available COVID-19 datasets and is the largest publicly available dataset with the highest number of COVID-19 positive cases. In this experiment, nearly 20,000 images from the following six datasets have been collected:

COVID-19 Image Data Collection: Developed and maintained by the National Center for Biotechnology Information (NCBI) of the National Institutes of Health (NIH), this dataset contains chest X-ray and CT scan images of multiple COVID-19 cases.

COVID-19 Chest X-ray Dataset Initiative: Collected by a group of developers from GitHub community based on public data, this dataset contains images of chest X-rays from different source, and includes COVID-19 cases.

Algorithm 1 Feddaw

INPUT: global update rounds T, client K, local training rounds E, proportion of participating training clients in each round ρ, learning rate η, model parameters ω_t^k for client k at round t, dataset D_k for client k, batch B

OUTPUT: global model parameter ω_T

Initialize global model parameter ω_0, broadcast ω_0 to all clients
Server aggregation:
  for Global model updating rounds t = 1, 2, …, T **do**
    $C_t \leftarrow \max(\rho \cdot K, 1)$ //determine the set of randomly selected clients
    for each client $k \in C_t$ parallelly **do**
      $\omega_{t+1}^k \leftarrow$ Client Update (k, ω_t)
    end for
  verify all local model accuracies a_t^k
  $p_k = \beta \frac{a_t^k}{\sum_{i=1}^{\rho \cdot K} a_t^i}$ //set client weights based on accuracy a_t^k
  $\omega_{t+1} \leftarrow \sum_{k=1}^{\rho \cdot K} p_k \omega_t^k$ //use the weight aggregation equation to obtain the new round of global model parameters
  Check whether the global model converged
    If converged, send a message to notify the client to stop model training
    If not converged, broadcast the aggregated model parameters ω_{t+1} to all clients
end for
Client Update (k, ω):
  batch $\beta \leftarrow$ randomly divide the dataset D_k into batches of size B
  for local training rounds $i = 1, 2, \ldots,$ E **do**
    for batch $b \in \beta$ **do**
      $p_{i,c}^k = \frac{\exp\left(\alpha_c^k w_c^{k^T} h_i^k\right)}{\sum_{j=1}^{C} \exp\left(\alpha_j^k w_j^{k^T} h_i^k\right)}$
      local model parameter$\omega \leftarrow \omega - \alpha \cdot \eta \sum_{i=1}^{N^k} p_{i,c}^k h_i^k$
    end for
  end for
  return ω to server

COVID-19 Chest X-ray Dataset Initiative: Collected by a group of developers from GitHub community based on public data, this dataset contains images of chest X-rays from different source, and includes COVID-19 cases.

Actualmed COVID-19 Chest X-ray Dataset: Created by Actualmed Health IT Solutions, this dataset contains chest X-ray images of multiple COVID-19 cases from different institutions and countries.

COVID-19 Radiography Dataset: Collected by a group of developers in the GitHub community based on public data, this dataset contains images of chest X-rays from a variety of sources, and includes images of COVID-19 cases.

RSNA Pneumonia Detection Challenges Dataset: Developed by the Radiological Society of North America (RSNA), this dataset contains chest x-ray images from multiple institutions, and includes pneumonia cases.

RSNA International COVID-19 Open Radiology Database: Developed by the Radiological Society of North America (RSNA), this dataset contains multiple COVID-19 chest CT scan images and other medical imaging images.

The KVASIR dataset is an eight-class dataset consisting of gastrointestinal disease images. It contains images of gastrointestinal anatomical landmarks, pathological findings, and gastrointestinal endoscopy procedures. There are 1,000 images for each class, for a total of 8,000 images in eight classes, of which 6,000 are used for training and 2,000 for testing. In total, the dataset contains 8,000 endoscopic images, with 1,000 image examples per class.

3.2 Non-IID Dataset Segmentation

To demonstrate the effectiveness of Feddaw on Non-IID datasets, this experiment redivided and reconstructed four datasets to simulate label distribution and sample volume shifts.

Taking the benchmark datasets as an example, CIFAR-10 and CIFAR-100 were divided into two IID datasets and four Non-IID datasets. The IID dataset divided the datasets into several subsets with uniform label distribution and equal sample volumes, such as C10-K100-M10, which divided CIFAR-10 into 100 subsets with uniform label distribution and equal sample volumes, and assigned to 100 clients.

And the Non-IID datasets were reconstructed as C10–100-M2, C10-K100-M5, C100-K100-M20 and C100-K100-M50. These four datasets have constructed scenarios of label distribution shift and data volume shift by partitioning labels and unevenly distributing data. Taking C10-K100-M5 as an example, the samples of each class in the CIFAR-10 dataset were divided into 50 parts, giving a total of $10 \times 50 = 500$ samples. At the same time, set up 100 clients, where 50 clients were divided into 7 samples and 50 clients were divided into 3 samples, ensuring both label distribution and data volume shift at the same time. COVIDx and KVASIR were divided in the same way to ensure the incompleteness of the label category and the variability of the data volume between clients.

3.3 Baselines and Implementation Details

Using FedAvg as a baseline in this experiment, the Feddaw is compared to other advanced FL methods to demonstrate the superiority of the method. The comparison is made as follows:

FedMMD [18] uses a two-stream model with a maximum mean deviation constraint instead of a single model for training on devices, reducing communication cycle by 20% on Non-IID datasets.

FedProx [19] is able to reduce the convergence time and is applicable to various different FL scenarios. It runs the SGD algorithm a variable number of times according to the system resources available on the client device, and uses data compression to update

the model to better account the differences in client data volume and computational resources.

FD [6] is based on federated distillation and federated augmentation, which reduces the size of the model output by allowing the client and server to exchange model outputs instead of exchanging model parameters. This helps to reduce the communication overhead.

FLDA [20] adapts the training and transport policy by dividing the universal and private models to reduce the impact of the Non-IID dataset. In high-noise environments, the accuracy is reduced by only 0.8%.

FedAwS [21] is a general framework that only uses positive labels for training and imposes a geometric regularizer after each round to encourage classes to unfold in the embedding space.

Scaffold [22] corrects for client shift in the local update process using methods such as variance reduction and control variables, thereby reducing the number of communication rounds without being affected by data heterogeneity or client sampling.

This experimental environment was run on a server with a CPU of Intel(R) Xeon(R) Silver 4114, 128GB of RAM, and four GPUs of Nvidia GeForce GTX 2080Ti 10GB. FL local training and FL algorithm sections were implemented using Python 3.6 and PyTorch 1.4.0 toolkit. Table 1 shows the model training parameter settings. And For all the methods, we use accuracy as the main evaluation metric to measure the correctness of the method, thus directly reflecting the performance of the method on the given task.

Table 1. Model training parameter settings

Parameters	Meaning	Parameter Value			
		CIFAR-10	CIFAR-100	COVIDx	KVASIR
T	global update rounds	1000	1000	100	200
E	local update rounds	5	5	3	3
η	learning rate	0.03	0.03	0.005	0.01
K	number of clients	100	100	60	80
ρ	client selection ratio	0.1	0.1	0.5	0.5
B	local training batch size	64	32	16	32
α	hyperparameter	0.50	0.50	0.60	0.50
β	hyperparameter	0.94	0.89	0.98	0.93

3.4 Performance Comparison with Baseline Methods

Performance Results and Analysis on Benchmark Datasets

According to Table 3–1, setting the model training parameters, and set the weight decay to 5e-4. Experiment on six reconstructed benchmark datasets of IID and Non-IID using

deep learning models VGG11 and CNN respectively, and select the average experimental results of the last 50 rounds.

Based on Table 2 and Table 3, on the C10-K100-M5 dataset and VGG11 model, Feddaw achieves the highest accuracy of 83.28%, significantly higher than other algorithms, with an average improvement of 1.46%. On the C10-K100-M5 dataset and CNN model, the performance of the other algorithms generally declines, but Feddaw achieves an accuracy of 77.49%, an average improvement of 5.38% over the other algorithms. Feddaw also performs better on the more complex data sets C100-K100-M20 and C100-K100-M50. On the C100-K100-M50 dataset, Feddaw achieves accuracies of 59.97% and 59.53% on the two models, which is an average improvement of 5.08% and 11.46% over the other algorithms. In summary, based on the four scenarios and two models in the benchmark dataset, the Feddaw has achieved an average accuracy improvement of 5.85%, and possesses better generalization ability.

Table 2. Experimental results of VGG11 on CIFAR10 and CIFAR100 datasets

Method	CIFAR10			C10-K100-M10		
	C10-K100-M10	C10-K100-M5	C10-K100-M2	C100-K100-M100	C100-K100-M50	C100-K100-M20
FedAvg	91.13	82.04	69.64	72.23	58.44	40.64
FedMMD	-	82.67	70.66	-	59.73	41.26
FedProx	-	82.79	68.48	-	57.89	40.74
FD	-	82.93	72.28	-	54.34	41.68
FLDA	-	76.84	57.74	-	49.87	37.16
FedAwS	-	82.85	71.52	-	54.32	43.84
Scaffold	-	82.62	71.81	-	56.99	41.36
Feddaw	-	**83.28**	**73.96**	-	**59.97**	**45.28**

Performance Results and Analysis in Medical Image Datasets

For the experiments on medical image datasets, repeated validations were performed during the experiments using different random seeds, comparing Feddaw with centralised training and the FedAvg method. Table 4 shows the global model accuracy of the three methods on the COVIDx and KVASIR datasets.

Experimental results on the COVIDx dataset show that the models using FedAvg and Feddaw have slightly lower accuracy than the centrally trained model. Of the four deep learning models, Feddaw achieves higher accuracy than FedAvg in three of the four deep learning models, with the highest accuracy improvement of 0.53%, and is closer to the centrally trained results.

Table 3. Experimental results of CNN on CIFAR10 and CIFAR100 datasets

Method	CIFAR10			C10-K100-M10		
	C10-K100-M10	C10-K100-M5	C10-K100-M2	C100-K100-M100	C100-K100-M50	C100-K100-M20
FedAvg	90.55	71.74	55.87	71.86	57.69	40.80
FedMMD	-	72.36	54.69	-	58.77	41.40
FedProx	-	71.53	53.43	-	56.59	41.10
FD	-	73.49	64.14	-	54.38	42.40
FLDA	-	68.13	58.95	-	50.12	38.10
FedAwS	-	75.24	62.72	-	53.84	44.20
Scaffold	-	72.27	55.89	-	55.67	42.50
Feddaw	-	**77.49**	**69.42**	-	**59.53**	**45.50**

Experimental results on the KVASIR dataset show that using the Feddaw achieves the best accuracy in both models, with an average improvement of 2.05% compared to FedAvg. This further demonstrates the superiority and generalization capabilities. of the Feddaw method.

Table 4. Global model accuracy on COVIDx dataset and KVASIR dataset

Method	COVIDx				KVASIR	
	Covid-Net	MobileNetsV2	ResNet50	ResNeXt	MobileNetsV2	ResNet50
Centralized Training	89.44	88.22	91.34	91.12	88.22	91.34
FedAvg	87.61	86.98	90.86	90.26	90.75	87.53
Feddaw	**88.14**	**87.01**	**90.84**	**90.31**	**92.98**	**91.45**

4 Conclusion

To address the problem of extreme dataset imbalance in medical scenarios, a novel FL method called Feddaw is presented in this paper. Which aims to improve the performance of the global FL model when dealing with datasets from different medical institutions. First, by limiting the scaling operation of the probability weighting factors in the classification layer during local training at the client to mitigate the negative impact of label distribution shift. Second, by verifying the accuracy of the local model at the server side to adjust the weight proportion towards clients with higher sample quality in the global model aggregation process. The Feddaw can be widely used in various disease diagnosis classification tasks and significantly improves the overall performance compared to other FL methods.

Acknowledgments. This work was supported in part by the National Key Research and Development Program of China (2021YFF1201300, 2023YFC3604600), the Hunan Provincial Natural Science Foundation of China (2022JJ30747).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Konecný, J., Mcmahan, H.B., et al.: Federated learning: strategies for improving communication efficiency. arXiv preprint arXiv:1610.05492 (2016)
2. Zhu, H., Xu, J., Liu, S., Jin, Y.: Federated learning on non-IID data: a survey. J. Neurocomputing. **465**, 371–390 (2021)
3. Xu, J., Glicksberg, B.S., Su, C., Walker, P., Bian, J., Wang, F.: Federated learning for healthcare informatics. J. Healthc. Inform. Res. **5**(1), 1–19 (2020)
4. Xiao, J., Du, C., Duan, Z., Guo, W.: A novel server-side aggregation strategy for federated learning in non-IID situations. In: 2021 20th International Symposium on Parallel and Distributed Computing (ISPDC), pp. 17–24. IEEE (2021)
5. Li, A., Sun, J., Wang, B., Duan, L., Li, H.: LotteryFL: personalized and communication-efficient federated learning with lottery ticket hypothesis on non-IID datasets. arXiv preprint arXiv:2008.03371 (2020)
6. Jeong, E., Oh, S., Kim, H., et al.: Communication-efficient on-device machine learning: federated distillation and augmentation under non-IID private data. arXiv preprint arXiv:1811.11479 (2018)
7. Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. arXiv preprint arXiv:1812.06127 (2018)
8. Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., Chandra, V.: Federated learning with non-IID data. arXiv preprint arXiv:1806.00582 (2018)
9. Wang, H., Kaplan, Z., Niu, D., Li, B.: Optimizing federated learning on non-IID data with reinforcement learning. In: IEEE INFOCOM 2020 - IEEE Conference on Computer Communications, pp. 1698–1707. IEEE (2020)
10. Kopparapu, K., Lin, E., Zhao, J.: FedCD: improving performance in non-IID federated learning. arXiv preprint arXiv:2006.09637 (2020)
11. Nagendran, M., Chen, Y., Lovejoy, C.A., Gordon, A.C., Komorowski, M., Harvey, H., et al: Artificial intelligence versus clinicians: systematic review of design, reporting standards, and claims of deep learning studies. BMJ, **368**, m689 (2020)
12. Li, X., Zhan, D.: FedRS: federated learning with restricted softmax for label distribution non-IID data. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 995–1005. (2021)
13. Mcmahan, H.B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.Y.: Federated learning of deep networks using model averaging. arXiv preprint arXiv:1602.05629 (2016)
14. Krizhevsky, A.: Convolutional deep belief networks on CIFAR-10. Unpublished Manuscript. **40**(7), 1–9 (2010)
15. Sharma, N., Jain, V., Mishra, A.: An analysis of convolutional neural networks for image classification. J. Procedia Comput. Sci. **132**, 377–384 (2018)
16. Wang, L., Lin, Z.Q., Wong, A.: COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images. Sci. Rep. **10**(1), 19549 (2020)

17. Pogorelov, K., Randel, K.R., Griwodz, C., et al.: KVASIR: a multi-class image dataset for computer aided gastrointestinal disease detection. In: Proceedings of the 8th ACM on Multimedia Systems Conference, pp. 164–169 (2017)
18. Yao, X., Huang, C., Sun, L.: Two-stream federated learning: reduce the communication costs. In: 2018 IEEE Visual Communications and Image Processing (VCIP), pp. 1–4. IEEE (2018)
19. Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M.: Federated optimization in heterogeneous networks. J. Proc. Mach. Learn. Syst. **2**, 429–450 (2020)
20. Peterson, D., Kanani, P., Marathe, V.J.: Private federated learning with domain adaptation. arXiv preprint arXiv:1912.06733 (2019)
21. Yu, F.X., Rawat, A.S., Menon, A., et al.: Federated learning with only positive labels. In: International Conference on Machine Learning, pp. 10946–10956. PMLR (2020)
22. Karimireddy, S.P., Kale, S., Mohri, M., et al.: Scaffold: stochastic controlled averaging for federated learning. In: International Conference on Machine Learning, pp. 5132–5143. PMLR (2020)

LoopNetica: Predicting Chromatin Loops Using Convolutional Neural Networks and Attention Mechanisms

Yang Lei, Li Tang, HanYu Luo, WenJie Huang, and Min Li(✉)

Hunan Provincial Key Lab on Bioinformatics, School of Computer Science and Engineering, Central South University, Changsha 410083, China
limin@mail.csu.edu.cn

Abstract. Within the cell nucleus, chromatin folds to form loop structures that bring distant genomic regions into close proximity, a foundational mechanism in gene regulation. These loop structures facilitate interactions among enhancers, promoters, and other regulatory elements, fundamentally influencing gene expression patterns. With the advent of high-throughput technologies such as Hi-C and ChIA-PET, researchers have begun to peel back the layers of the genome's complex three-dimensional organization, identifying thousands of looping interactions that vary across cell types and are conserved across species. However, these experimental techniques often require extremely high costs and complex experimental workflows, and their resolution is often low, while existing computational methods do not take into account the extremely imbalanced challenges of samples and often require additional epigenetic data, which are not always available. To overcome this problem, we propose a new deep learning computational tool called LoopNetica by utilizing a combination of one-dimensional convolutional neural networks and a multi-head attention mechanism. It can accurately predict the formation of chromatin loops using only sequences. Its accuracy is higher than existing methods, and LoopNetica can still maintain its accuracy even when the sample distribution is extremely imbalanced. With a simple and exquisite architecture, LoopNetica has high performance and very fast training speed. LoopNetica not only marks a major leap in computational exploration of genome structure, but also lays the foundation for a deeper understanding of the regulatory environment that drives gene expression and disease.

Keywords: Chromatin loops · Convolutional neural network · Multi head attention · DNA sequences · Deep Learning

1 Introduction

After extensive research into the correlation between genes, gene expression, and disease mechanisms, biologists have discovered that gene expression and transcriptional regulation are influenced not only by the arrangement of bases but also by the three-dimensional structure of genes [1]. For example, the formation of chromatin loops in mammals can enhance interactions within chromatin,

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2024
W. Peng et al. (Eds.): ISBRA 2024, LNBI 14956, pp. 14–27, 2024.
https://doi.org/10.1007/978-981-97-5087-0_2

affecting gene regulation [2], while changes in the three-dimensional structure of the genome in plants can affect the level and pattern of gene expression, thereby regulating plant growth, stress responses, and phenotypic diversity [3]. Whether in animals or plants, the three-dimensional structure of chromatin plays a crucial role in genome regulation and gene expression. Chromatin loops are an important structure within the three-dimensional chromatin structure. Scientists have proposed many models for the formation of chromatin loops, with the loop extrusion model introduced by Nasmyth being widely accepted [4]. This model suggests that loop extrusion is a motor-driven process, where protein complexes attached to chromatin fibers gradually squeeze the chromatin from both sides to form loops. The regions where interactions form loops are called "anchors," and strong interactions between two specific anchors create loops. These loops, mediated by protein complexes, are known as chromatin loops. Many anchors overlap with gene regulatory elements, with promoters and enhancers making up the majority. Chromatin loops formed in this way can bring two genomic sites that are far apart into close spatial proximity and interact, thereby regulating genes. Rao and others have found that there are about ten thousand loop structures in the human and mouse genome, which usually link promoters and enhancers, are related to gene activation, and are conserved across different cell types and species. In mammalian cells, abnormalities in chromatin loop structures can cause changes in gene expression and regulation, leading to disease. Scientists from institutions such as Northwestern University in the United States have discovered that patients with acute myeloid leukemia carry large-scale genomic changes, which may be related to specific DNA folding patterns or chromatin loops in leukemia cells, and these novel chromatin loops could serve as potential therapeutic targets for acute myeloid leukemia [5].

In recent years, high-throughput experimental techniques such as High-throughput chromosome conformation capture (Hi-C) [6] and Chromatin Interaction Analysis by Paired-End Tag (ChIA-PET) [7] have been developed to detect whole-genome chromatin interactions. Advancements in three-dimensional genomics technologies such as Hi-C, ChIP-seq, and ChIA-PET have significantly improved our understanding of chromatin interactions and their role in transcriptional regulation. These technologies provide insights at various scales: Hi-C captures all chromatin interactions across the genome, providing a broad structural overview, whereas ChIA-PET offers more detailed data on specific protein-DNA interactions and chromatin loops, yielding higher resolution insights. However, these methods are expensive and technically demanding, requiring sophisticated equipment and complex analysis techniques. Developing methods to predict chromatin's three-dimensional structure from DNA sequences could greatly simplify this process, enhancing our understanding of chromatin organization and its relationship to diseases. Methods like DeepSEA [8], DeepMILO [9] demonstrate that predicting many transcription factor binding sites, including CTCF, directly from gene sequences is feasible. Akita [10], Orca [11] use gene sequences combined with epigenetic information to predict chromatin interaction maps, achieving very good results. However, most methods to date

rely on using functional genomics data, including chromatin immunoprecipitation sequencing (ChIP-seq) [12], ATAC-seq [13], DNase-seq [14], and gene expression data, for joint predictions. If it is difficult to complete so many sequencing tasks in a short time and understand the changes in chromatin structure in clinical patients, and the cost is high, in addition, many samples lack these functional genomics data. Therefore, being able to infer the three-dimensional structure of chromatin with as little experimental data as possible has high practical application value and theoretical significance for in-depth studies on the regulatory mechanisms of the genome, the association between disease and the three-dimensional genome, and other related biological research. ChINN is a method for predicting chromatin loops using gene sequences from open chromatin regions, capturing pattern information in the sequence well and achieving good results with distance features in prediction, but its use of only convolutional networks and integrated models means the model has low interpretability.

To overcome these challenges, we propose an efficient model with an attention mechanism, LoopNetica, to accurately predict chromatin loops using only DNA sequences. And the model can handle the situation of extremely imbalanced sample distribution. In addition, LoopNetica can help find cell type-specific motifs, which is of great significance for studying the relationship between sequence structure and cell type. It is also a major advancement in gene structure calculation methods.

2 Results

2.1 LoopNetica: Effectively Combines Convolutional Neural Networks and Attention Mechanisms

The LoopNetica model is designed to predict chromatin loops from DNA sequences. The model uses a combination of one-dimensional convolutional neural networks (1D CNN) and a multi-head attention mechanism to analyze DNA sequences. The convolution part consists of a total of three convolution layers, which are in-depth layer by layer, with a maximum pooling layer in the middle of each two layers. The purpose of the convolution layer is to gradually extract local features until all local features at multiple scales are extracted. After the convolutional network, we added an attention mechanism, which can extract the global features of the sequence on top of the convolutional features, and its performance and effect are better than the long short-term memory neural network (LSTM). This combination has proven to be very effective in natural language processing. Our model diagram is shown in the Fig. 1.

Our method is able to effectively learn spatial hierarchies from sequence data, leveraging convolutional layers for pattern recognition and attention mechanisms to understand sequence interactions. The combination of these technologies enables LoopNetica to accurately predict the presence of chromatin loops, making significant advances in the field of genomics through detailed analysis of chromatin organization based solely on sequence data.

These findings highlight LoopNetica's capabilities as a genome analysis tool, particularly in identifying chromatin loops, and demonstrate its potential use in further understanding genome structure-function relationships and their impact on gene regulation and disease.

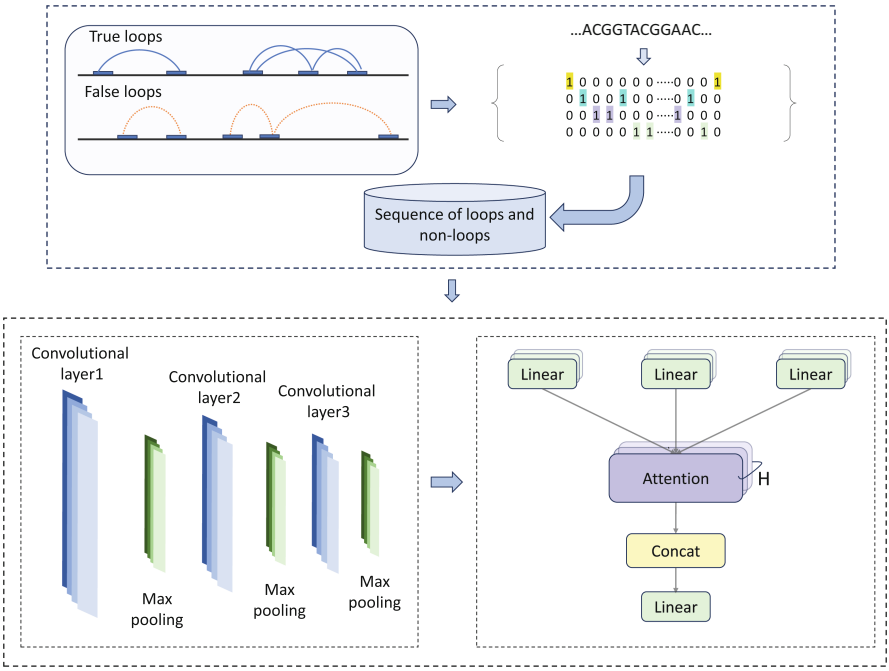

Fig. 1. Overview of the inputs and architecture of the LoopNetica model

2.2 LoopNetica Can Accurately Predict Chromatin Loops

The positive samples in our experiments were derived from the chia-pet files of the ENCODE project, involving cell lines such as GM12878, HCT116, H1 and WCT11. The model training uses a positive and negative sample ratio of 1:5 to examine the performance of the model in the case of sample imbalance. First, through a carefully designed training strategy, we observed a clear downward trend in the model's loss curve during the training process (Fig. 2), indicating that significant progress in learning has been made and the model's ability to fit the training data has gradually improved. In addition, we also use AUC and AUPR, two important performance evaluation indicators widely recognized in the field of machine learning, to comprehensively reflect the performance of the classification model. The AUC value reflects the overall efficiency of the model in classifying positive and negative samples, while AUPR focuses more on the accuracy of positive sample prediction. For tasks with an imbalance of

positive and negative samples, AUPR can effectively evaluate the classification ability of the model. LoopNetica shows excellent performance on both indicators, fully demonstrating its powerful classification ability under imbalanced sample conditions. We also compared LoopoNetica with CHINN, Logistic regression (LR) and Random Forest (RF), showing that our model has better classification ability with this sample ratio (Table 1).

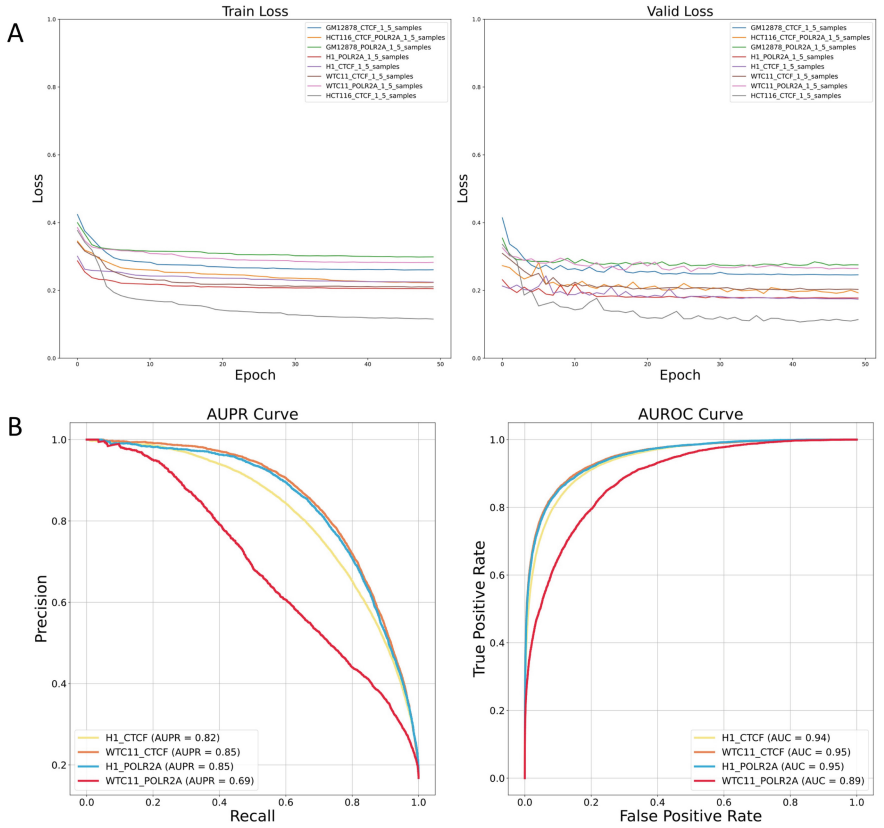

Fig. 2. Loss reduction and performance graph of the model **A.** Training (left) set and validation (right) set loss reduction plots on each cell line. **B.** auprc (left) and auc (right) indicators on various cell lines

Table 1. Comparison with other models on various metrics

sample	model	acc	auc	aupr	precision	recall	f1
HCT116 CTCF_POLR2A	LoopNetica	0.94	0.95	0.88	0.93	0.67	0.78
	ChINN	0.93	0.95	0.87	0.94	0.61	0.73
	LR	0.83	0.78	0.56	0.75	0.26	0.38
	RF	0.79	0.75	0.44	0.00	0.00	0.00
GM12878 CTCF	LoopNetica	0.9	0.92	0.77	0.81	0.53	0.64
	ChINN	0.88	0.90	0.70	0.78	0.40	0.53
	LR	0.81	0.60	0.23	0.28	0.04	0.08
	RF	0.82	0.63	0.26	0.00	0.00	0.00
GM12878 POLR2A	LoopNetica	0.88	0.89	0.67	0.79	0.37	0.51
	ChINN	0.86	0.87	0.66	0.97	0.29	0.45
	LR	0.84	0.69	0.35	0.41	0.08	0.14
	RF	0.85	0.67	0.26	0.00	0.00	0.00
H1 CTCF	LoopNetica	0.91	0.94	0.82	0.84	0.6	0.70
	ChINN	0.91	0.94	0.80	0.87	0.50	0.64
	LR	0.84	0.79	0.50	0.52	0.41	0.46
	RF	0.84	0.78	0.47	0.00	0.00	0.00
H1 POLR2A	LoopNetica	0.92	0.95	0.85	0.88	0.64	0.74
	ChINN	0.91	0.94	0.85	0.93	0.53	0.67
	LR	0.89	0.91	0.73	0.72	0.58	0.64
	RF	0.83	0.84	0.56	0.00	0.00	0.00
WTC11 CTCF	LoopNetica	0.92	0.95	0.85	0.89	0.60	0.72
	ChINN	0.91	0.95	0.82	0.87	0.54	0.66
	LR	0.87	0.89	0.72	0.77	0.50	0.61
	RF	0.80	0.83	0.56	0.00	0.00	0.00
WTC11 POLR2A	LoopNetica	0.88	0.89	0.69	0.83	0.36	0.50
	ChINN	0.86	0.89	0.68	0.83	0.35	0.49
	LR	0.81	0.80	0.43	0.40	0.05	0.09
	RF	0.81	0.71	0.36	0.00	0.00	0.00

2.3 The LoopNetica Model Performs Exceptionally Well in Scenarios with Extremely Imbalanced Positive and Negative Samples

In further research, we conducted an analysis of the proportion of loop regions to the total length of the DNA chain (Fig. 3A). This analysis step is crucial for evaluating the feasibility and effectiveness of the model in real-world applications. Due to the sparse distribution of loop regions on the DNA chain, identifying these regions becomes particularly challenging, a difficulty exacerbated

Table 2. Calculation of different evaluation metrics.

sample	ratio	acc	auc	aupr	precision	recall	f1
HCT116 CTCF POLR2A	1:5	0.94	0.95	0.88	0.93	0.67	0.78
	1:10	0.96	0.95	0.83	0.87	0.67	0.76
	1:15	0.97	0.95	0.79	0.80	0.67	0.73
	1:20	0.97	0.95	0.76	0.77	0.67	0.72
GM12878 CTCF	1:5	0.90	0.92	0.77	0.81	0.53	0.64
	1:10	0.94	0.92	0.68	0.69	0.53	0.60
	1:15	0.95	0.92	0.62	0.60	0.53	0.56
	1:20	0.95	0.92	0.58	0.52	0.53	0.53
GM12878 POLR2A	1:5	0.88	0.89	0.67	0.79	0.37	0.51
	1:10	0.92	0.89	0.54	0.64	0.37	0.47
	1:15	0.94	0.89	0.47	0.56	0.37	0.45
	1:20	0.95	0.89	0.41	0.48	0.37	0.42
H1 CTCF	1:5	0.91	0.94	0.82	0.84	0.60	0.70
	1:10	0.94	0.94	0.73	0.72	0.60	0.66
	1:15	0.95	0.94	0.68	0.63	0.60	0.62
	1:20	0.96	0.94	0.63	0.56	0.60	0.58
H1 POLR2A	1:5	0.92	0.95	0.85	0.88	0.64	0.74
	1:10	0.95	0.95	0.78	0.79	0.64	0.70
	1:15	0.96	0.95	0.74	0.71	0.64	0.67
	1:20	0.97	0.95	0.70	0.66	0.64	0.65
WTC11 CTCF	1:5	0.92	0.95	0.85	0.89	0.60	0.72
	1:10	0.95	0.95	0.78	0.82	0.60	0.70
	1:15	0.96	0.95	0.73	0.74	0.60	0.67
	1:20	0.97	0.95	0.69	0.68	0.60	0.64
WTC11 POLR2A	1:5	0.88	0.89	0.69	0.83	0.36	0.50
	1:10	0.93	0.89	0.57	0.71	0.36	0.48
	1:15	0.95	0.89	0.51	0.63	0.36	0.46
	1:20	0.96	0.89	0.46	0.57	0.36	0.44

when faced with extreme imbalances in positive and negative sample ratios. We demonstrated the proportions of loop regions on different chromosomes in various cell lines. Through in-depth analysis of the sparsity of loop regions, we gained a more comprehensive understanding of the impact of sample imbalance on model performance and emphasized the importance of considering sample imbalance during model design and evaluation (Table 2).

To comprehensively assess the robustness of the LoopNetica model, we not only trained and tested it under a 1:5 ratio of positive to negative samples but

also extended the evaluation to ratios of 1:10, 1:15, and 1:20, using the same model parameters for a series of performance assessments. By plotting AUPR curves under these extreme imbalanced conditions (Fig. 3B), we found that the LoopNetica model could largely maintain its accuracy, demonstrating a high degree of adaptability to different sample ratios. This result not only validates the potential application of the LoopNetica model in various sample imbalance scenarios but also showcases its significant value as an advanced sequence analysis tool in genomic research.

2.4 LoopNetica Successfully Captures Sequence Features and Discovers Type-Specific Motifs

We analyzed the outputs of the convolutional layers and the first fully connected layer in LoopNetica, ranking their outputs by variance and selecting those neurons with higher variance rankings. We believe that the outputs of these high-variance neurons can play a crucial role in classification. We visualized these

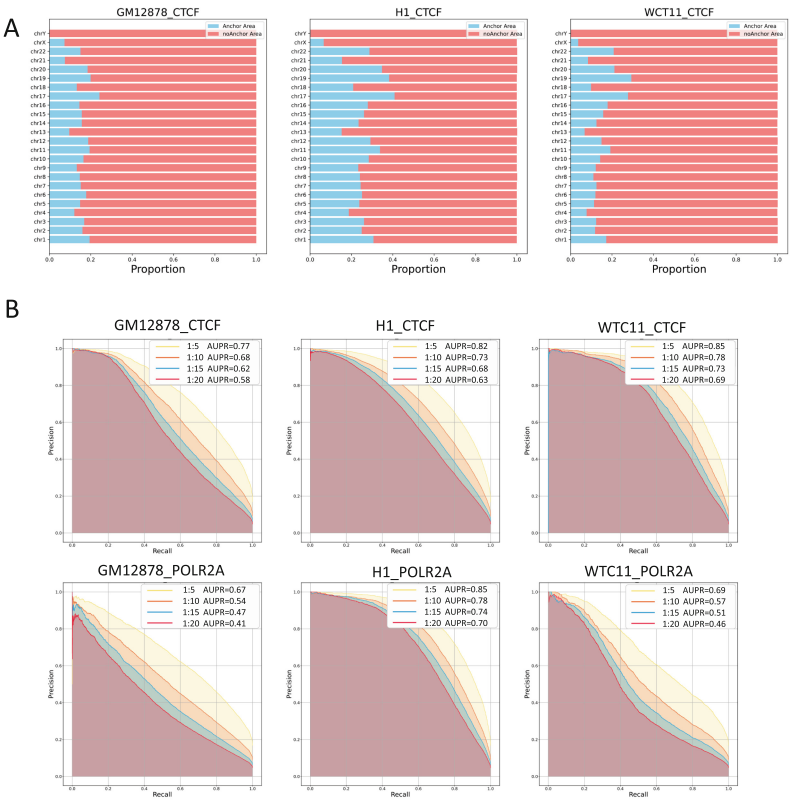

Fig. 3. The proportion of loop area and the aupr diagram of the model at each ratio **A.** The proportion of chromatin loop regions in the three data sets (left GM12878 CTCF middle H1 CTCF right WCT11 CTCF) **B.** The aupr of each data set when the positive and negative sample ratios are 1:5, 1:10, 1:15, and 1:20

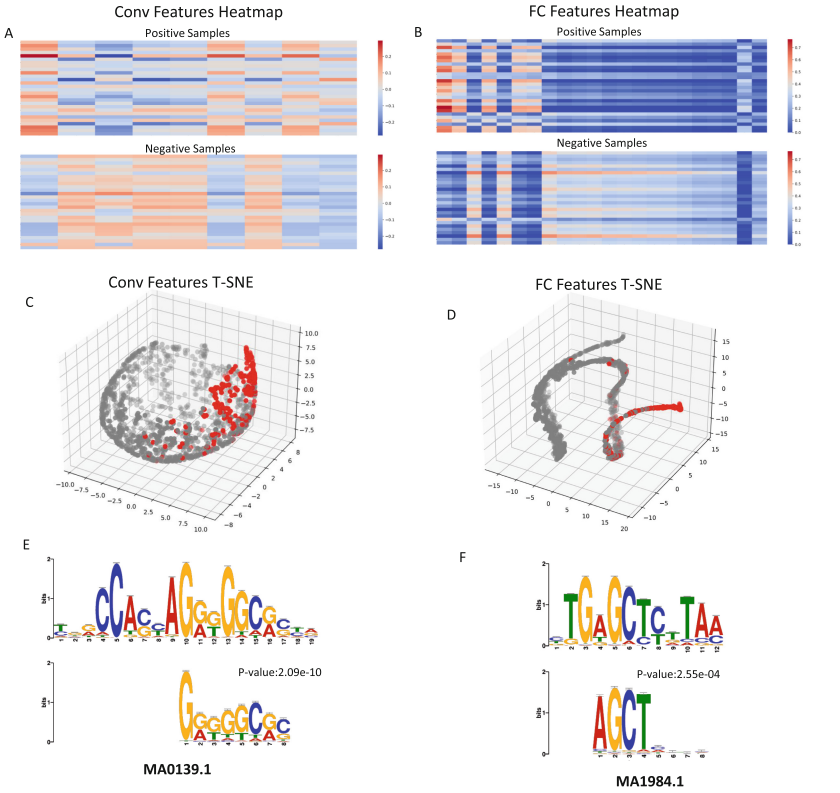

Fig. 4. Feature visualization and motif comparison of model extraction **A.** The output heat map of the convolutional layer (left) and the output heat map of the first fully connected layer (right). **B.** The output t-sne dimensionality reduction visualization of the convolutional layer (left) and the output t-sne dimensionality reduction visualization of the first fully connected layer (right). **C.** The convolution kernel highly similar to the CTCF motif in the GM12878 CTCF data set (left), and the convolution kernel highly similar to the zinc finger protein ZNF667 in the H1 CTCF data set (right)

outputs separately (Fig. 4A, B), and from the heat maps, we can clearly see the overall difference between positive and negative samples, especially after passing through the first fully connected layer, where this difference becomes more pronounced. In addition, we further applied T-SNE [15] dimensionality reduction to these outputs (Fig. 4C, D), after which we observed clear distinctions between positive and negative samples based on these output values. This indicates that LoopNetica successfully extracts features from sequences sufficient for effective classification.

We analyzed LoopNetica's first convolutional layer by normalizing its convolution kernel weights with Softmax and converting these to Position Frequency Matrices (PFMs) to visualize DNA sequence patterns. By comparing these PFMs

against a motif database using the TomTom tool, we identified significant similarities. Specifically, kernels from the GM12878 CTCF-trained model closely matched the CTCF motif, essential for chromatin loop formation (Fig. 4E). Similarly, kernels from the H1 cell line model showed alignment with various zinc finger proteins (Fig. 4F), as noted by Victor V and colleagues [16]. This indicates that chromatin loop formation may vary across cell lines, influenced by distinct transcription proteins, which the model utilizes to predict loop formation.

3 Methods

3.1 Data Preparation

We used chromatin loop data mediated by CTCF/POLR2A proteins from GM12878, HCT116, H1_CTCF, and WTC11 cell lines to train and test our model. These loops were captured by CTCF ChIA-PET and POLR2A ChIA-PET experiments, sourced from ENCODE and NCBI. The following steps were applied to preprocess the loop data: 1. The data underwent initial screening based on PET Counts, followed by duplicate removal. 2. The two anchor points of the chromatin loops were trimmed to a length of 2000 bases centered around the peak. 3. The sequences were converted into one-hot encoded matrices of size [2000x4].

3.2 LoopNetica Model

Train and Testing Datasets in LoopNetica. We defined positive and negative samples for the LoopNetica model's learning process. Positive samples were generated based on bedpe files derived from ChIA-PET experiments, specifically by trimming each loop's anchor points to a length of 2000 base pairs centered around the peak. Negative samples were randomly selected from the genome by pairing together two sequences of 2000 base pairs each, called pseudo-anchors, and we ensured that the sequences all belonged to non-coding regions, and we screened them using CTCF chip-seq data to avoid false-negative samples. We ensured that the distance between these pseudo-anchors did not exceed the maximum distance between anchors in positive samples, and that pseudo-anchors did not overlap with anchors in positive samples.

In the model, we utilized one-dimensional convolutional neural networks and multi-head attention mechanisms. This architecture enables the model to capture both local features within the sequence and global features with the assistance of multi-head attention mechanisms, facilitating classification. The combination of these two structures has been widely used in the field of natural language processing and has achieved outstanding results [17,18].

One-Dimensional Convolution Operation. The one-dimensional convolution operation involves a convolution kernel or filter that slides over the input

data to extract local features. For one-dimensional data x and a convolution kernel w of size k, the convolution operation can be represented as:

$$y(t) = (w * x)(t) = \sum_{s=-\infty}^{\infty} x(s) \cdot w(t-s) \tag{1}$$

In practical applications, both the input data and the convolution kernel are finite, so the above formula can be simplified to:

$$y(t) = \sum_{s=0}^{k-1} x(t+s) \cdot w(s) \tag{2}$$

Where t represents the current position of the convolution operation on the input data, s is the element index within the convolution kernel, and $y(t)$ is the convolution output.

Multi-head Attention Mechanism. The multi-head attention mechanism is a key feature in natural language processing (NLP), particularly in Transformer models. Introduced by Vaswani et al. in their 2017 paper "Attention is All You Need" [19], this technique enhances sequence-to-sequence models, useful in tasks like translation and managing long-distance dependencies in text.

This mechanism splits the traditional attention into multiple "heads," each learning distinct facets of the input sequence through components like queries, keys, and values, obtained via learned weights. By calculating attention weights through a dot product between queries and keys and normalizing with a softmax function, each head modifies the values to produce outputs. The final output is a combination of all heads' outputs through a linear layer, enhancing the model's ability to process complex dependencies from different perspectives.

The mathematical representation is as follows:

$$Attention(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{3}$$

Where Q, K, and V are matrices representing queries, keys, and values respectively, and d_k is the dimensionality of the keys, used to scale the dot product to prevent excessively large values that may lead to small gradients in the softmax function. The multi-head attention mechanism performs the aforementioned operations in parallel with h different heads, where each head learns different representations, thereby increasing the model's expressive power:

$$\text{MultilHead}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_\text{h})W^o \tag{4}$$

Where each $head\_i$ is calculated as follows:

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{5}$$

where W_i^Q, W_i^K, W_i^V and W^O are parameter matrices learned by the model.

3.3 Training Strategy

The model's data was divided into training, validation, and testing sets. Both the validation and testing sets adopted a 1:5 ratio of positive to negative samples, with the testing set further containing various ratios such as 1:5, 1:10, 1:15, and 1:20. Specifically, data from chromosome 11 was used as the validation set, while data from chromosomes 12 and 13 were chosen for the testing set, and data from other chromosomes were used for training. Through this design, we ensured that the model could learn and predict the probability of loop formation across a wide range of genomic regions. During model training, we employed the Adam optimization algorithm with an initial learning rate of 0.001. To address potential overfitting during training, we introduced a patience-based learning rate scheduling strategy, where the learning rate was reduced by a factor of 0.5 whenever the performance on the validation set did not improve for two consecutive iterations, until it reached a minimum value of 2e−9. Additionally, we used binary cross-entropy (BCE) as the loss function to accurately measure the difference between the model's predicted values and the actual labels.

$$loss(X_i, y_i) = -w_i[y_i log x_i + (1 - y_i) log(1 - x_i)] \tag{6}$$

The entire model training process consists of 50 to 100 training epochs, ensuring sufficient learning and parameter adjustment.

4 Discussion

The LoopNetica model, leveraging one-dimensional convolutional neural networks and multi-head attention mechanisms, effectively processes imbalanced genomic datasets, showing superior performance in predicting chromatin loop formation compared to traditional ChIA-PET and Hi-C methods. This makes it a cost-effective option for resource-limited research groups. Despite its strengths, the model's generalizability and data quality dependence need enhancement. Future developments should focus on expanding its applications and refining its architecture to improve accuracy and interpretability, underscoring deep learning's potential in advancing genomic and transcriptomic research, with significant implications for understanding gene regulation and aiding therapeutic discoveries.

5 Conclusion

In this study, we introduced the LoopNetica model, a computational tool that utilizes one-dimensional convolutional neural networks and multi-head attention mechanisms to predict chromatin loop formations from genomic data. Demonstrating high accuracy across various cell lines such as GM12878 and HCT116, the model marks a significant advancement in analyzing genomic structures. However, it requires further validation to confirm its generalizability and is influenced by data quality. Looking forward, LoopNetica's potential applications in

gene regulation, disease mechanism exploration, and therapeutic target identification are promising, offering profound contributions to computational biology and insights into genomic structure-function relationships.

Acknowledge. We are grateful to the High-Performance Computing Center of Central South University for partial support of this work.

Funding Information. National Natural Science Foundation of China [62320106009, 62225209 to M.L.]; Funding for open access charge: National Natural Science Foundation of China [62320106009]. Postdoctoral Fellowship Program of CPSF [GZC20233161 to L.T.]

References

1. Dekker, J., Marti-Renom, M.A., Mirny, L.A.: Exploring the three-dimensional organization of genomes: interpreting chromatin interaction data. Nat. Rev. Genet. **14**(6), 390–403 (2013). https://doi.org/10.1038/nrg3454
2. Oudelaar, A.M., Higgs, D.R.: The relationship between genome structure and function. Nat. Rev. Genet. **22**(3), 154–168 (2021). https://doi.org/10.1038/s41576-020-00303-x
3. Pei, L., Li, G., Lindsey, K., Zhang, X., Wang, M.: Plant 3D genomics: the exploration and application of chromatin organization. New Phytol. **230**(5), 1772–1786 (2021)
4. Nasmyth, K.: Disseminating the genome: joining, resolving, and separating sister chromatids during mitosis and meiosis. Ann. Rev. Genet. **35**, 673–745 (2001)
5. LaFleur, T.L., Hossain, A., Salis, H.M.: Automated model-predictive design of synthetic promoters to control transcriptional profiles in bacteria. Nat. Commun. **13**(1), 5159 (2022). https://doi.org/10.1038/s41467-022-32829-5
6. Lieberman-Aiden, E., et al.: Comprehensive mapping of long-range interactions reveals folding principles of the human genome. Science **326**(5950), 289–293 (2009)
7. Fullwood, M.J., et al.: An oestrogen-receptor-α-bound human chromatin interactome. Nature **462**(7269), 58–64 (2009). https://doi.org/10.1038/nature08497
8. Zhou, J., Troyanskaya, O.G.: Predicting effects of noncoding variants with deep learning-based sequence model. Nat. Meth. **12**(10), 931–934 (2015)
9. Trieu, T., Martinez-Fundichely, A., Khurana, E.: DeepMILO: a deep learning approach to predict the impact of non-coding sequence variants on 3D chromatin structure. Genome **21**(1), 79 (2020). https://doi.org/10.1186/s13059-020-01987-4
10. Fudenberg, G., Kelley, D.R., Pollard, K.S.: Predicting 3D genome folding from DNA sequence with Akita. Nat. Meth. **17**(11), 1111–1117 (2020). https://doi.org/10.1038/s41592-020-0958-x
11. Zhou, J.: Sequence-based modeling of three-dimensional genome architecture from kilobase to chromosome scale. Nat. Genet. **54**(5), 725–734 (2022). https://doi.org/10.1038/s41588-022-01065-4
12. Robertson, G., et al.: Genome-wide profiles of STAT1 DNA association using chromatin immunoprecipitation and massively parallel sequencing. Nat. Meth. **4**(8), 651–657 (2007). https://doi.org/10.1038/nmeth1068

13. Buenrostro, J.D., Giresi, P.G., Zaba, L.C., Chang, H.Y., Greenleaf, W.J.: Transposition of native chromatin for fast and sensitive epigenomic profiling of open chromatin, DNA-binding proteins and nucleosome position. Nat. Meth. **10**(12), 1213–1218 (2013). https://doi.org/10.1038/nmeth.2688
14. Song, L., Crawford, G.E.: DNase-seq: a high-resolution technique for mapping active gene regulatory elements across the genome from mammalian cells. Cold Spring Harb. Protoc. **2010**(2), pdb.prot5384 (2010)
15. Van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. **9**(11), 2579–2605 (2008)
16. Bartsevich, V.V., Miller, J.C., Case, C.C., Pabo, C.O.: Engineered zinc finger proteins for controlling stem cell fate. Stem Cells **21**(6), 632–637 (2003)
17. Fang, Y., Gao, J., Huang, C., Peng, H., Runpu, W.: Self multi-head attention-based convolutional neural networks for fake news detection. PLoS ONE **14**(9), e0222713 (2019)
18. Li, X., Ran, L., Liu, P., Zhu, Z.: Graph convolutional networks with hierarchical multi-head attention for aspect-level sentiment classification. J. Supercomput. **78**(13), 14846–14865 (2022)
19. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)

Check for updates

Probabilistic and Machine Learning Models for the Protein Scaffold Gap Filling Problem

Kushal Badal[1], Letu Qingge[1], Xiaowen Liu[2], and Binhai Zhu[3(✉)]

[1] Department of Computer Science, North Carolina A&T State University, Greensboro, NC, USA
kbadal@aggies.ncat.edu, lqingge@ncat.edu

[2] John W. Deming Department of Medicine, Tulane University, New Orleans, LA, USA
xwliu@tulane.edu

[3] Gianforte School of Computing, Montana State University, Bozeman, MT, USA
bhz@montana.edu

Abstract. In de novo protein sequencing, we often could only obtain an incomplete protein sequence, namely scaffold, from top-down and bottom-up tandem mass spectrometry. While most sections of the proteins can be inferred from its homologous sequences, some specific section of proteins is always missing and it is hard to predict the missing amino acids in the gaps of the scaffold. Thus, we only focus on predicting the gaps based on a probabilistic algorithm and machine learning models instead predicting the complete protein sequence using generative AI models in this paper. We study two versions of the protein scaffold filling problem with known size gaps and known mass gaps. For the known size gaps version, we develop several machine learning models based on random forest, k-nearest neighbors, decision tree and fully connected neural network. For the known mass gap problem, we design a probabilistic algorithm to predict the missing amino acids in the gaps. The experimental results on both real and simulation data show that our proposed algorithms show promising results of 100% and close to 100% accuracy.

Keywords: Protein sequencing · Protein Scaffold filling · Machine learning · Probablistic model · Heuristic algorithms

1 Introduction

In the fields of proteomics, protein sequencing determines the amino acid code of a protein. Protein sequencing is a widely researched area, as it is beneficial for highlighting the structures and functions of proteins. With such information, researchers across a realm of fields in biology, chemistry, and medicine can identify and develop more effective solutions to long-standing problems, such as

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2024
W. Peng et al. (Eds.): ISBRA 2024, LNBI 14956, pp. 28–39, 2024.
https://doi.org/10.1007/978-981-97-5087-0_3

pharmaceutical drug development and understanding the role that proteins play in various diseases and conditions.

The advent of mass spectrometry marked a pivotal shift in protein sequencing technologies, offering substantial improvements over traditional methods like Edman degradation, which, despite its utility, was limited by low throughput and substantial sample requirements [4,8]. Mass spectrometry's sensitivity to attomole quantities of peptides represents a significant advancement, facilitating rapid and high-coverage data acquisition [4]. Early mass spectrometry-based strategies for peptide sequencing laid the groundwork for the sophisticated techniques in use today [2,12].

De novo sequencing and database searching are the two main methods commonly used in mass spectrometry protein sequencing [1]. With de novo protein sequencing, there are two mass spectrometry based methods, known as top-down and bottom-up approaches. Despite recent progress, most assembled proteins are still in an incomplete form with gaps in the scaffold [11]. While most sections of the proteins can be inferred from its homologous sequences, some specific section of proteins is always missing and hard to predict the missing amino acids in the gaps of the scaffold [11]. (Due to space constraint, we refer more background and references on *de novo* protein sequencing to [1,5,11].)

We study two versions of the protein scaffold gap filling problems (PSGF). In the first variant, we assume that the size of gaps (i.e., number of missing amino acids) within a protein scaffold is known. For this version, we develop several machine learning models with data pre-processing techniques to accurately predict the missing amino acids in the gaps. In the second variant, we handle the gap filling problem of known mass (of the gaps) but unknown gap size; to be precise, only the total mass of the missing amino acids required to fill the gaps is known. For this version, we design a probabilistic algorithm to fill the missing amino acids in the gaps.

When a homologous reference protein is given, a number of useful polynomial-time algorithms based on local search and dynamic programming were developed in 2017 by Qingge et al. for the protein scaffold gap filling problem [7]. The running time of their developed algorithms to obtain optimal solutions is $O(n^{26})$, where n is the size of the reference protein (also the total length of the protein to be filled) [7]. When a reference protein is not given, different innovative approaches based on deep learning models such as a convolutional neural network and long short-term memory for the protein scaffold gap filling problem were designed [10]. Sturtz et al. also introduced a convolutional denoising autoencoder model, achieving remarkable accuracy in gap filling [9]. These methods mainly focus on the protein scaffold filling problem with known gap size. In this paper, we will also solve the gap filling problem with known mass gaps in the scaffold.

2 Methodology

Given an (unknown) target protein sequence T and a protein scaffold $S = (S_1, S_2, ..., S_m)$ of m contigs, with a gap composed of missing amino acids

between contigs S_i and S_{i+1}, the protein scaffold gap filling (PSGF) problem is to fill the missing amino acids in S to obtain S' such that the number of one-to-one matching amino acids between the filled sequence S' and target sequence T is maximized. (S' is used as the predicted protein sequence for T.)

It is important to note that the exact sizes and masses of these missing gaps may not be known. Our paper stands out due to its innovative approach to addressing protein sequencing challenge by employing different algorithms. Specifically, we introduce techniques for filling gaps in two different types of PSGF problems as illustrated in Fig. 1. The first PSGF problem is focused on the case with known gap size and the second on the case with known gap mass. Machine learning models, such as random forest, k-nearest neighbors, decision tree and fully connected neural network are used for the first version. For the second PSGF problem (with known gap mass) we design a new probabilistic algorithm (See Fig. 1). We provide a detailed description of our methods to tackle both version of the PSGF problem below.

Fig. 1. An illustration for the two PSGF problems.

2.1 Data Collection

Two types of protein scaffold datasets, alemtuzumab's light chain (MabCampath) [5] and antibody light chain proteoforms of Homo sapiens (P5A) [3] collected from Dupré et al. [3] are used to evaluate the performance of our proposed methods. The basic idea is that we will fill the gaps in the given scaffolds and compare the similarity between constructed protein sequence with the ground truth of the target protein sequences of alemtuzumab's light chain (MabCampath) [5] and the antibody light chain proteoforms of Homo sapiens proteoform (P5A) [3]. The real MabCampth scaffold 1 data [5] is generated by combining the bottom-up and top-down mass spectrometry approaches, which consists of five contigs and six gaps in the scaffold. We also generate simulated scaffold 2 data from MabCampth target sequence to further test the model performance. Additionally, we generate two sets of simulated scaffold 3 and 4 datasets from the proteoforms P5A by randomly introducing gaps in the target sequence. Figure 2 shows the features of the scaffold data, in which gap sizes are denoted by red dashed lines and gap mass are given by the mass in Dalton. Scaffolds 1 and 3 has fewer and shorter gaps than scaffolds 2 and 4. These scaffold data will be used to test models in different scenarios. Moreover, we retrieve 1000 homologous sequences for each scaffold sequence from NCBI's Protein Blast Server [6] as training data in our proposed machine learning models.

Fig. 2. Target and scaffold sequences of Mabcampath and P5A proteoform.

2.2 Data Preprocessing

For the PSGF problem with known gap size, we have the input-output samples in our training dataset by generating 11-mers starting from the first position of each homologous sequence and shift it to the next position until we reach to end of the sequence. Then we introduce gaps at the start and end position of each input 11-mer to simulate scenarios where certain amino acids are missing and the corresponding masked amino acids are output labels. For instance, in a 11-mer "DIQMTQSPSSL", gaps are added to create sequences "-IQMTQSPSSL" and "DIQMTQSPSS-". The corresponding output labels for these sequences would be "D" and "L" respectively. This technique of creating training data results in training the model twice (in forward direction and reverse direction) as in [10]. Additionally, these sequences undergo a label encoding transformation, converting each amino acid into a unique numeric value and making them compatible with machine learning algorithms. We further refine the feature space through feature engineering techniques, such as singular value decomposition (SVD) and row averaging. SVD is employed for dimensional reduction while preserving essential patterns in the data, and row averaging simplifies the sequences and calculates the average of numeric representations, aiding the models in detecting significant patterns within the protein sequences. For the PSGF problem with known mass, no specific data preprocessing is needed. We directly apply our proposed algorithm to fill the scaffolds using its homologous sequences (obtained from NCBI's server).

2.3 The Proposed Models for the PSGF Problem with Known Gap Size

In this subsection, we develop machine learning models, such as decision tree, KNN, random forest and combination of these models with row average and singular value decomposition (SVD) techniques to solve the PSGF problem with known gap size. Due to space constraint, we leave out all the methods related to k-nearest neighbor and random forest to the full version.

The use of singular value decomposition (SVD) and row average in the context of processing and analyzing protein scaffold gap filling is motivated by their ability to simplify complex protein sequence data. By applying SVD, the dimensionality of the sequence can be reduced while maintaining its primary structural and functional properties. This reduction technique improves the models' capacity to correctly predict the missing amino acids in the scaffold with less time and resources. The row average approach simplifies complex protein sequences by reducing the complexity of protein sequences to a single numerical number that represents the average of the encoded values of the amino acids in a kmer. This simplicity becomes particularly advantageous when they are used as a preprocessing step for machine learning algorithms, such as decision trees or random forest or KNN, allowing for quick and efficient analysis.

Decision Tree. In this subsection, we employ a decision tree algorithm for the protein scaffold gap filling problem. The initial step involves preprocessing protein sequences into 11-mers which includes gap ("-") at the start or end positions, which will be used as input data and the corresponding amino acid at the gap position will be the output label. Algorithm 1 illustrates the proposed decision tree algorithm designed to solve protein scaffold gap filling problem. In Algorithm 1 there are terms as Gini impurity, features available for splitting and stopping criteria. Gini impurity is a way to measure how mixed up or "impure" a group of items is in decision trees. In PSGF, Gini impurity measures how mixed the target y labels (missing amino acids) are among a set of items (11-mer sequences) at a specific node in the decision tree. The best feature (position in the 11-mer) and its value are selected by decision trees using Gini impurity at each node to separate the dataset. In order to create child nodes that are more "pure" in terms of their target y labels, the algorithm looks for splits that will produce the largest decrease in Gini impurity. In decision trees, stopping criteria are guidelines or conditions that determine when the algorithm should stop dividing the nodes further. Feature availability refers to whether a feature can still be used for making further splits in a decision tree. Feature availability refers to whether a feature (every position in 11-mers) can still be used for making further splits in a decision tree. If all features have been visited or if the potential splits do not reduce Gini impurity, then no more features are available for splitting at that node. Figure 3 shows simple decision tree illustration for smaller 11-mers dataset.

We have also combined row average with decision tree provides a novel method (denoted as **Row Average + Decision Tree**) for prediction of

Algorithm 1: Decision Tree Algorithm for Protein Scaffold Gap Filling

Input: Training dataset of numerically encoded 11-mers with gaps ("-") and corresponding output labels y for each sample, which will be used to fill gaps;
Step 1: Initialize the Tree
Root node has a set of samples S including all 11-mers and corresponding output labels y; Calculate the initial Gini impurity $Gini(S)$ using the formula $Gini(S) = 1 - \sum(p_i)^2$, where p_i is the proportion of items labeled with the i^{th} y labels in the set S
Step 2: Build the Decision Tree
while *features available for splitting and the stopping criteria (Gini impurity = 0) not met* **do**
    for *each feature f at each position in the 11-mers* **do**
        Identify all unique values V at feature position f and sort them
        Calculate midpoints between consecutive values in V to determine potential thresholds
        for *each potential threshold t calculated from midpoints* **do**
            Partition S into subsets S_{left} and S_{right} based on t
            S_{left} contains all 11-mers with feature f value $\leq t$
            S_{right} contains all 11-mers with feature f value $> t$
            Calculate Gini impurity for S_{left} and S_{right}
        end
        Compute weighted Gini impurity for each split using the formula:

$$Weighted\_Gini_{f,t} = \left(\frac{|S_{left}|}{|S|}\right) \times Gini(S_{left}) + \left(\frac{|S_{right}|}{|S|}\right) \times Gini(S_{right})$$

        where $|S_{left}|$ and $|S_{right}|$ are the counts of unique output y labels in each subset, and $|S|$ is the total count of 11-mers in set S
    end
    Choose the feature f and threshold t combination with the lowest $Weighted\_Gini_{f,t}$ for splitting
    If multiple thresholds yield the same lowest $Weighted\_Gini_{f,t}$, select one randomly
    Split S into S_{left} and S_{right} using the chosen threshold t at feature f
    Create child nodes for S_{left} and S_{right}, assigning the corresponding subset to each
    Assign a class (y label) to each node based on the majority class of the subset at that node
    Recursively apply the above steps to each child node, treating each as a new root
end
Step 3: Predict Missing Amino Acid for New 11-mers with gaps
for *each 11-mer* **do**
    node ← root of the tree
    while *node is not leaf* **do**
        v ← value at *node*'s feature in 11-mer
        node ← $v \leq$ *node*'s threshold ? left child : right child
    end
    Output *node*'s class
end

missing amino acids for PSGF. An additional efficient method for predicting missing amino acids in protein sequences is to combine decision tree modeling and SVD in the PSGF (denoted as **SVD + Decision Tree**). Again, due to space constraint we will cover the details in the full version.

Algorithm 2: Algorithm for the PSGF Problem with Known Gap Masses

Set-up and the Goal:
Assume $P(B) > 0.5$ (say 0.9, which means that the contigs are more similar to ground truth) and $P(\overline{B}) < 0.5$ (say 0.1, which means that the contigs are less similar to ground truth), the algorithm considers $P(\overline{B})$ a small probability.
The goal is to maximize $P(A)$, ensure $P(A \cap B)$ is large and $P(A \cap \overline{B})$ is small.
Compute $P(A \cap B)$ and $P(A \cap \overline{B})$:
Generate all possible sequences of amino acids of the mass Δ_i and pick one at a time, say t_i.
Break $S_i t_i S_{i+1}$ into peptides of different lengths and compute the number of these peptides appearing in the raw peptide data from the input sequence, say $\alpha(t_i)$.
Among all sampled t_i, select the one with the $\max(\alpha(t_i))$. (In practice, record the largest 10, say.)
For $P(A \cap \overline{B})$, peptides are formed by the left end of S_{i+1}, t_i, and the right end of S_i.
Similarly, to compute $P(A \cap \overline{B})$, select t_i with the $\min(\beta(t_i))$. (In practice, record the smallest 10, say).
Compute $P(A)$:
If $\alpha(t_i)$ and $\beta(t_i)$ are recorded, select the t_i with $\max\{\alpha(t_i) - \beta(t_i)\}$ to maximize $P(A)$.

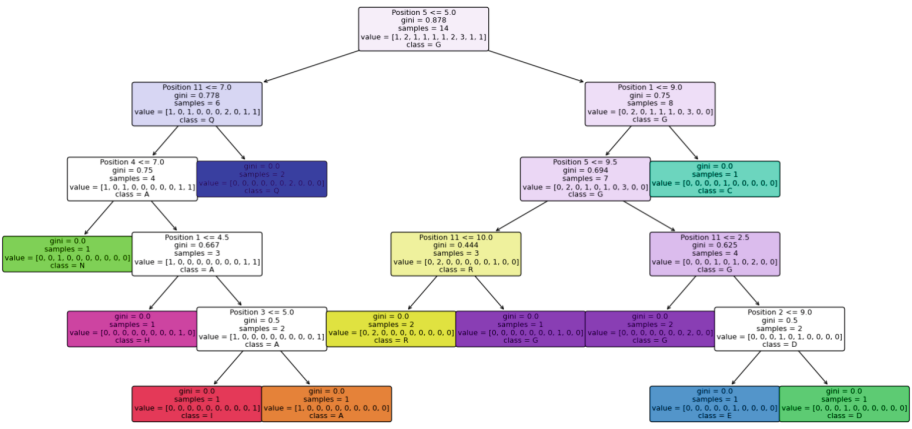

Fig. 3. Simple Decision Tree Illustration Example for PSGF.

2.4 A Probabilistic Algorithm for the PSGF Problem with Known Gap Mass

In this section, we consider another variant of the protein scaffold gap filling problem where the mass of missing amino acids in the gaps are known, while their precise sizes of how many amino acids missing are unknown. Given the scaffold $S = (S_1, S_2, \cdots, S_m)$, to fill the gap between S_i and S_{i+1} with protein sequence T_i of the right total mass Δ_i, our idea is roughly sample through all peptides with the total mass Δ_i (and we will show how to avoid a brute-force method).

Define the Event Space

Let A denote the event that the gap between contigs S_i and S_{i+1} is filled with some protein sequence T_i of the desired mass Δ_i. Let B denote the event that S_i and S_{i+1} are the correct contigs (i.e., with no error in them). The event space can be further illustrated as following

$$\underbrace{S_i \quad \overbrace{T_i}^{\text{A}} \quad S_{i+1}}_{\text{The gap between contigs is filled by } T_i \text{ with mass } \Delta_i}$$

Probability Computation:

The probability of A is calculated as:

$$P(A) = P(A|B) + P(A|\overline{B}).$$

And by the formula of conditional probability:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}, \quad P(A|\overline{B}) = \frac{P(A \cap \overline{B})}{P(\overline{B})}.$$

Algorithm 2 shows the procedure to solve the PSGF problem with known gap masses.

The main challenge of this algorithm is to generate all possible sequences of amino acids of certain mass Δ_i, even though we know the mass of each of the amino acids. To tackle this challenge we first find the minimum and maximum possible length of a sequence with the targeted mass, i.e., min_length and max_length, and then generate all sequences with a length from min_length to max_length. For a large mass max_length can be large too (and it can take $O(n^{20})$ time to generate them, where n = max_length). Hence, to make this algorithm more feasible, we generate the possible sequences of amino acids of a total mass of Δ_i using homologous sequences generated from NCBI's server for each scaffold.

We generate possible sequences of amino acids of a target mass Δ_i using the concept of kmers. For each length in [min_length, max_length] we generate kmers using homologous sequences and choose the kmers of the target mass Δ_i as a candidate for T_i. For instance, consider a certain mass 300 Daltons for

which the minimum and maximum possible length of a sequence with this mass can be 3 and 6 respectively. We can then generate kmers of length 3,4,5 and 6 respectively, using homologous sequences like "DIQMTQSPSSLSASVI...", etc. We select the kmers "DIQ", "SIS", "SGTD", "NRGEC", "IISSCT", etc., which has a total target mass $\Delta_i = 300$. Generating t_i this way makes it feasible to fill the gap with a large mass.

Additionally, when we construct the peptides from the left of S_i and the right of S_{i+1}, we simply select up to 5 amino acids from them. In other words, in $S_i t_i S_{i+1}$ and when computing $P(A \cap \overline{B})$, we only consider up to 5 amino acids on the right (resp. left) end of S_i (resp. S_{i+1}).

3 Experimental Results

We test our proposed protein scaffold gap filling problem with known size on Scaffold 1 and Scaffold 3, and test the proposed protein scaffold gap filling problem with known mass on Scaffold 2 and Scaffold 4, which have been introduced in Sect. 2.1.

3.1 Results for PSGF with Known Gap Size Using ML Models

To increase the proposed machine learning models prediction accuracy, we utilize data preprocessing techniques including row average and singular value decomposition (SVD) to extract the important features from input data. We evaluate the model performance by comparing the similarity between the filled gaps in the scaffold with the corresponding positions of target sequence. The gap filling accuracy is a fraction of the number of one-to-one matching amino acids in the gaps with the corresponding position at the target sequence with the length of amino acids in the gaps of the scaffold. Each model is used to predict the

Table 1. The Model Training and Validation Accuracy on Mabcampath Data.

	Train Acc.	Validation Acc.
KNN	94.37	92.84
Decision Tree	97.52	94.46
Random Forest	97.52	95.60
SVD + KNN	94.43	93.01
SVD + Decision Tree	97.52	92.40
SVD + Random Forest	97.52	94.16
Row Average + KNN	95.74	94.50
Row Average + Decision Tree	97.51	96.92
Row Average + Random Forest	97.51	97.11
Fully Connected NN	94.92	92.83

missing amino acids in the gaps of the scaffold one after another. The training and validation accuracy of these models on Mabcampath data are illustrated in Table 1.

To demonstrate the performance of our models on different scaffolds with larger gaps, we run the models on Scaffold 1–4 data. All models show promising performance on filling the gaps of these scaffolds. Table 4 shows the gap filling accuracy of the proposed models on MabCampath and P5A protein scaffolds. The prediction results show the proposed machine learning models can fill the gaps of the protein scaffold accurately. Our proposed models also achieve 100% gap filling accuracy compared with CNN-LSTM model developed in [10] on the real MabCampth data. Figure 4 shows the 100% gap filling prediction accuracy results on the Scaffold 1 and 3, which have smaller size of gaps. For the scaffolds 2 and 4 having the larger size of gaps, our proposed machine learning models also achieve higher prediction accuracy up to 94.73% and 100% respectively. Table 4 summarizes all models prediction accuracy on Scaffold 1–4 datasets.

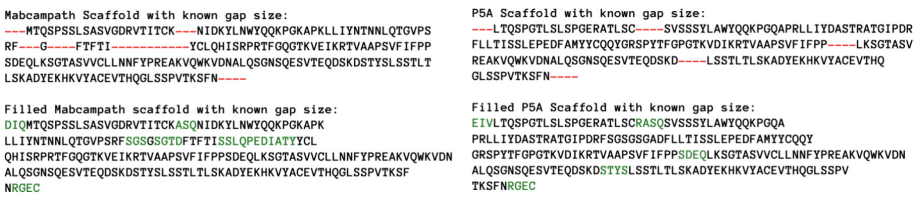

Fig. 4. Filled Mabcampath scaffold 1 and P5A proteoform scaffold 3 for all the models.

Table 2. Gap filling accuracy on MabCampath and P5A scaffold.

	Gap Filling Acc.			
	Mab		P5A	
	Scaffold 1	Scaffold 2	Scaffold 3	Scaffold 4
KNN	100.0	94.73	100.0	97.73
SVD + KNN	100.0	94.73	100.0	94.73
Row Average + KNN	100.0	94.73	100.0	94.73
Decision Tree	100.0	93.42	100.0	100.0
SVD + Decision Tree	100.0	93.42	100.0	94.73
Row Average + Decision Tree	100.0	93.42	100.0	100.0
Random Forest	100.0	93.42	100.0	100.0
SVD + Random Forest	100.0	93.42	100.0	97.36
Row Average + Random Forest	100.0	93.42	100.0	100.0
Fully Connected Neural Network	100.0	92.10	100.0	97.36

3.2 Results for PSGF with Known Gap Masses Using the Probabilistic Algorithm

We design and implement the Algorithm 2 to fill the gaps in the protein scaffold gap filling problem. To fill the gaps, we search the maximum value of $\alpha(t_i)$ and minimum value of $\beta(t_i)$ from all sample t_i. We test our algorithm on all the generated scaffolds 1–4 shown in Fig. 2. Table 3 shows $\max(\alpha(t_i))$ and $\min(\beta(t_i)))$ values of each gap for Mabcampath scaffold. We achieve 100% gap filling accuracy on all the scaffolds data. It demonstrates that our designed probabilistic algorithm can fill missing amino acids in gaps accurately in the scenario of known gap mass with unknown size of the gap. Table 3 and 4 show the computed amino acids to fill Mabcampath and P5A scaffolds.

Table 3. Gap filling on MabCampath scaffold 1 with known mass.

Gap Mass	Predicted combination	$\text{Max}(\alpha(t_i))$ and $\text{Min}(\beta(t_i))$
356 Dalton	DIQ	[572, 3]
286 Dalton	ASQ	[3, 9]
231 Dalton	SGS	[734, 31]
360 Dalton	SGTD	[70, 24]
1204 Dalton	SSLQPEDIATY	[8, 2]
445 Dalton	RGEC	[886, 0]

Table 4. Gap filling on P5A scaffold 3 with known mass.

Gap Mass	Predicted combination	$\text{Max}(\alpha(t_i))$ and $\text{Min}(\beta(t_i))$
341 Dalton	EIV	[292, 0]
442 Dalton	RASQ	[238, 200]
459 Dalton	SDEQ	[991, 0]
438 Dalton	STYS	[990, 3]
445 Dalton	RGEC	[944, 0]

4 Conclusions

In this paper, we consider two versions of the protein scaffold gap filling (PSGF) problem. For PSGF with known gap size, we propose several machine learning models combined with data preprocessing engineering techniques, such as decision tree, KNN, random forest and also develop fully connected neural network

to fill the gaps iteratively until all the gaps are filled. For the PSGF problem with known gap mass, we design a probabilistic model to fill the missing amino acids in the gaps. The experimental results on different scenario of scaffolds show that our proposed algorithms achieve promising results on both real and simulation datasets, in fact, with 100% or close to 100% accuracy in general. Moreover, the proposed algorithms are simple, yet effective to solve the protein scaffold gap filling problem.

Acknowledgements. This work is supported by the NSF of the United States under Award 2307571, 2307572 and 2307573. We also thank anonymous reviewers for their insightful comments and inputs.

References

1. Aebersold, R., Mann, M.: Mass spectrometry-based proteomics. Nature **422**(6928), 198–207 (2003)
2. Bricas, E., Van Heijenoort, J., Barber, M., Wolstenholme, W., Das, B., Lederer, E.: Determination of amino acid sequences in oligopeptides by mass spectrometry. IV. Synthetic n-acyl oligopeptide methyl esters. Biochemistry **4**(10), 2254–2260 (1965)
3. Dupré, M., et al.: De novo sequencing of antibody light chain proteoforms from patients with multiple myeloma. Anal. Chem. **93**(30), 10627–10634 (2021). pMID: 34292722. https://doi.org/10.1021/acs.analchem.1c01955
4. Kinter, M., Sherman, N.E.: Protein Sequencing and Identification Using Tandem Mass Spectrometry. Wiley, Hoboken (2005)
5. Liu, X., et al.: De novo protein sequencing by combining top-down and bottom-up tandem mass spectra. J. Proteome Res. **13**(7), 3241–3248 (2014)
6. National Center for Biotechnology Information: Blast (2023). https://blast.ncbi.nlm.nih.gov/Blast.cgi?PAGE=Proteins
7. Qingge, L., Liu, X., Zhong, F., Zhu, B.: Filling a protein scaffold with a reference. IEEE Trans. Nanobiosci. **16**(2), 123–130 (2017)
8. Standing, K.G.: Peptide and protein de novo sequencing by mass spectrometry. Curr. Opin. Struct. Biol. **13**(5), 595–601 (2003)
9. Sturtz, J., Annan, R., Zhu, B., Liu, X., Qingge, L.: A convolutional denoising autoencoder for protein scaffold filling. In: Guo, X., Mangul, S., Patterson, M., Zelikovsky, A. (eds.) Bioinformatics Research and Applications, ISBRA 2023. LNCS, vol. 14248, pp. 518–529. Springer, Singapore (2023). https://doi.org/10.1007/978-981-99-7074-2_42
10. Sturtz, J., Zhu, B., Liu, X., Fu, X., Yuan, X., Qingge, L.: Deep learning approaches for the protein scaffold filling problem. In: 2022 IEEE 34th International Conference on Tools with Artificial Intelligence (ICTAI), pp. 1055–1061. IEEE (2022)
11. Tran, N.H., Rahman, M.Z., He, L., Xin, L., Shan, B., Li, M.: Complete de novo assembly of monoclonal antibody sequences. Sci. Rep. **6**(1), 1–10 (2016)
12. Wulfson, N., et al.: Mass spectrometric determination of the amino (hydroxy) acid sequence in peptides and depsipeptides. Tetrahedron Lett. **6**(32), 2805–2812 (1965)

Patient Anticancer Drug Response Prediction Based on Single-Cell Deconvolution

Wei Peng[1,2(✉)], Chuyue Chen[1], and Wei Dai[1,2]

[1] Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650050, China
weipeng1980@gmail.com, daiwei@kust.edu.cn

[2] Computer Technology Application Key Lab of Yunnan Province, Kunming University of Science and Technology, Kunming 650050, China

Abstract. Predicting patient responses to anticancer drugs is essential for the development of effective treatment strategies. Tumors, intricate structures comprised of diverse cell types, exhibit considerable cellular heterogeneity. Leveraging single-cell data offers a promising avenue for deciphering this complexity and enhancing the accuracy of drug response prediction. In this study, we propose the Single-Cell Deconvolution Guided method for Patient Anticancer Drug Response Prediction (ScPDRP). ScPDRP utilizes single-cell gene expression data to deconvolve both cell line and patient gene expression profiles. Through the employment of several encoders and generative adversarial training, ScPDRP extracts domain-invariant features from cell line and patient data, facilitating downstream drug response prediction tasks. Evaluation of our model on a curated selection of drug datasets from the clinical TCGA dataset demonstrates its superior performance over existing state-of-the-art methods across nearly all drug datasets.

Keywords: deconvolution · single-cell · transfer learning · anticancer drug response

1 Introduction

Cancer is the abnormal growth of cells that can be life-threatening. Anticancer drugs can help treat cancer by stopping the growth and spread of cancer cells. Predicting patient anticancer drug responses accurately is crucial for developing personalized cancer treatment plans [1]. Tumors are made up of different types of cancer cells. Single-cell data can offer insight into the tumor cellular heterogeneity and improve drug sensitivity prediction. However, single-cell sequencing is expensive, so more previous drug response studies have used large amounts of gene expression data from in vitro cell lines.

Several publicly available databases, such as the Cancer Cell Line Encyclopedia (CCLE) [2] and the Genomics of Drug Sensitivity in Cancer (GDSC) [3], house a plethora of in vitro cell line responses to drugs. Much prior research has focused on developing drug response prediction methods utilizing these databases. Li et al. proposed DeepDSC [4], a model that uses stacked autoencoders to extract cellular features

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2024
W. Peng et al. (Eds.): ISBRA 2024, LNBI 14956, pp. 40–51, 2024.
https://doi.org/10.1007/978-981-97-5087-0_4

from gene expression data, which are subsequently combined with medicinal chemistry features for drug response prediction. Liu et al. proposed DeepCDR [5], a GCN-based anticancer drug response predictor that integrates multi-omics features of cell lines and drugs. Peng et al. proposed the MOFGCN [6], which uses a graphical convolutional neural network to find similarities between cell lines and drugs. Liu et al. proposed the NIHGCN [7], which uses a heterogeneous network to show how drugs and cell lines interact in complex ways.

The models described above are lack of effectiveness in predicting patient responses to anticancer drugs because of differences in the distribution of cell line data versus patient data. But it is impossible to directly train an effective model due to the lack of high-quality patient drug response data. Recently, many methods for predicting patient anticancer drug response using transfer learning [8] have been proposed. These methods transfer knowledge from cell line-drug response to the patient domain. Ma proposed TCRP [9], which was first trained on a large cell line dataset and then fine-tuned with a limited amount of patient drug response data. TCRP still requires labelled patient domain data, so more methods map source and target data into a shared latent space. These methods use constraints to train the model, e.g., Maximum Mean Distribution Discrepancy (MMD) [10], Covariance Difference [11], and Generative Adversarial Methods [12] to achieve feature adaptation in the latent space. Chen and Ma [10] transferred features efficiently based on MMD loss by mapping gene expression features of cancer cell lines and clinical samples to a shared latent space and maintaining consistency. B. Dincer and Lee [13] introduced an adversarial module in the feature learning process of an autoencoder to generate more efficient features. He and Wu [14] created several autoencoders to extract shared and private features and added an adversarial module to help adapt feature distributions. Sharifi-Noghabi et al. proposed AITL [15], which aligns feature distributions in the source and target domains through adversarial domain adaptation and multi-task learning.

However, none of the patient drug response prediction models mentioned above consider the heterogeneity of cells within tumors. Patient tissue samples and cell lines often contain multiple cell types, resulting in gene expression profiles that are a mixture of different cell types. Simple feature alignment methods can obscure the uniqueness and diversity of individual cells in patient tissue and cell line samples. These methods can also blur the distinction between drug-sensitive and drug-resistant cells by aligning their features. By using single-cell data, we can resolve the cellular heterogeneity of tumors and achieve a more accurate alignment between cell line and patient data [16]. This provides new insights for predicting drug sensitivity.

Considering the above, we introduce the Single-Cell Deconvolution Guided method for Patient Anticancer Drug Response Prediction, denoted as ScPDRP. ScPDRP employs single-cell data to deconvolve both cell line and patient data, discerning the single-cell composition of cell lines and patient tissue samples. This approach provides a comprehensive characterization of tumor cell biology. We align the features of cell lines and tissue samples from a single-cell perspective, crucial for analyzing the tumor microenvironment of patients and enhancing the accuracy of predicting their response to anticancer drugs. Subsequently, we utilize the learned domain-invariant features for downstream tasks associated with anticancer drug response prediction. The results of an

experiment conducted on the clinical dataset The Cancer Genome Atlas (TCGA) [17] demonstrated that ScPDRP outperformed existing algorithms.

2 Materials

In this study, we utilize gene expression data for ScPDRP, where the source domain data comprises cell line gene expression data, and the target domain data consists of patient gene expression data. The model employs single-cell gene expression data to deconvolve these datasets, subsequently transferring knowledge from the source domain to the target domain. Our source domain datasets comprise CCLE and GDSC, while Our target domain dataset is TCGA. These datasets were widely recognized and utilized within the fields of bioinformatics and drug development. In addition, the single-cell data utilized for the deconvolution step were sourced from the Broad Institute's Single Cell Portal (SCP198), comprising raw data for 198 cancer cell lines [18].

For the cell lines, we first used the cell line names of the samples to find their corresponding single-cell gene expression data in the 198 cell lines single-cell RNA-seq. Then, we obtained the corresponding single-cell RNA-seq data for the remaining cell lines based on the correspondence between the lineage to which the cell line samples belonged and the tissue to which the single-cell data belonged. This gene expression data will be used for our subsequent analysis. Finally, we removed cell line samples that were not associated with these 198 single cell data, and filtered out a total of 1067 cell lines as the source domain data for the analysis in this paper. Furthermore, we collected cell line anticancer drug response data from two versions of the GDSC database: GDSC1 and GDSC2. The data collected from GDSC1 amounted to 310,905 cell line anticancer drug responses, while GDSC2 contributed 135,243. In cases where there were duplicate samples in both databases, we retained the response data of GDSC2.

For the patients, we collected gene expression data for 9808 patients from the clinical dataset TCGA and we identify the corresponding tissues or organs of TCGA sample based on the "Sample-Type" item in the sample information file. We then corresponded the tissue type of the patient to the tissue type of the single cell to obtain the single cell RNA-seq data corresponding to the patient's tissue. In cases where there is no specific matching single-cell data for a given sample, we deconvolve the sample using all 198 cell lines single-cell data samples and apply PCA to reduce its dimensionality.

The cell line and patient's gene expression data involved 1,426 genes. We selected the top 1,000 genes with the highest percentage of unique expression values from cancer cell lines and tumor tissue samples using the gene selection method. Then, we combined the two gene sets to obtain a total of 1,426 genes.

We chose drugs of interest and constructed labels for samples associated with these drugs using the following approach. For drugs recorded in TCGA, patient clinical drug responses are provided in the recent work, which includes records of clinical responses to two chemotherapy drugs, gemcitabine and fluorouracil (referred to as TGem and TFu, respectively). In addition, we extracted patients' "new tumor events in the days after treatment" from TCGA as a criterion to classify patients into sensitivity and resistance and used the median number of days to new tumor events as a threshold to construct binary labels for the samples. We included in this test dataset only patients who received

a single agent throughout the treatment period. We selected drugs with more than 20 labelled samples, namely cisplatin, 5-fluorouracil, gemcitabine, sorafenib, and temozolomide (referred to as Cis, Fu, Gem, Sor, and Tem, respectively). All drugs mentioned above were also present in the GDSC database. We established associations between samples in GDSC and those in CCLE by utilizing the cosmic-id provided in the GDSC data to create labels for the samples in CCLE. Utilizing Z-scores from GDSC drug response data, we set a threshold of 0 to create binary labels of sensitivity and resistance for cell line-drug responses. Following data processing, we obtained the number of labelled samples in TCGA, as illustrated in Table 1.

Table 1. The number of labelled samples for each drug in TCGA.

Drug	CCLE		TCGA	
	resistance	sensitivity	resistance	sensitivity
Cis	275	271	20	20
Fu	319	228	11	10
Gem	307	245	23	23
Sor	261	224	13	13
Tem	274	266	23	23
TGem	307	245	55	37
TFu	319	228	25	35

3 Methods

Our model consists of three key steps to predict patient anticancer drug response, as depicted in Fig. 1. First, we process single-cell raw data and used them to deconvolve gene expression data from source-domain cell lines and target-domain patients. This deconvolution process aids in uncovering the underlying cellular composition of the samples, facilitating more accurate analysis. Second, we extracted domain-invariant features for the source and target domains. Finally, the domain-invariant features are used to train classifiers for downstream tasks to evaluate patient anticancer drug response prediction performance.

3.1 Gene Expression Data Deconvolution

In this step, we utilize single-cell data to deconvolve the gene expression profiles of both the source-domain cell line gene expression data and the target-domain patient gene expression data at the cellular level. First, we process the raw single-cell data. This involves two steps: (1) applying the log(x+1) operation to the single-cell gene expression data; (2) performing PCA on the single-cell samples. After applying PCA, we obtained multiple single-cell matrices consisting of 20 rows and 1426 columns, representing the

desired genes. These operations are commonly used in single-cell data processing. We input the processed single-cell gene expression data, cell line gene expression data, and patient gene expression data into the deconvolution module. Our process for constructing a deconvolution module is similar to CIBERSORT [19].

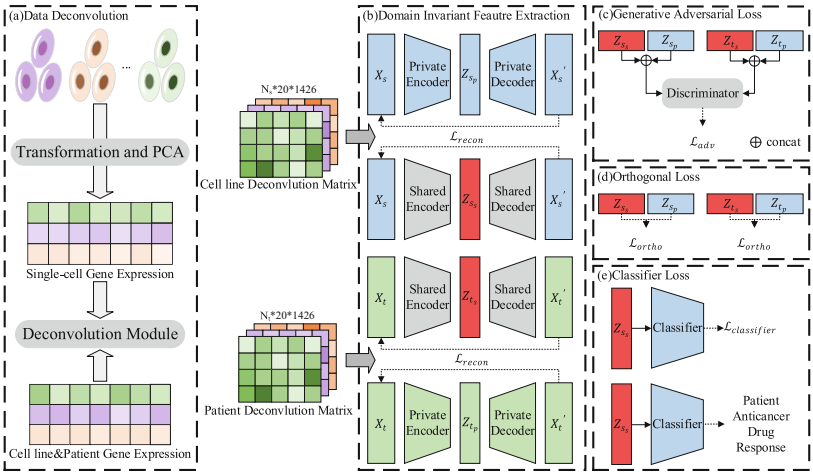

Fig. 1. Workflow of ScPDRP

We conducted a deconvolution operation using nu support vector regression (nu-SVR) [20] to determine the contribution of each single-cell sample $x_{sc}^{(i)}$ of tissue or lineage associated with the cell line sample $x_s^{(i)}$ or patient sample $x_t^{(i)}$:

$$x^{(i)} = \sum_{i=1}^{n} \omega_i x_{sc}^{(i)} \tag{1}$$

where n = 20, ω_i means the contribution of sample $x_{sc}^{(i)}$ to $x^{(i)}$. Each sample after the reverse convolution is then represented as:

$$X^{(i)} = \begin{bmatrix} \omega_1 x_{sc}^{(1)} \\ \omega_2 x_{sc}^{(2)} \\ \cdots \\ \omega_n x_{sc}^{(n)} \end{bmatrix} \tag{2}$$

After this step, we get the cell line data and patient data after deconvolution denoted as $X_s = \left\{X_s^{(i)}\right\}_{i=1}^{N_s}$ and $X_t = \left\{X_t^{(i)}\right\}_{i=1}^{N_t}$, respectively.

3.2 Domain Invariant Feature Extraction

In this stage, we aim to construct a domain-invariant feature extraction module to extract domain-invariant features of cell line data and patient data. Based on the idea of domain

separation network [21], we constructed three autoencoder modules to extract the respective private and domain-invariant features of cell line data and patient data, including two private autoencoder modules and one shared autoencoder module, as depicted in Fig. 1(b). We employ CNN as the primary component of the encode. We utilize three convolutional layers, three pooling layers, and two linear layers. For n samples, the dimension is n × 20 × 1426, and after the encoder the dimension will become n × 32. Furthermore, the decoder's structure mirrors that of the encoder, featuring the inversion of fully connected layers achieved via a fully connected layer with reversed input and output dimensions, the inversion of CNN through the Conv-transpose layer, and the inversion of the pooling layer accomplished through the interpolate operation.

Utilizing both shared and private encoders, we encode the deconvolution data X_s and X_t of the source and target domains into shared and private source as well as target domain features Z_{s_s}, Z_{t_s}, Z_{s_p}, Z_{t_p}, respectively. Subsequently, the decoder reconstructs the obtained 32-dimensional vectors to X'.

The domain-invariant feature extraction module trained twice. First, we pretrain the module. We introduce a reconstruction loss to ensure that the data restored by the decoder is similar to the input deconvolutional data:

$$\mathcal{L}_{recon} = \sum_{d\in\{s,t\}} \frac{1}{N_d} \sum_{i=1}^{N_d} \left(X_d^{(i)} - X_d^{(i)} \right)^2 \tag{3}$$

where d denotes source domain s and target domains t, N_d denotes the number of samples in domain d.

In addition, we introduce an orthogonal loss for the constraints on the features generated by the private and shared encoders, with the aim of ensuring that the private encoder generates private features that are not relevant to the prediction of drug response:

$$\mathcal{L}_{ortho} = \frac{1}{n} \left\{ \left[\left(\frac{Z_{s_p}}{\|Z_{s_p}\|_2} \right)^{\mathrm{T}} \frac{Z_{s_s}}{\|Z_{s_s}\|_2} \right]^2 + \left[\left(\frac{Z_{t_p}}{\|Z_{t_p}\|_2} \right)^{\mathrm{T}} \frac{Z_{t_s}}{\|Z_{t_s}\|_2} \right]^2 \right\} \tag{4}$$

where $\|\cdot\|_2$ denotes the L2 normalization and n means the num of data in a batch. We pre-train the model using Eq. 3 and Eq. 4, while using an early-stopping strategy, where we stop the first pre-training step of the module if the sum of the reconstruction loss and the orthogonal loss does not decrease for 20 consecutive times during the training process.

After the first pre-training step, we introduced WGAN-GP [22] for the second training of the domain-invariant feature extraction module to better fit the features. Introducing a discriminator D and the generating adversarial constraints can encourage shared encoders to generate domain-invariant features. The loss function is as follows:

$$\mathcal{L}_{adv} : \begin{cases} \mathcal{L}_{critic} = \frac{1}{N_t} \sum_{i=1}^{N_t} D\left(z_{\mathrm{t}}^{(i)}\right) - \frac{1}{N_s} \sum_{s=1}^{N_s} D\left(z_{\mathrm{s}}^{(i)}\right) + +\lambda(\|\nabla_{\tilde{z}} D(\tilde{z})\|_2 - 1)^2 \\ \mathcal{L}_{gen} = -\frac{1}{N_{\mathrm{t}}} \sum_{i=1}^{N_{\mathrm{t}}} D\left(z_{\mathrm{t}}^{(i)}\right) \end{cases} \tag{5}$$

where $z_t^{(i)} = z_{t_s}^{(i)} \oplus z_{t_p}^{(i)}$, $z_s^{(i)} = z_{s_s}^{(i)} \oplus z_{s_p}^{(i)}$, $\|\cdot\|_2$ denotes the L2 normalization, λ is a hyper parameter and we set it to 10, and $\tilde{z} = \epsilon z_s + (1-\epsilon)z_t$ in which the ϵ is sampled from a standard normal distribution $U(0, 1)$. During the second training of the module, in every epoch, we train the model four times using $\mathcal{L}_{gen}$ and then once using $\mathcal{L}_{critic} + \mathcal{L}_{ortho} + \mathcal{L}_{recon}$. At the same time, we use an early-stopping strategy, where we stop training the model if the validation set is not decreasing in loss for 20 consecutive times.

We use a learning rate of 0.001 and a cosine annealing strategy in both training processes of the domain-invariant feature extraction module. At the same time, we save the domain-invariant feature extraction module for different number of pre-training epochs and number of generative adversarial training epochs.

3.3 Training Classifier

In this stage, we use the features learned from the domain-invariant feature extraction module for patient anticancer drug response prediction. We construct a classifier C, which consists of two linear layers and a layer of Relu activation function. The model was trained using the training set data from the source domain, and the training loss is presented below:

$$\mathcal{L}_{Classifier} = -\frac{1}{N_{train}} \sum_{i=1}^{N_{train}} (1-y_i) log\left[1-\sigma\left(C\left(z_{s_s(i)}\right)\right)\right] + y_i log\sigma\left(C\left(z_{s_{s(i)}}\right)\right) \quad (6)$$

where $\sigma(\cdot)$ is the sigmoid function, $z_{s_s(i)}$ denotes the ith sample in z_{s_s}, and y_i denotes its corresponding true label. We divide the validation set from the source domain data, and if the AUC of the validation set does not rise for 20 consecutive times, we stop the training of the model and save the model with the highest AUC in the validation set.

Finally, we employ the patient's domain-invariant features as the test set and utilize the trained completed model to predict the patient anticancer drug response.

4 Experiment

To evaluate the performance of ScPDRP, we experimentally compare the model with the following state-of-the-art baseline methods, including Codeae-adv [14], Codeae-mmd [14], ADAE [13], DAE [23], DSN [21] and its variant DSN-DANN are comprehensively compared. We used unlabeled CCLE data and TCGA data for domain invariant feature learning. During classifier training, we divide the source domain data into training set: validation set = 4:1 and consider the target domain data as test set. We perform a 5-fold cross-validation, and the average AUC and AUPRC of the test set in the 5-fold are used to evaluate the model performance.

4.1 Drug Response Prediction for Clinical TCGA Dataset

We evaluated the performance of our model on the TCGA dataset. Table 2 shows the 5-fold mean AUC, AUPRC and their variances for ScPDRP compared to individual

Table 2. TCGA dataset AUC and AUPRC

	Algorithm	Cis	Fu	Gem	Sor	Tem	TGem	Tfu
AUC	Codeae-adv	0.5985 ± 2E-2	0.6147 ± 4E-3	0.5334 ± 8E-3	0.5714 ± 8E-3	0.7316 ± 1E-4	0.5510 ± 2E-3	0.5883 ± 7E-3
	Codeae-mmd	0.5326 ± 6E-3	0.5492 ± 9E-3	0.5240 ± 2E-3	0.4817 ± 9E-4	0.6011 ± 3E-4	0.5644 ± 4E-4	0.5766 ± 3E-3
	ADAE	0.5445 ± 4E-3	0.5727 ± 9E-4	0.4738 ± 6E-3	0.5796 ± 1E-3	0.6507 ± 5E-4	0.4647 ± 2E-3	0.6095 ± 8E-3
	DSNA	0.5822 ± 9E-3	0.5873 ± 5E-4	0.5336 ± 5E-3	0.5148 ± 7E-3	0.6994 ± 8E-4	0.5579 ± 5E-5	0.6264 ± 4E-3
	DSN	0.5790 ± 2E-3	0.5818 ± 6E-3	0.5278 ± 3E-3	0.4837 ± 1E-2	0.6328 ± 5E-3	0.5366 ± 6E-4	0.5614 ± 6E-3
	DAE	0.5900 ± 3E-3	0.5200 ± 2E-3	0.5075 ± 4E-3	0.4609 ± 5E-3	0.6042 ± 3E-3	0.4553 ± 2E-3	0.5914 ± 5E-3
	ScPDRP	**0.6805 ± 1E-5**	**0.7800 ± 8E-3**	**0.6374 ± 3E-4**	**0.7704 ± 4E-3**	**0.7328 ± 1E-5**	**0.5993 ± 3E-4**	**0.6634 ± 9E-3**
AUPRC	Codeae-adv	0.5573 ± 8E-3	0.6393 ± 1E-2	0.5496 ± 7E-3	0.5802 ± 1E-2	0.7364 ± 3E-4	0.4225 ± 2E-3	0.6482 ± 5E-3
	Codeae-mmd	0.5008 ± 4E-3	0.5303 ± 8E-3	0.5685 ± 2E-3	0.5125 ± 2E-3	0.6129 ± 2E-3	0.4337 ± 4E-4	0.6321 ± 2E-3
	ADAE	0.5567 ± 8E-3	0.5416 ± 5E-3	0.5019 ± 6E-3	0.6111 ± 2E-3	0.7132 ± 4E-4	0.3982 ± 1E-3	0.6976 ± 4E-3
	DSNA	0.5633 ± 8E-3	0.5443 ± 6E-3	0.5752 ± 6E-3	0.5704 ± 1E-2	0.7433 ± 7E-4	0.4594 ± 1E-3	**0.7208 ± 1E-3**
	DSN	0.5269 ± 4E-3	0.5269 ± 4E-3	0.5768 ± 5E-3	0.5353 ± 1E-2	0.6607 ± 5E-3	0.4614 ± 5E-4	0.6597 ± 6E-3
	DAE	0.5243 ± 3E-3	0.5130 ± 3E-3	0.5448 ± 5E-3	0.5194 ± 6E-3	0.5867 ± 3E-3	0.3727 ± 9E-4	0.6852 ± 3E-3
	ScPDRP	**0.6932 ± 8E-4**	**0.8181 ± 2E-3**	**0.6561 ± 2E-4**	**0.7787 ± 4E-3**	**0.7587 ± 3E-4**	**0.4888 ± 3E-4**	0.7133 ± 2E-3

Table 3. AUC and AUPRC for ablation experiments

	Algorithm	Cis	Fu	Gem	Sor	Tem	TGem	Tfu
AUC	ScPDRP-DP	0.6012 ± 2E-3	0.6416 ± 1E-3	0.5445 ± 4E-4	0.6021 ± 3E-4	0.7032 ± 1E-4	0.5441 ± 3E-3	0.5844 ± 3E-3
	ScPDRP-DIF	0.5413 ± 4E-3	0.5524 ± 1E-3	0.5315 ± 3E-3	0.4910 ± 1E-3	0.5913 ± 2E-3	0.4831 ± 4E-4	0.5345 ± 7E-3
	ScPDRP	**0.6805 ± 1E-5**	**0.7800 ± 8E-3**	**0.6374 ± 3E-4**	**0.7704 ± 4E-3**	**0.7328 ± 1E-5**	**0.5993 ± 3E-4**	**0.6634 ± 9E-3**
AUPRC	ScPDRP-DP	0.6212 ± 4E-3	0.6678 ± 2E-3	0.5560 ± 1E-4	0.6092 ± 3E-4	0.7151 ± 2E-3	0.4354 ± 3E-3	0.6274 ± 3E-3
	ScPDRP-DIF	0.5314 ± 2E-2	0.5679 ± 3E-3	0.5347 ± 3E-3	0.5117 ± 1E-3	0.5913 ± 1E-2	0.4246 ± 5E-3	0.5565 ± 2E-3
	ScPDRP	**0.6932 ± 8E-4**	**0.8181 ± 2E-3**	**0.6561 ± 2E-4**	**0.7787 ± 4E-3**	**0.7587 ± 3E-4**	**0.4888 ± 3E-4**	**0.7133 ± 2E-3**

baseline methods across the 7 drug-labelled datasets of the TCGA. The results show that ScPDRP performs best on TCGA.As can be seen from the table, compared to Codeae-adv, which is the best overall performer among the baseline methods, our method is 8.2%, 16.53%, 10.4%, 19.9%, 0.12%, 4.83%, and 7.51% higher on the AUC of the seven drug datasets, and also higher on the AUPRC by 13.59%, 17.88%, 10.65%, 19.85%, 2.23%, 6.63%, and 6.51% on the seven drug datasets respectively. These results strongly suggest that our method, ScPDRP, has superior performance compared to other baseline methods in predicting anticancer drug responses in clinical datasets.

4.2 Results of Ablation Experiments

For ScPDRP, we performed ablation experiments to verify the effectiveness of the deconvolution process and the domain-invariant feature extraction module. Since the data for each sample after the deconvolution is a two-dimensional matrix and the data for each sample before the deconvolution is a one-dimensional vector, after ablating the deconvolution step, we modified the encoder and decoder of the model to be constructed as simple full connections. The AUC and AUPRC results of our ablation experiments are shown in Table 3. Where ScPDRP-DP refers to ablating our model away from the deconvolution process and ScPDRP-DIF refers to ablating our model away from the domain-invariant feature extraction process.

As can be seen from the table, the absence of a domain-invariant feature extraction phase has a very large impact on the performance of the model, and the underlying reason is still because of the difference in distribution between the cell line data and the patient data. In addition, the results in the table show that the performance of our model is greatly improved after performing the deconvolution operation on the data, especially on Fu and Sor drugs, the AUC results are improved by about 14% and 17% respectively, and the AUPRC results are improved by about 15% and 17% respectively.

5 Conclusion

This study introduces a Single-Cell Deconvolution Guided method for Patient Anticancer Drug Response Prediction (ScPDRP). Recognizing the cellular heterogeneity inherent in tumors, ScPDRP leverages single-cell data to deconvolve both cell line and patient data, thereby elucidating the single-cell composition of cell lines and patient tissue samples and providing a comprehensive characterization of tumor cell biology. Subsequently, domain-invariant features are learned through the construction of multiple autoencoder modules primarily based on CNN, augmented with generative adversarial training to achieve feature alignment at the cellular level. The effectiveness of ScPDRP in predicting anticancer drug responses is evaluated on TCGA and PDTC datasets. Compared to other baseline methods, ScPDRP demonstrates strong competitiveness on these datasets, achieving the highest AUC and AUPRC for the majority of drugs. Nevertheless, there remains room for improvement. Bias might occur in the deconvolution process, wherein certain patient and cell line samples utilize single-cell data solely from their corresponding organ, rather than data specific to a particular disease tissue. In addition, our future research will focus on developing enhanced deconvolution models to ensure more accurate resolution of biological features in tumor samples.

Acknowledgements. This work is supported in part by the National Natural Science Foundation of China (No. 61972185). Yunnan Ten Thousand Talents Plan young.

References

1. Menden, M., et al.: Community assessment to advance computational prediction of cancer drug combinations in a pharmacogenomic screen. Nat. Commun. **10**, 2674 (2019). https://doi.org/10.1038/s41467-019-09799-2
2. Barretina, J., et al.: The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity. Nature **483**, 603–607 (2012). https://doi.org/10.1038/nature11003
3. Yang, W., et al.: Genomics of Drug Sensitivity in Cancer (GDSC): a resource for therapeutic biomarker discovery in cancer cells. Nucleic Acids Res. **41**(D1), D955–D961 (2012). https://doi.org/10.1093/nar/gks1111
4. Li, M., et al.: DeepDSC: a deep learning method to predict drug sensitivity of cancer cell lines. IEEE/ACM Trans. Comput. Biol. Bioinform. **18**(2), 575–582 (2021)
5. Liu, Q., Hu, Z., Jiang, R., Zhou, M.J.B.: DeepCDR: a hybrid graph convolutional network for predicting cancer drug response. Bioinformatics **36**, i911–i918 (2020)
6. Peng, W., Chen, T., Dai, W.: Predicting drug response based on multi-omics fusion and graph convolution. IEEE J. Biomed. Health Inf. **26**, 1384–1393 (2021)
7. Peng, W., Liu, H., Dai, W., Yu, N., Wang, J.: Predicting cancer drug response using parallel heterogeneous graph convolutional networks with neighborhood interactions. Bioinformatics **38**, 4546–4553 (2022)
8. Zhuang, F., et al.: A comprehensive survey on transfer learning. Proc. IEEE **109**, 43–76 (2020)
9. Ma, J., et al.: Few-shot learning creates predictive models of drug response that translate from high-throughput screens to individual patients. Nat. Cancer **2**, 233–244 (2021)
10. Chen, J., et al.: Deep transfer learning of cancer drug responses by integrating bulk and single-cell RNA-seq data. Nat. Commun. **13**, 6494 (2022). https://doi.org/10.1038/s41467-022-34277-7
11. Sharifi-Noghabi, H., Harjandi, P.A., Zolotareva, O., Collins, C.C., Ester, M.: Out-of-distribution generalization from labelled and unlabelled gene expression data for drug response prediction. Nat. Mach. Intell. **3**, 962–972 (2021). https://doi.org/10.1038/s42256-021-00408-w
12. Peres, R., da Silva, C., Suphavilai, N.N.: TUGDA: task uncertainty guided domain adaptation for robust generalization of cancer drug response prediction from in vitro to in vivo settings. Bioinformatics **37**(Suppl._1), i76–i83 (2021). https://doi.org/10.1093/bioinformatics/btab299
13. Dincer, A.B., Janizek, J.D., Lee, S.-I.: Adversarial deconfounding autoencoder for learning robust gene expression embeddings. Bioinformatics **36**, i573–i582 (2020)
14. He, D., Liu, Q., Wu, Y., Xie, L.: A context-aware deconfounding autoencoder for robust prediction of personalized clinical drug response from cell-line compound screening. Nat. Mach. Intell. **4**, 879–892 (2022). https://doi.org/10.1038/s42256-022-00541-0
15. Sharifi-Noghabi, H., Peng, S., Zolotareva, O., Collins, C.C., Ester, M.: AITL: adversarial inductive transfer learning with input and output space adaptation for pharmacogenomics. Bioinformatics **36**(Suppl._1), i380–i388 (2020). https://doi.org/10.1093/bioinformatics/btaa442
16. Suphavilai, C., et al.: Predicting heterogeneity in clone-specific therapeutic vulnerabilities using single-cell transcriptomic signatures. Genome Med. **13**, 189 (2021). https://doi.org/10.1186/s13073-021-01000-y

17. Tomczak, K., Czerwińska, P., Wiznerowicz, M.: The Cancer Genome Atlas (TCGA): an immeasurable source of knowledge. Contemp. Oncol. (Pozn) **19**, A68–A77 (2015)
18. Kinker, G.S., et al.: Pan-cancer single-cell RNA-seq identifies recurring programs of cellular heterogeneity. Nat. Genet. **52**, 1208–1218 (2020). https://doi.org/10.1038/s41588-020-00726-6
19. Newman, A.M., et al.: Robust enumeration of cell subsets from tissue expression profiles. Nat. Meth. **12**, 453–457 (2015)
20. Schölkopf, B., Smola, A.J., Williamson, R.C., Bartlett, P.L.: New support vector algorithms. Neural Comput. **12**, 1207–1245 (2000)
21. Bousmalis, K., Trigeorgis, G., Silberman, N., Krishnan, D., Erhan, D.: Domain separation networks. In: Advances in Neural Information Processing Systems, vol. 29 (2016)
22. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: International Conference on Machine Learning, pp. 214–223. PMLR (2017)
23. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.-A.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th International Conference on Machine Learning, pp. 1096–1103 (2008)

A Data Set of Paired Structural Segments Between Protein Data Bank and AlphaFold DB for Medium-Resolution Cryo-EM Density Maps: A Gap in Overall Structural Quality

Thu Nguyen[1], Willy Wriggers[2], and Jing He[1(✉)]

[1] Department of Computer Science, Old Dominion University, Norfolk, VA 23529, USA
jhe@cs.odu.edu

[2] Department of Mechanical and Aerospace Engineering, Old Dominion University, Norfolk, VA 23529, USA

Abstract. The recent publication of the AlphaFold Protein Structure Database (AlphaFold DB) offers an opportunity to revisit and refine many existing structural models in computational biology. We tested the hypothesis that models determined from less-than-ideal cryo-electron microscopy (cryo-EM) data could benefit from more recent AlphaFold predictions. Ideally, atomic structures in the Protein Data Bank (PDB) are directly solved from experimental densities with a resolution higher than 4 Å. However, medium-resolution (5–10 Å) cryo-EM maps are also increasingly deposited in the Electron Microscopy Data Bank (EMDB), many of which have associated atomic models of varying quality. It is difficult to solve atomic structures from such medium-resolution maps directly, so modeling approaches often rely on the indirect fitting of known templates. Therefore, we hypothesize that early structural interpretations of medium-resolution cryo-EM maps in the EMDB could benefit from potentially more reliable AlphaFold models derived later after more structural templates become available in the PDB. To study the utility of AlphaFold-predicted models, we conducted systematic mapping between the PDB and AlphaFold DB for structures derived from medium-resolution cryo-EM density maps. A dataset of 918 nonredundant pairs of structural segments was established. Using MolProbity, a structural validation method, we observed a significant difference in the distributions of MolProbity scores between paired structural segments in the PDB and AlphaFold DB. The structural segments in the AlphaFold DB exhibit a unimodal distribution, with an average of 0.96 MolProbity score, which is better than the average (1.98) of their corresponding segments in the PDB. The MolProbity scores of structural segments in the PDB vary significantly more than those in the AlphaFold DB and exhibit a bimodal distribution with a longer tail, indicating a wider range of model quality owing to the diverse structure fitting and refinement strategies used in the past.

Keywords: Protein · Protein Structure · Structure Validation

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2024
W. Peng et al. (Eds.): ISBRA 2024, LNBI 14956, pp. 52–63, 2024.
https://doi.org/10.1007/978-981-97-5087-0_5

1 Introduction

As new experimental and computational techniques were introduced to structural biology, vast knowledge about molecular structures has accumulated over the last 20 years. The available models and data have been freely made available in interconnected databases. Before the "resolution revolution" of cryo-electron microscopy (cryo-EM) in 2014, atomic structures in the Protein Data Bank (PDB) were predominantly determined by X-ray crystallography and NMR. In contrast, as of April 2024, 19,708 entries in the PDB were molecular structures determined using 3D electron microscopy density maps. The rapid accumulation of atomic structures of proteins in recent decades has made it possible to computationally predict the atomic structure directly from its amino acid sequence. AlphaFold is one of the top-performing methods, and it has proven to yield high accuracy at the CASP14 competition [1]. The AlphaFold Protein Structure Database (AlphaFold DB) was recently released, providing access to more than 200 million protein structural models predicted using the AlphaFold2 method [1, 2]. The database covers most of the protein sequences in the Universal Protein Knowledgebase (UniProtKB), which is a hub for protein sequences and functional information [3]. A model in the AlphaFold DB was identified using the UniProt accession ID.

Among the numerous structure validation methods, MolProbity is one of the methods used in PDB validation reports during the deposition process. MolProbity—an all-atom validation method for proteins and nucleic acids—provides scores in multiple categories, such as clashing, Ramachandran outlier, side chain outlier, and RNA backbone percentiles [4]. Although MolProbity was initially constructed with features specifically tailored for X-ray crystallography, it is also suitable and used for models derived from cryo-EM, neutron, NMR, and computationally predicted models [5]. Here, we used MolProbity to evaluate the local structural quality of the protein models deposited in the PDB and AlphaFold DB.

The rapid increase in the number of atomic structures in the PDB is caused by the increasing number of associated cryo-EM density maps with 2-4 Å resolution that can be used to determine the structure directly [6–8]. However, for medium-resolution (5-10 Å) cryo-EM maps, many details of the protein structure, such as the backbone and side chains, are not well resolved, which makes it difficult to produce an atomic model. Despite this challenge, a significant number of atomic models have been constructed with medium-resolution density maps, many of which are generated through template fitting. Template structures were first identified from existing atomic structures that share sufficient similarity with the target protein structure. The template structures were then iteratively modified to optimize the agreement between the model and density map. Therefore, the accuracy of the templates and the methods used in fitting and refinement are important for producing accurate models. Improving the reliability and accuracy of these medium-resolution cryo-EM-derived models remains challenging.

It has been demonstrated at CASP14 that AlphaFold2, a deep learning method, produces surprisingly accurate structural models in both the FM (free modeling) and TBM (template-based modeling) target groups [9, 10]. Beyond AlphaFold2, a few fast prediction methods have been introduced for "orphan proteins", which have few homologs in a protein family [11]. AlphaFold2, as well as RoseTTAFold, relies on multiple sequence alignment (MSA) of protein sequences within the same family [1, 12], which is of limited

utility for orphan proteins or families with highly diverse sequences [13, 14]. RGN2 is an MSA-free method that uses a protein language model that may outperform AlphaFold2 in the case of orphan proteins [15]. EMBER2 uses a pretrained protein language model to predict interresidue distances [14]. Such language models can provide faster predictions because MSA is often time-consuming. However, none of the available alternative prediction models rival the scope and ease of access of the AlphaFold DB, with more than 200 million predicted models.

The release of the AlphaFold DB raises the question of whether AlphaFold2-predicted models are more accurate than structures already in the PDB obtained through the fitting and refinement of medium-resolution density maps. The answer to this question can inform improvements to existing methods for interpreting medium-resolution maps in atomic detail. For such a comparison, however, a dataset must first be established to pair PDB structures with AlphaFold2-predicted models. Although the Electron Microscopy Data Bank (EMDB), PDB, and AlphaFold DB are linked, the related PDB and AlphaFold DB models often do not correspond to the exact same amino acid sequence [2, 16, 17]. Often, entries are linked to contain similar but nonidentical sequences or to share specific segments that are identical but do not cover the entire length of the chains. In the present paper, we performed systematic mapping between the PDB and AlphaFold DB to identify a nonredundant set of 918 pairs of structural segments. Each pair exhibited identical sequences for the two structural segments, one from the AlphaFold DB and the other from PDB entries derived from medium-resolution cryo-EM density maps. We then used MolProbity to quantify the structural quality of the corresponding models from the PDB and AlphaFold DB.

2 Methods

We developed a screening process to match PDB structures only with AlphaFold DB models, which share the exact same sequence. The EMDB search engine was used to obtain a list of all cryo-EM entries that satisfied three criteria: (1) the resolution of the density map was between 5 and 10 Å; (2) the density map had an associated atomic model deposited in the PDB; and (3) the EMDB entry (EMD) was cross-referenced to the AlphaFold DB. The resulting list provides initial information to relate the EMDB, PDB, and AlphaFold DB, but the cross-reference does not indicate specific matching segments in structural models. The PDB entries were then matched to their corresponding AlphaFold DB entries by using the mapping tool Structure Integration with Function, Taxonomy, and Sequences (SIFTS) [18]. SIFTS provides a summary of residue-level mapping between entries in PDB and the UniProt Knowledgebase (UniProtKB) through UniProt accession. In summary, a segment of the PDB chain is matched to a segment of the AlphaFold DB model using SIFTS to identify the start and ending residues if they share high sequence similarity [18]. Any pairs of entries containing “None” as the start/end residue were eliminated. The resulting list of pairs after SIFTS mapping was further processed to eliminate those pairs with nonidentical sequence segments. This ensured that pairs with similar but not identical sequences were removed. The 3,802 resulting pairs were obtained in this way for matched structural segments between the PDB and AlphaFold DB.

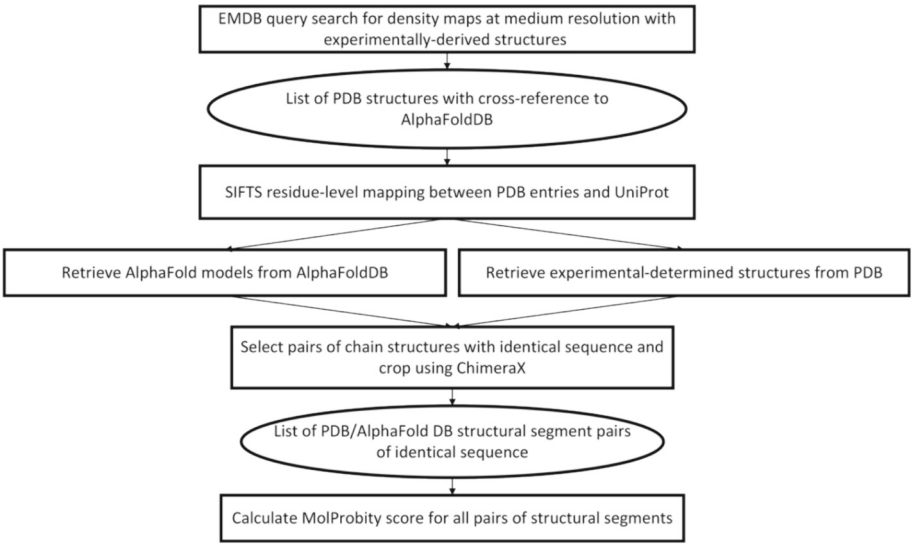

Fig. 1. Workflow for obtaining nonredundant matched structural segments between the PDB and AlphaFold DB for those structures derived from medium-resolution cryo-EM density maps.

The PDB ID and UniProt accession were used to retrieve structures/models from the PDB and AlphaFold DB, respectively, for the 3,802 pairs of structural segments. For each pair, the matched structural segments were cropped from their entire structure using ChimeraX, which is based on the beginning/ending position in the mapped segments [19]. Because a PDB entry often contains multiple copies of the same chain sequence for a cryo-EM-derived atomic structure, the dataset contains redundant pairs that correspond to the same sequence. To remove redundancy, the pair in which the PDB structural segment had the lowest MolProbity score was kept among pairs with the same sequence. This process yielded 918 nonredundant pairs of mapped structural segments, one derived from a medium-resolution cryo-EM density map and the other predicted using AlphaFold2.

Many methods exist for comparing the accuracy of a model with respect to a reference structure. In CASP, predicted models are compared with experimentally determined structures that are often obtained from high-resolution X-ray crystallography data. Models are evaluated using Global Distance Test (GDT) methods, such as GDT-TS and GDT-HA [21, 22]. Although variants of GDT methods require superposition of two models, superposition-independent methods, including the CAD (residue contact area difference) and lDDT (local distance difference) methods, have been used [23, 24]. AlphaFold2 uses the confidence measure of a predicted model using pLDDT [1]. Despite various existing measures for structural comparison, the goal of the current study is beyond measuring the difference between the two models. Because both the predicted AlphaFold2 models and PDB models, which are derived from medium-resolution EM density maps, may contain errors, the goal is to estimate the quality of each model. The quality of a model is partially reflected by local characteristics such as steric clashes,

rotamer distributions, and backbone torsion angle distributions. The MolProbity score is a quick measure that combines the clash score, rotamer outlier, and Ramachandran outlier. In general, a lower MolProbity score reflects a better overall quality of a structure. MolProbity all-atom validation was performed through the Phenix software package using the command phenix.molprobity [5, 20].

3 Results and Discussion

Table 1. A set of 20 matched pairs of structural segments from the PDB/AlphaFold DB. Left to right: PDB entry and chain identifier, corresponding identifier of the density map in EMDB, resolution (res.) of the cryo-EM density map, UniProt accession of AlphaFold models, matched structural segment (number of Cα atoms, start-end residue of the matched structural segment in the PDB chain and AlphaFold model, respectively), and MolProbity scores of the structural segments.

PDB	EMD	Res.	UniProt	Sequence segment			MolProbity	
				Len.	PDB	UniProt	PDB	AF
2j28_9[c]	1261	9.5	P0AGD7	430	2-431	2-431	4.59	1.26
4adx_G	2012	6.6	O27647	78	1-78	1-78	1.65	0.6
4cg7_C	2510	6.9	P60467	36	61-96	61-96	2.21	2.91
5fil_T	3170	7.1	P13619	174	1-174	76-249	0.82	0.8
5fl2_K	3206	6.2	P31473	106	332-437	332-437	1.53	0.83
5k0y_a[b]	8190	5.8	G1TG89	129	2-130	2-130	0.95	1.21
5gpn_6	9534	5.4	P04038	73	1-73	2-74	1	1.1
5gpn_D	9534	5.4	P00125	241	1-241	85-325	2.49	0.7
5vfu_B	8668	5.8	P62191	348	93-440	93-440	2.96	1.19
6eny_B	3906	5.8	P61769	99	1-99	21-119	2.6	0.52
6fxc_AV	3637	6.76	Q2YXJ2	58	3-60	3-60	1.76	2.12
6gsm_N	0057	5.15	Q6CJK0	150	2-151	2-151	2.31	0.89
6i7o_AH	4427	5.3	P04456	120	23-142	23-142	1.72	0.97
6qc9_D5	4501	5.7	O78756	606	1-606	1-606	1.42	1.46
7blo_G	12221	9.5	O60493	155	4-158	4-158	1.1	0.95
7jti_C	22474	7.4	P12661	300	2-301	633-932	1.62	0.9
7lrg_G	23497	6.1	Q14896	94	358-451	358-451	3.59	2.53
8b5r_X	15861	6.1	Q14CS0	11	215-225	215-225	1.26	1.08
8f26_a[d]	28808	9.7	P54274	27	28-54	404-430	0.5	0.99
8th8_G[a]	41251	7.4	Q24CL2	345	1-345	1-345	2.79	1.53

3.1 The Dataset of Matched Structural Segments in the PDB/AlphaFold DB

We processed all entries in the PDB as of January 20, 2024, that were obtained from medium-resolution cryo-EM density maps and obtained their exact sequence matches in the AlphaFold DB (see Sect. 2). The resulting 918 nonredundant pairs of matched structural segments were identified, in which each pair consisted of a full/partial chain in a PDB entry and a full/partial chain with an identical sequence in an AlphaFold DB entry. A sample of 20 pairs, randomly selected from the 918 pairs, is shown in Table 1. As an example (the second from the last row in Table 1), amino acids 28-46 in chain A of PDB ID 8f26 corresponding to EMD 28808 have sequences identical to those of amino acids 404-430 of the AlphaFold model P54274 (UniProt ID).

Among the nonredundant pairs of matched structural segments, the majority (505 pairs) had fewer than 150 amino acids in the matched segment (Table 2). However, 156 pairs had more than 300 amino acids in the matched segment, representing larger models predicted using AlphaFold2.

3.2 Evaluation of Matched Structural Segments Using MolProbity

The 918 nonredundant pairs of structural segments from the PDB and AlphaFold DB represent two corresponding sets of structural models with identical protein sequences obtained by two different approaches, one using structural model building from a medium-resolution cryo-EM density map and the other using AlphaFold2 prediction. We evaluated the structural quality of the 918 segments in each set using the MolProbity tool, which assesses the overall local quality of the stereochemistry of the atomic model. The MolProbity score is a log-weighted function of individual scores for steric clash, Ramachandran (backbone quality), and rotamer (sidechain quality) so that the combined value represents the spatial resolution number at which those scores would be expected [4, 5].

Table 2. Overall MolProbity scores for 918 matched pairs of structural segments in the PDB/AlphaFold DB. The 918 pairs were divided into three subgroups based on the number of Cα atoms in each matched segment. Subgroup 1: Chains with fewer than 150 Cα atoms. Subgroup 2: Chain with a number of Cα atoms between 150 and 300. Subgroup 3: Chains with more than 300 Cα atoms. Left to right: The number of pairs in each group, the average number of Cα atoms per chain, and the average MolProbity scores of the matched structural segment in the PDB and AlphaFold DB, respectively.

	Num.	Avg. Cα	Avg. MolProbity	
			PDB	AlphaFold DB
Subgroup 1	505	95	1.98	0.94
Subgroup 2	257	209	1.95	0.91
Subgroup 3	156	468	2.04	1.10
All	918	190	1.98	0.96

As shown in Table 2, for the dataset of 918 pairs, the average MolProbity score of the structural segments in the AlphaFold DB (0.96) was significantly better than that of their corresponding experimentally derived structural segments with matching sequences (1.98). This pattern was the same in the three subgroups with different length bins (Table 2). As shown in Fig. 2, most AlphaFold DB models exhibited a score between 0.5 and 1.0, with the worst MolProbity score being 2.91 in the AlphaFold DB dataset. This unimodal distribution of scores reflects the high local quality of the AlphaFold DB models, which were produced through deep learning of PDB structures solved directly with high-resolution experimental techniques. However, the MolProbity scores varied considerably for the experimentally derived structural segments, ranging from 0.5 to 4.64, a lower resolution of the maps that informed these models. The average difference between the scores of the AlphaFold-predicted and experimental structural segments was 1.07.

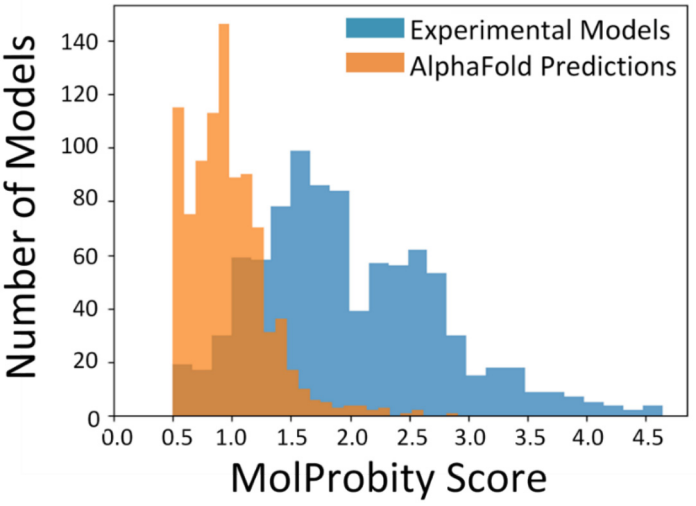

Fig. 2. Distribution of MolProbity scores for 918 structural segments in each PDB and AlphaFold DB set.

Although, on average, the MolProbity scores of the AlphaFold-predicted models were lower than the corresponding PDB scores, the distributions overlap, and a small number, 66 pairs, actually exhibited lower MolProbity scores for experimentally derived structures. Such high-quality models were mostly deposited in PDB within the last 10 years, as 60 of them were deposited after 2015, and the results reflected better computational local refinement strategies available as well as the better quality of cryo-EM maps after the DDD-detector revolution of 2014, which benefited not only high-resolution maps but also maps at medium resolution [25].

3.3 Differences in Local Quality Between the Four Models Derived from Medium-Resolution Cryo-EM Maps and Those Predicted with AlphaFold

The MolProbity evaluation (Fig. 2) revealed an overall difference between the two distributions, one obtained through medium-resolution cryo-EM maps and the other from the AlphaFold DB. A lower MolProbity score does not necessarily indicate a better fit with the density because the score reflects only local characteristics such as steric clashes, Ramachandran violations (backbone quality), and sidechain rotamer quality. It is possible for a low MolProbity scoring model to exhibit a global problem, such as a relative orientation mismatch between two domains or two helices. Additionally, the score does not reflect similarity with the cryo-EM map. To better understand the nature of the differences in the distributions, we visually explored four cases with diverse MolProbity scoring results. Figure 3A shows the mapped structural segments aligned in ChimeraX for visual inspection. In this case, the fragment covers the entire G chain of 8th8 (PDB ID), a kinase domain of 345 amino acids in length. The AlphaFold DB model Q24CL2 (UniProt accession) had a much better MolProbity score of 1.53 versus 2.79 for the PDB chain. The two structural models aligned very well, particularly for the two backbones. The similarity of the two backbones suggested that the AlphaFold2-predicted model may fit as well as the G chain in the 7.4 Å resolution cryo-EM density maps (although we did not fit the model into the density map). The difference in the MolProbity score suggested that the AlphaFold2 model could be a good candidate contributing to any improvement in the G chain deposited in the PDB in July 2023.

We also randomly selected one of the 66 curated cases in which the mapped segment in the PDB had a better MolProbity score than that of the AlphaFold DB model (Fig. 3B). In this case, the "a" chain (author annotated) of PDB ID 5k0y, a ribosomal protein, uS8, has the entire chain of 129 amino acids in the matched fragment. Its MolProbity score is 0.95, which is better than the 1.21 of the AlphaFold DB model. The cryo-EM density map that was used to derive the a chain has 5.8 Å resolution, at which secondary structures such as helices and β-sheets are often well visible. It is also a ribosomal protein, and ribosome structures are among the most studied structures using cryo-EM methods. Atomic structures that are derived from medium-resolution cryo-EM density maps are predominantly obtained through a fitting process in which a template structure is modified locally by how well the structure agrees with the density map. Therefore, a good template structure and a density map with good visibility of secondary structure elements may produce an atomic structure with good accuracy. The minor difference in scores in such cases reflects how aggressively the modeling process improved the local quality of the refined structure.

We also inspected case 2j28 chain 9 (PDB ID) (pink in Fig. 3C), which had a large MolProbity score of 4.59. The mapped structural segment covers chain 9, which has 430 amino acids. This chain, noted as a signal recognition particle 54, is a component of the ribosome. The structural segment in the PDB model significantly disagrees with that in the newer AlphaFold2-predicted model because they cannot be aligned well in ChimeraX (pink vs cyan in Fig. 3C). The disagreement in this case was significantly greater than that in the two previous cases. The main disagreement was the relatively different positioning of the two domains, which is a global property not necessarily

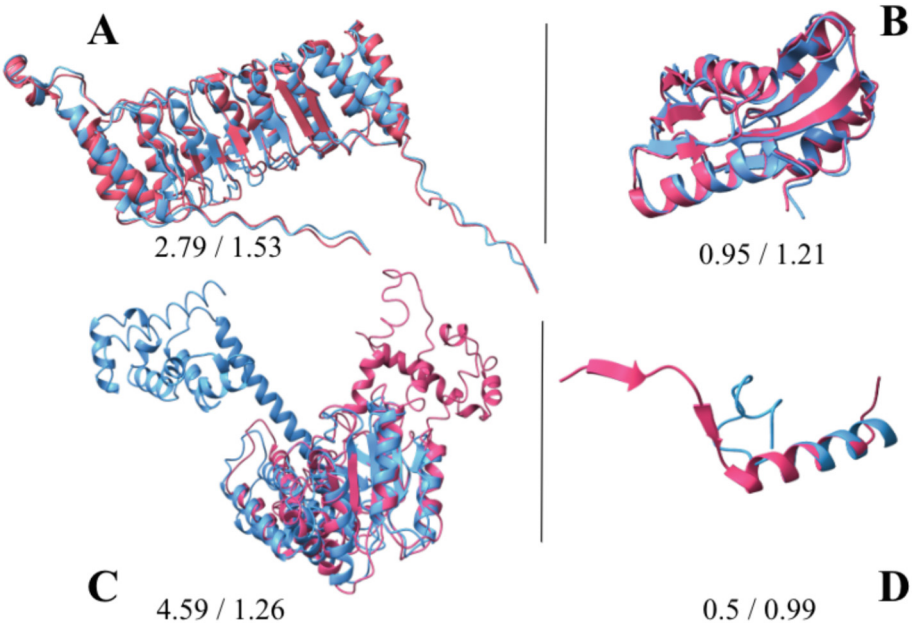

Fig. 3. Examples of mapped pairs of structural segments. Four randomly selected pairs of structural segments in the PDB (pink ribbon) and AlphaFold DB (cyan ribbon) were superimposed in ChimeraX. The structural segments with PDB ID and the AlphaFold DB UniProt accession are provided in (A) 8th8, chain G/Q24CL2, (B) 5k0y, chain a/G1TG89, (C) 2j28, chain 9/P0AGD7, (D) 8f26, chain a/P54274. The MolProbity score for the PDB/AlphaFold DB structural segment is noted. (Color figure online)

reflected in the MolProbity score. To answer the question of which of the two structural segments is more accurate, further studies are needed to compare their fit to the cryo-EM map and to consider other known ribosome structures. However, the MolProbity score of 4.59 in this case reflects the relatively low (9.5 Å) resolution of the relatively old (2006) cryo-EM density map.

Among the 918 matched pairs, some involved relatively short segments, such as the 8f26 chain a/P54274 pair, comprising only 27 amino acids (Fig. 3D). In this case, the sequence of chain a in the PDB structure only shared an identical sequence with P54217 (UniProt accession) in this short segment. Although such short pairs exist in the dataset pool, they are likely not as valuable as longer segments for future studies because of the lack of structural complexity.

4 Conclusion

As a byproduct of the growth of cryo-EM, the number of medium-resolution cryo-EM density maps deposited in the EMDB has increased steadily over the past 10 years [25]. Unlike high-resolution density maps, which can be solved directly, medium-resolution maps are less detailed, which makes it challenging to derive atomic models. Therefore,

atomic models are predominantly obtained by fitting template structures in medium-resolution density maps. During the fitting process, template structures are refined to maximize the agreement between the structural model and the density map, but the local quality of the model is often less important, especially in earlier models. With the availability of AlphaFold2, we now have a tool at our disposal to improve the modeling accuracy achieved in earlier medium-resolution density maps. One important step toward developing an improved method is to understand which discrepancies exist between AlphaFold2-predicted models and early models in the PDB that were originally derived from and deposited with medium-resolution density maps.

We conducted systematic matching between the AlphaFold DB and the PDB for the medium-resolution-derived models. Our evaluation of the dataset using MolProbity showed that there was a systematic difference in the local quality among the matched structural segments, although the PDB and AlphaFold DB distributions overlapped, and there were notable outliers. The unimodal AlphaFold DB score distribution reflects the high resolution of the underlying experimental data trickled into the AlphaFold2 models (AlphaFold 2 is a deep learning method trained from PDB structures that are most often obtained from high-resolution experimental data). On the other hand, the MolProbity scores for the medium-resolution PDB models (Fig. 2) show a much wider range and a bimodal distribution with a long tail. Although the MolProbity scores do not have a unit, they are normalized to reflect the equivalent numeric spatial resolution (in Å). Our results confirm that models derived from medium-resolution cryo-EM maps have a reduced local quality, with a score that (on average) reflects the greater spatial resolution (in Å) of the cryo-EM datasets. We also observed two distinct classes of cryo-EM-derived models in the distribution, which is likely because of differences in modeling approaches (rigid body or flexible fitting). The structural causes of this bimodal distribution will be analyzed in future work.

A lower MolProbity score does not necessarily mean a better model. Further studies are needed to consider the agreement between the model and the density map as well as any other information available about the structure. However, we argue that a predicted AlphaFold DB model with a lower MolProbity score has the potential to improve the structural accuracy of an accepted model in the PDB. Our dataset of 918 matched pairs, which is available upon request, is potentially useful for the development of a more accurate modeling method for interpreting medium-resolution density maps.

Acknowledgments. This work was supported by NIH Grant No. R01-GM062968 and the ODU Batten Endowment to W.W.

Disclosure of Interests. None.

References

1. Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., et al.: Highly accurate protein structure prediction with AlphaFold. Nature **596**(7873), 583–589 (2021). https://doi.org/10.1038/s41586-021-03819-2

2. Varadi, M., Anyango, S., Deshpande, M., Nair, S., Natassia, C., Yordanova, G., et al.: AlphaFold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. Nucleic Acids Res. **50**(D1), D439–D444 (2022). https://doi.org/10.1093/nar/gkab1061
3. Consortium, U.: UniProt: the universal protein knowledgebase in 2023. Nucleic Acids Res. **51**(D1), D523–D31 (2023). https://doi.org/10.1093/nar/gkac1052
4. Chen, V.B., Arendall, W.B., Headd, J.J., Keedy, D.A., Immormino, R.M., Kapral, G.J., et al.: MolProbity: all-atom structure validation for macromolecular crystallography. Acta Crystallogr D Biol Crystallogr **66**(Pt 1), 12–21 (2010). https://doi.org/10.1107/s0907444909042073
5. Williams, C.J., Headd, J.J., Moriarty, N.W., Prisant, M.G., Videau, L.L., Deis, L.N., et al.: MolProbity: more and better reference data for improved all-atom structure validation. Protein Sci. **27**(1), 293–315 (2018). https://doi.org/10.1002/pro.3330
6. Chari, A., Stark, H.: Prospects and limitations of high-resolution single-particle cryo-electron microscopy. Ann. Rev. Biophys. **52**, 391–411 (2023). https://doi.org/10.1146/annurev-biophys-111622-091300
7. Yip, K.M., Fischer, N., Paknia, E., Chari, A., Stark, H.: Atomic-resolution protein structure determination by cryo-EM. Nature **587**(7832), 157–161 (2020). https://doi.org/10.1038/s41586-020-2833-4
8. Vilas, J.L., Carazo, J.M., Sorzano, C.O.S.: Emerging themes in CryoEM–Single particle analysis image processing. Chem. Rev. **122**(17), 13915–13951 (2022). https://doi.org/10.1021/acs.chemrev.1c00850
9. Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., et al.: Applying and improving AlphaFold at CASP14. Proteins: Struct., Funct., Bioinf. **89**(12), 1711–21 (2021). https://doi.org/10.1002/prot.26257
10. Kryshtafovych, A., Schwede, T., Topf, M., Fidelis, K., Moult, J.: Critical assessment of methods of protein structure prediction (CASP)-Round XIV. Proteins **89**(12), 1607–1617 (2021). https://doi.org/10.1002/prot.26237
11. Bertoline, L.M.F., Lima, A.N., Krieger, J.E., Teixeira, S.K.: Before and after AlphaFold2: an over-view of protein structure prediction. Front. Bioinf. **3**, 1120370 (2023)
12. Baek, M., DiMaio, F., Anishchenko, I., Dauparas, J., Ovchinnikov, S., Lee, G.R., et al.: Accurate prediction of protein structures and interactions using a three-track neural network. Science **373**(6557), 871–876 (2021). https://doi.org/10.1126/science.abj8754
13. Michaud, J.M., Madani, A., Fraser, J.S.: A language model beats alphafold2 on orphans. Nat. Biotechnol. **40**(11), 1576–1577 (2022). https://doi.org/10.1038/s41587-022-01466-0
14. Weissenow, K., Heinzinger, M., Rost, B.: Protein language-model embeddings for fast, accurate, and alignment-free protein structure prediction. Structure **30**(8), 1169–77.e4 (2022). https://doi.org/10.1016/j.str.2022.05.001
15. Chowdhury, R., Bouatta, N., Biswas, S., Floristean, C., Kharkar, A., Roy, K., et al.: Single-sequence protein structure prediction using a language model and deep learning. Nat. Biotechnol. **40**(11), 1617–1623 (2022). https://doi.org/10.1038/s41587-022-01432-w
16. Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., et al.: The protein data bank. Nucleic Acids Res. **28**(1), 235–242 (2000). https://doi.org/10.1093/nar/28.1.235
17. The ww PDBC: EMDB—the electron microscopy data bank. Nucleic Acids Res. **52**(D1), D456–D65 (2024). https://doi.org/10.1093/nar/gkad1019
18. Dana, J.M., Gutmanas, A., Tyagi, N., Qi, G., O'Donovan, C., Martin, M., et al.: SIFTS: updated structure integration with function, taxonomy and sequences resource allows 40-fold increase in coverage of structure-based annotations for proteins. Nucleic Acids Res. **47**(D1), D482–D489 (2018). https://doi.org/10.1093/nar/gky1114

19. Pettersen, E.F., Goddard, T.D., Huang, C.C., Meng, E.C., Couch, G.S., Croll, T.I., et al.: UCSF Chime-raX: structure visualization for researchers, educators, and developers. Protein Sci. **30**(1), 70–82 (2021)
20. Liebschner, D., Afonine, P.V., Baker, M.L., Bunkóczi, G., Chen, V.B., Croll, T.I., et al.: Macromolecular structure determination using X-rays, neutrons and electrons: recent developments in Phenix. Acta Crystallogr. D Struct. Biol. **75**(Pt 10), 861–877 (2019). https://doi.org/10.1107/s2059798319011471
21. Kryshtafovych, A., Monastyrskyy, B., Fidelis, K.: CASP prediction center infrastructure and evaluation measures in CASP10 and CASP ROLL. Proteins: Struct., Funct., Bioinf. **82**(S2), 7–13 (2014). https://doi.org/10.1002/prot.24399
22. Zemla, A.: LGA: a method for finding 3D similarities in protein structures. Nucleic Acids Res. **31**(13), 3370–3374 (2003). https://doi.org/10.1093/nar/gkg571
23. Olechnovič, K., Kulberkytė, E., Venclovas, C.: CAD-score: a new contact area difference-based function for evaluation of protein structural models. Proteins **81**(1), 149–162 (2013). https://doi.org/10.1002/prot.24172
24. Mariani, V., Biasini, M., Barbato, A., Schwede, T.: LDDT: a local superposition-free score for comparing protein structures and models using distance difference tests. Bioinformatics **29**(21), 2722–2728 (2013). https://doi.org/10.1093/bioinformatics/btt473
25. Kühlbrandt, W.: The resolution revolution. Science **343**(6178), 1443–1444 (2014). https://doi.org/10.1126/science.1251652

PmmNDD: Predicting the Pathogenicity of Missense Mutations in Neurodegenerative Diseases via Ensemble Learning

Xijian Li[1], Ying Huang[2], Runxuan Tang[1], Guangcheng Xiao[1], Xiaochuan Chen[1], Ruilin He[1], Zhaolei Zhang[3,4,5], Jiana Luo[1], Yanjie Wei[2(✉)], Yijun Mao[1(✉)], and Huiling Zhang[1(✉)]

[1] College of Mathematics and Information, South China Agricultural University, Guangzhou 510642, China
{maoyijun,hl.zhang}@scau.edu.cn
[2] Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China
yj.wei@siat.ac.cn
[3] Department of Molecular Genetics, University of Toronto, Toronto, ON M1C1A4, Canada
[4] Donnelly Centre for Cellular and Biomolecular Research, University of Toronto, Toronto, ON M1C1A4, Canada
[5] Department of Computer Science, University of Toronto, Toronto, ON M1C1A4, Canada

Abstract. Accurately distinguishing between pathogenic and benign mutations continues to pose a significant challenge in the clinical genetic testing of patients with neurodegenerative diseases (NDDs). In theory, computational methods have the potential to facilitate the interpretation of genetic variants in NDDs on a large scale. However, individual tools often exhibit disagreements, biases, and variations in quality. As a result, the predictions derived from them are considered insufficiently reliable. In this study, we developed PmmNDD, an ensemble method for predicting pathogenicity of missense variants in NDDs. PmmNDD integrated the prediction scores from other methods along with amino acid characteristics as features, and was constructed with the categorical boosting (CatBoost) model. The stability and generalization ability of PmmNDD were validated through leave-one-gene-out cross-validation and independent test. We also demonstrated PmmNDD's superior performance over 20 other methods. Furthermore, we provided pre-computed PmmNDD scores for all possible NDDs missense variants to facilitate the identification of pathogenic variants in the sea of rare variants discovered as sequencing studies expand in scale. In summary, our work suggests that models from ensemble learning can provide valuable independent evidence for NDD mutation interpretation that will be widely useful in research and clinical scenarios.

Keywords: missense mutation · neurodegenerative diseases · ensemble learning · mutation interpretation

X. Li and Y. Huang—The authors contribute equally to this work.

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2024
W. Peng et al. (Eds.): ISBRA 2024, LNBI 14956, pp. 64–75, 2024.
https://doi.org/10.1007/978-981-97-5087-0_6

1 Introduction

Neurodegenerative diseases encompass a diverse group of conditions characterized by the progressive degeneration of nerve cells, resulting in debilitating symptoms such as cognitive decline, motor dysfunction, and impaired memory. Conditions like Alzheimer's disease, Parkinson's disease, and amyotrophic lateral sclerosis (ALS) are among the most well-known neurodegenerative disorders. These diseases pose immense challenges to patients, caregivers, and healthcare systems worldwide, necessitating a deeper understanding of their underlying mechanisms and the development of novel therapies. Research efforts focused on identifying pathogenic variants [1], unraveling the variant pathophysiology [2], and exploring potential diagnosis options [3, 4] offer hope for improved management and outcomes for individuals affected by neurodegenerative diseases.

Currently, variants are commonly assessed using the American College of Medical Genetics and Genomics (ACMG) criteria or similar frameworks. The application of the ACMG criteria often results in the classification of VUS, providing limited clinical utility. The complexity of categorizing missense variants is exacerbated by the prevalence of rare missense variants in genomes. Although multiplexed assays of variant effect (MAVEs) offer a systematic approach to measuring missense mutation effects and predicting clinical outcomes accurately [5], a comprehensive proteome-wide assessment of variant pathogenicity remains elusive due to the resource-intensive nature of MAVE experiments. Therefore, there is a pressing need for additional methods to improve the classification of missense variants. Computational methods leverage patterns in biological data to predict the pathogenicity of unannotated variants, bridging the gap in variant interpretation. There have been many emerging methods in the field of missense mutation prediction, which can be broadly categorized into three main types, including conservation methods, machine learning methods, and deep learning methods. Conservation methods predict the deleteriousness of mutations in coding regions by quantifying the evolutionary constraints on affected residues, and representative methods in this category are MutationAssessor [6] and LIST-S2 [7]. Representative machine learning-based methods include PolyPhen [8], VEST4 [9], MetaSVM [10], BayesDel [11], ClinPred [12], FATHMM_MKL [13], CADD [14], REVEL [15], DEOGEN2 [16]. Deep learning models can learn high-order dependencies between amino acids from protein sequences and have shown good performance, such as DANN [17], PrimateAI [18], gMVP [19]. The significant success in predicting protein 3D structures has ushered in a new era for predicting the pathogenicity of missense mutations. AlphaMissense [20], ESM1b [21] has transferred protein 3D structure prediction models to pathogenicity prediction. Recently, with the continuous improvement of AlphaFold database [22, 23], methods based on the predicted structures of AlphaFold have been gradually developed, AlphaScore [24], primateAI-3D [25], and others.

However, individual tools frequently display discrepancies, biases, and inconsistencies in quality, leading to predictions that are deemed unreliably accurate. To address this issue, we have introduced PmmNDD, a novel ensemble learning approach designed to predict the pathogenicity of missense variants in neurodegenerative diseases (NDDs).

PmmNDD incorporated prediction scores from other methods and amino acid characteristics as features, and was constructed using the CatBoost model. The leave-one-gene-out cross-validation and independent test of PmmNDD show comparable prediction performance with AUCs around 0.948 and AUPRs about 0.931, indicating the stability and generalization capabilities of PmmNDD. The results also demonstrated that the mutation-function relationship learned by our ensemble model from non-NDD genes yielded comparable results to models trained with NDD genes. Moreover, PmmNDD exhibited superior performance over 20 other methods. Additionally, we have provided pre-computed PmmNDD scores for all potential NDDs missense variants to aid in identifying pathogenic variants amidst the multitude of rare variants discovered as sequencing studies expand. In conclusion, our study suggests that ensemble learning models can offer valuable independent evidence for interpreting NDD mutations, which will be widely applicable in research and clinical settings.

2 Materials and Methods

2.1 Overall Workflow

The overall workflow of PmmNDD is shown in Fig. 1. The proposed framework can be summarized as the following four main steps: (i) dataset construction; (ii) model construction; (iii) model analysis and validation; (iv) online database. These steps are further elaborated on in the following sections.

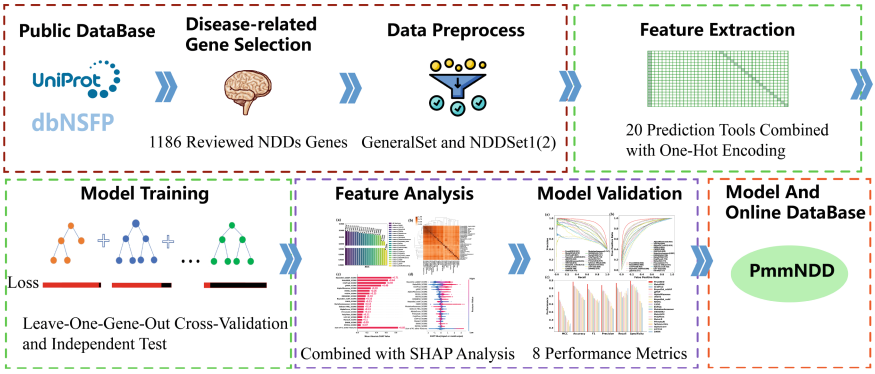

Fig. 1. The overall workflow of PmmNDD.

2.2 Dataset Construction

We extracted genetic and clinical mutation data from a total of 20,516 human protein-coding genes, as curated in UniProt [26]. Among these, 10,538 genes (DataSet1) have mutations associated with clinical diseases as documented in ClinVar [27] (positive labels are clinical significance recorded as “pathogenic” or “likely pathogenic”; negative

labels are clinical significance annotated as “benign” or “likely benign”). We conducted a search for genes related to NDDs such as Alzheimer, Parkinson, amyotrophic lateral sclerosis, metabolic disorders of the cerebrospinal fluid, Gaucher disease, Huntington’s disease, cerebellar atrophy, multiple sclerosis, primary lateral sclerosis, spinal muscular atrophy, cerebral ischemia, spastic paraplegia, and myasthenia gravis in the UniProt database. After excluding unreviewed genes, we retained a total of 1186 genes (NDDSet) associated with NDDs. 595 genes in DataSet3 have prediction scores returned from the methods described in Sect. 2.3, designated as NDDSet1. 461 genes in NDDSet1 have positive or negative labels recorded in ClinVar, denoted a s NDDSet2. Through removing NDDSet and culling the redundant sequences, 5359 genes in DataSet1 were retained, represented as GeneralSet. There are 93721 and 14668 labeled mutations in GeneralSet and NDDSet2 (NDDSet1), respectively.

2.3 Feature Extraction

This study utilized two types of features, including prediction scores from other methods and one-hot encoding of amino acids before and after site mutations. We obtained missense mutation prediction results from 43 methods from their official websites and the dbNSFP database [28]. Through Jaccard similarity coefficient analysis, we eliminated methods with high similarity and limited result outputs, and eventually selected prediction scores from 20 methods as features for the ensemble model. These methods included MutationAssessor, VEST4, MetaSVM, MetaLR, MetaRNN, LIST-S2, BayesDel_addAF, BayesDel_noAF, PrimateAI, ClinPred, FATHMM_MKL, DANN, PolyPhen, CADD, REVEL, DEOGEN2, gMVP, AlphaMissense, AlphaScore and ESM1b. The proposed method also incorporates one-hot encoding features aimed at accurately capturing the changes in amino acid states before and after mutation. Each amino acid residue is represented by a 20-dimensional vector, where only one position out of the 20 elements is 1, and the rest are 0. Since there are only 20 types of amino acids in nature, using 20 of these vectors (with the position of 1 being different) can represent all amino acid residues. Specifically, for each missense mutation, there are 20 dimensions of one-hot encoding before and after the mutation, generating a 40-dimensional vector.

Overall, we utilize the predictive scores from 20 prediction methods and 40-dimensional one-hot encoding to form the feature vector, aiming to better explore and understand the impact of missense mutation pathogenicity in NDDs.

2.4 PmmNDD Model Training

In this study, we employed seven different machine learning algorithms to construct mutation pathogenicity prediction models. These machine learning algorithms include CatBoost, XGBoost, LightGBM, random forest (RF), support vector machine (SVM), linear regression (LR), and AdaBoost. Additionally, by utilizing parameter optimization strategies such as grid search, Bayesian optimization, and manual parameter tuning, we meticulously fine-tuned and optimized the model’s hyperparameters aiming to achieve the best predictive performance.

To verify the stability and generalization capability of PmmNDD, we conducted two sets of comparative experiments: the leave-one-gene-out cross-validation and the independent test. In the leave-one-gene-out cross-validation, each time labeled mutations from only one gene in NDDSet2 is kept as the validation data, while the labeled mutations from rest genes in NDDSet2 and all genes in GeneralSet are used as the training data. By using different training and validation sets each time, we were able to train and validate the model 461 times, helping us comprehensively evaluate the model's performance on individual gene in a disease-specific manner. In the independent test, GeneralSet is performed as the training set and NDDSet2 is used as the test set. For the online database, the unlabeled mutations in NDDSet1 is used for blind prediction.

2.5 Evaluation Metrics

To comprehensively evaluate the performance of PmmNDD against peer methods, we considered a wide range of metrics, including the Area Under the Receiver Operating Characteristic Curve (AUC), Area Under the Precision-Recall Curve (AUPR), Matthews Correlation Coefficient (MCC), accuracy, F1-score, precision, recall and specificity. Different thresholds for a method will result in different values of MCC, accuracy, F1_score, recall and specificity. In this study, we use the Youden index on the AUC to determine the optimal threshold and calculate the corresponding MCC, accuracy, F1_score, recall and specificity.

3 Results and Discussion

3.1 Performance of Different Ensemble Models of PmmNDD

In this section, we extensively explored the performance of different PmmNDD ensemble models for mutation pathogenicity prediction in NDDs. We evaluated seven different machine learning algorithms, including CatBoost, XGBoost, LightGBM, RF, SVM, LR and AdaBoost, through two sets of experiments.

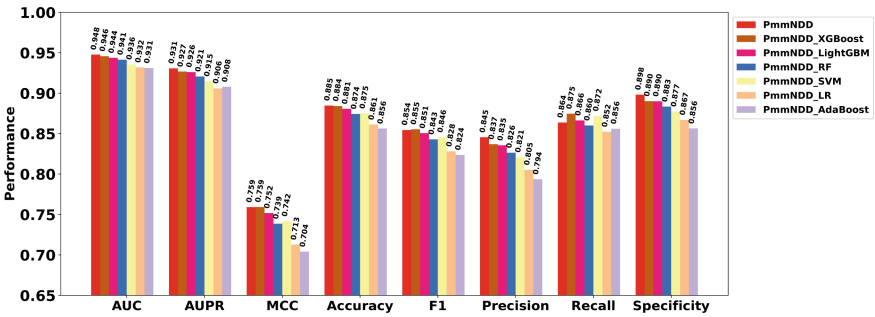

Fig. 2. Performance comparison of seven ensemble learning models based on leave-one-gene-out cross validation

In the first set of experiments, we carried out the leave-one-gene-out cross validation to assess the models' performance. The experimental results, as shown in Fig. 2,

indicated that CatBoost with an AUC of 0.948, significantly outperformed the other algorithms. Additionally, CatBoost also demonstrated leading positions in other performance metrics, showcasing its potential in predicting the pathogenicity of missense mutations in NDDs.

To further assess the model's stability and generalization ability, we carried out the independent test. The detailed results are presented in Fig. 3, where the CatBoost algorithm once again demonstrated its outstanding performance, maintaining a stable AUC around 0.948. This result highlights the robustness of the CatBoost algorithm, showing minimal performance fluctuations even under rigorous independent test. Because of its demonstrated strong potential and consistent high performance, we have chosen the CatBoost algorithm as the method for our predictive model. This decision is based not only on its exceptional performance in statistical metrics but also considering several unique advantages of CatBoost: including, but not limited to, its inherent ability to handle categorical features, minimal parameter tuning requirements, and fast training speed. These features make CatBoost an ideal choice for handling complex biological data, particularly in the field of neurodegenerative disease research where the interpretation of a large number of genetic variations impacting disease risk is required.

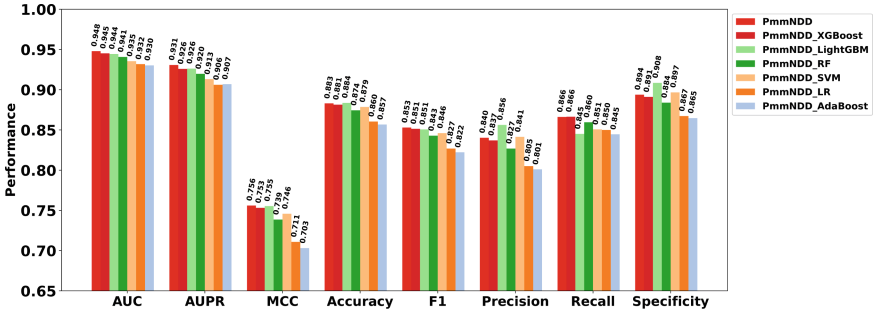

Fig. 3. Performance comparison of seven ensemble learning models based on independent test

By comparing the results from Fig. 2 and Fig. 3, we further observed that PmmNDD can achieve similar prediction performance when using leave-one-gene-out-cross-validation and independent test. The results demonstrated the mutation-function relationship learned by our ensemble model from non-NDD genes generalizes comparably results as models trained with NDD genes, highlighting the effectiveness of the curated datasets and PmmNDD's high model generalization capability.

3.2 Analysis of Feature Importance

The PmmNDD model is constructed based on 60 features, and to further understand how these features impact predictive performance, we have undertaken detailed analysis.

Firstly, we adopt an iterative feature exclusion method to train the model, observing a consistent decrease in AUC values as illustrated in Fig. 4(a). Particularly, the most significant decrease in prediction accuracy is observed when removing one-hot encoding,

gMVP_SCORE, and MetaRNN_SCORE, highlighting the importance of each feature in the model's performance. To compare the similarity of scores from different prediction tools, we conduct Jaccard coefficient evaluations accompanied by clustering analysis, as shown in Fig. 4(b). The results demonstrated that the similarity scores among most features fall within the range of 0.5 to 0.7, indicating a relative degree of independence among these features, contributing to the model's diversified prediction capabilities. In the clustering analysis, the scores of BayesDel_addAF, MetaRNN, and ClinPred are closely aligned with PmmNDD score, categorizing these methods into the same group and affirming their significance in the PmmNDD modeling process.

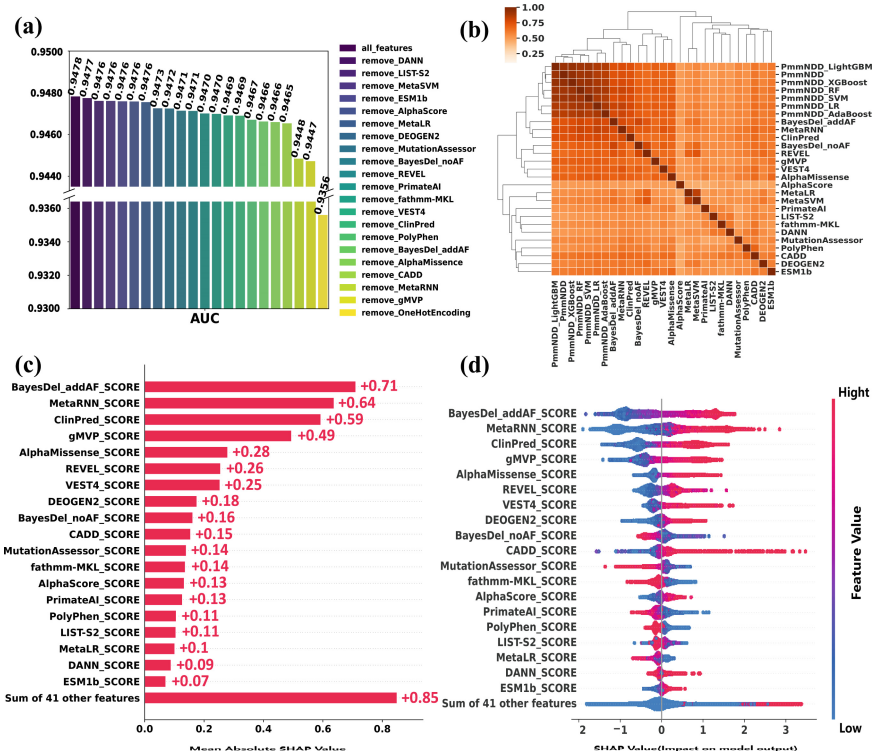

Fig. 4. Analysis of feature importance

Subsequently, we utilize Shapley Additive exPlanations (SHAP) analysis to provide interpretive metrics for model predictions by averaging the marginal contributions of various possible combinations of features. By utilizing a beeswarm feature density scatter plot as illustrated in Fig. 4(c), we can clearly observe the influence of each feature on the model's prediction performance. The plot uses color differentiation to represent the magnitude of feature values, with red indicating higher feature values and blue representing lower values. For example, the feature values of BayesDel_addAF_score exhibit a high concentration of red points in the positive SHAP value region, indicating a positive impact on the model's predictive outcomes; on the contrary, blue points are

mostly clustered in the negative SHAP value region, suggesting a potential negative effect. It is important to note that the positive or negative effect here does not directly represent the positive or negative effect on model accuracy, but instead signifies the enhancement or reduction in predictive outcomes due to that feature value. To gain a comprehensive understanding of the collective impact of each feature, we analyze how the sample points of different features spread along the central axis. It can be observed that compared to other individual features, the sample points of BayesDel_addAF_score have a broader coverage range, with dense points concentrated at both ends of the contour lines, implying a significant overall impact on the prediction performance. In contrast, Bayes_noAF_score exhibits a more centric distribution of sample points around the central line, indicating that most sample points have small SHAP values and, therefore, have a relatively minor overall impact on the prediction performance.

Finally, SHAP values provide a numerical and intuitive understanding of the importance of each feature, as shown in Fig. 4(d). Through analysis of these values, we can directly assess the relative importance of each feature in influencing the model's predictive outcomes.

3.3 Comparison with Existing Prediction Methods

Based on the independent test results, PmmNDD underwent a detailed performance comparison with 20 existing prediction tools. Figure 5 showed the overall performance comparison of PmmNDD with 20 other algorithms based on the whole NDDSet2. PmmNDD model exhibited significant advantages across multiple performance evaluation metrics, obtaining an AUC of 0.948 and an AUPR of 0.931. Additionally, it achieved the highest MCC, accuracy, F1, precision, recall and specificity with values of 0.756, 0.883, 0.853, 0.840, 0.866 and 0.894, surpassing the peer prediction methods.

To conduct a comprehensive evaluation of PmmNDD, we further analyzed the distribution of performance metrics for the 461 genes in the independent test set. Considering some genes contain only pathogenic or only benign mutations and extreme sparse labels in individual genes may cause bias to the final results, genes with fewer than 10 or 20 labels were excluded before re-evaluation in Fig. 6 and Fig. 7. Figure 6 showed the distribution of different metrics on individual NDD genes with more than 10 labels. PmmNDD outperformed the other 20 methods across six major metrics. The predictive results for AUC, AUPR, accuracy, F1, and precision predominantly ranged from 0.8 to 1, exhibiting a noticeable funnel-shaped distribution; while most MCC scores were above 0.6. The PmmNDD model also demonstrated a comparative advantage in terms of the average values (blue dot in the corresponding violin plot) of all evaluation metrics. Figure 7 showed the distribution of different metrics on individual NDD genes with more than 20 labels. PmmNDD model exhibits a more concentrated distribution of metrics and generally outperforms the peer methods, with average AUC, AUPR, accuracy, MCC, F1 and precision of 0.913, 0.847, 0.890, 0.670, 0.768 and 0.785, respectively.

These consistent and comprehensive leading performances not only highlight the excellent prediction performance of the PmmNDD model in predicting the pathogenicity of missense mutations related to NDDs but also reveal the model's unique strength in understanding and predicting the relevance of genetic variation to diseases.

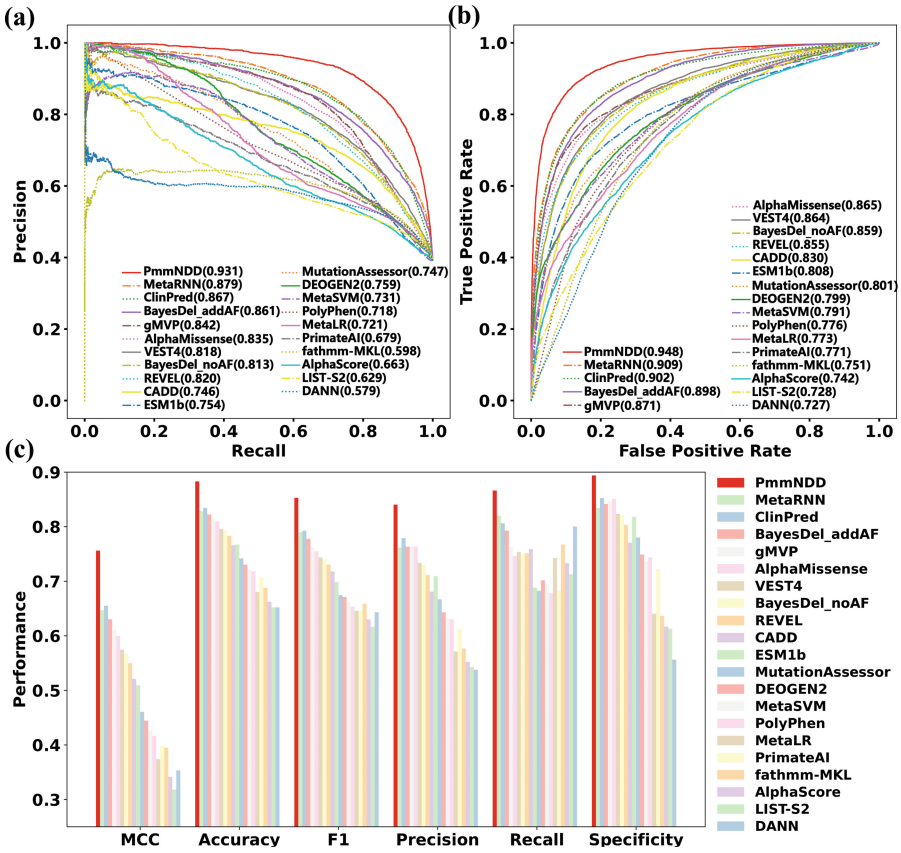

Fig. 5. Overall performance comparison of PmmNDD with 20 other algorithms

3.4 Predictions for 3 Million Missense Mutations in NDDs

We provide both continuous PmmNDD scores and class assignments for the 3285131 single amino acid mutations across 595 NDD-associated genes. Of these mutations, approximately 399032 have been observed in at least one human, but only around 19% (most are uncertain significance) have any clinical interpretation in ClinVar, 3.7% are interpreted as pathogenic or benign in ClinVar. The PmmNDD class assignments, after dropping the uncertain variants to keep accuracy at approximately 90%, provide an interpretation for about 2.4 million variants in total and over 250000 (approximately 60%) of the variants seen to date in humans. Pre-calculated PmmNDD score and class assignments for all missense mutations in 595 NDDs genes can be accessed through online database (http://www.csbio.top/pmmndd) or downloaded from the URL (https://github.com/QNGFM/PmmNDD).

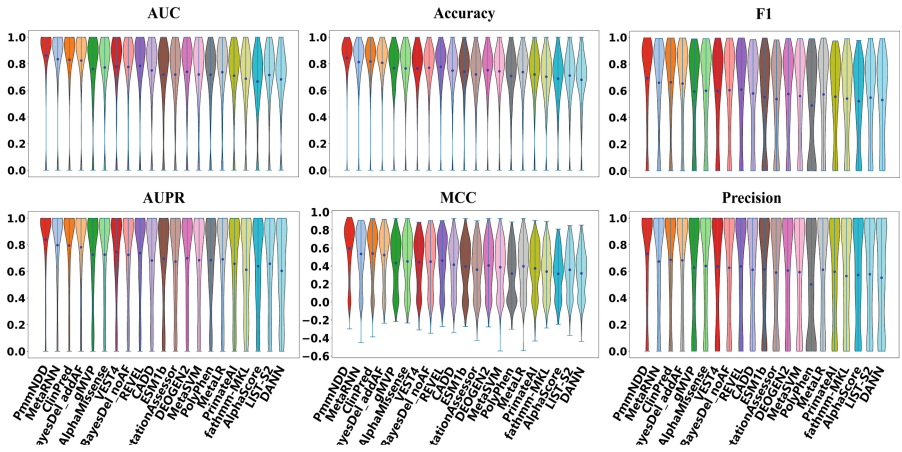

Fig. 6. Performance distribution of NDD genes with more than 10 labels

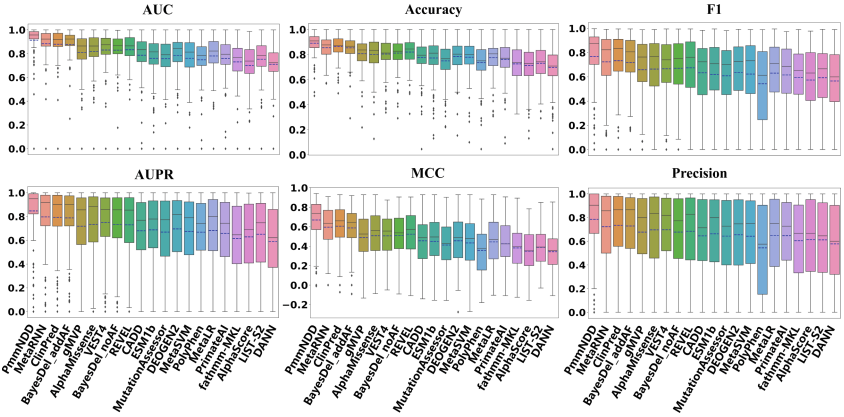

Fig. 7. Performance distribution of NDD genes with more than 20 labels

4 Conclusions

Neurodegenerative diseases are a group of neurological disorders characterized by the progressive loss of neurons in the central nervous system or peripheral nervous system. Common neurodegenerative diseases such as Alzheimer's disease and Parkinson's disease affect over 50 million people worldwide, with the numbers expected to steadily increase in the next decade, posing a significant burden on global public health. Missense mutations in neurodegenerative diseases can lead to abnormal protein folding, metabolic disruptions, or altered signal transduction, resulting in damage or even death of nerve cells. Accurately predicting the harmful effects of missense mutations plays a crucial role in clinical diagnosis and treatment, as well as in understanding the pathogenic mechanisms of neurodegenerative diseases.

In this study, we developed PmmNDD, an ensemble method for predicting pathogenicity of missense variants in NDDs. PmmNDD integrated the prediction scores from 20 different methods along with 40 dimensions of amino acid one-hot encoding as features, and was constructed with the categorical boosting model. The evaluation of PmmNDD demonstrated its superior performance over individual methods. We also demonstrated the mutation-function relationship learned by our ensemble model from non-NDD genes generalizes comparably results as models trained with NDD genes. Although PmmNDD has demonstrated superior performance compared to individual prediction methods, it is important to acknowledge its limitations and potential impacts in clinical settings. PmmNDD's reliance on pre-existing prediction tools introduces a dependency on the accuracy and biases inherent in these underlying methods. Future endeavors will involve improving the generalizability and robustness of the model by developing sub-models based on other types of features to reduce reliance on existing methods.

In summary, our study demonstrates the potential power gain of integrating prediction scores of various methods for pathogenicity prediction of missense mutations. PmmNDD contributes as a valuable tool for NDD mutation interpretation that will be widely useful in research and clinical scenarios.

Acknowledgements. This work was partly supported by Natural Science Foundation of Guangdong Province (General Program to Huiling Zhang), the National Science Foundation of China (Grant No. 62272449), the Shenzhen Basic Research Fund (Grant No. RCYX20200714114734194, KQTD20200820113106007), and the National Key R&D Program of China (Grant No. 2021YFD1300100).

References

1. Acosta-Uribe, J., et al.: A neurodegenerative disease landscape of rare mutations in Colombia due to founder effects. Genome Med. **14**(1), 27 (2022)
2. Dilliott, A.A., et al.: Contribution of rare variant associations to neurodegenerative disease presentation. NPJ Genom. Med. **6**(1), 80 (2021)
3. Strianese, O., et al.: Precision and personalized medicine: how genomic approach improves the management of cardiovascular and neurodegenerative disease. Genes **11**(7), 747 (2020)
4. Baldacci, F., et al.: The path to biomarker-based diagnostic criteria for the spectrum of neurodegenerative diseases. Expert Rev. Mol. Diagn. **20**(4), 421–441 (2020)
5. Findlay, G.M., et al.: Accurate classification of BRCA1 variants with saturation genome editing. Nature **562**(7726), 217–222 (2018)
6. Reva, B., Antipin, Y., Sander, C.: Predicting the functional impact of protein mutations: application to cancer genomics. Nucleic Acids Res. **39**(17), e118–e118 (2011)
7. Malhis, N., Jacobson, M., Jones, S.J., Gsponer, J.: LIST-S2: taxonomy based sorting of deleterious missense mutations across species. Nucleic Acids Res. **48**(W1), W154–W161 (2020)
8. Adzhubei, I.A., et al.: A method and server for predicting damaging missense mutations. Nat. Methods **7**(4), 248–249 (2010)
9. Carter, H., Douville, C., Stenson, P.D., Cooper, D.N., Karchin, R.: Identifying Mendelian disease genes with the variant effect scoring tool. BMC Genomics **14**, 1–16 (2013)

10. Dong, C., et al.: Comparison and integration of deleteriousness prediction methods for non-synonymous SNVs in whole exome sequencing studies. Hum. Mol. Genet. **24**(8), 2125–2137 (2015)
11. Feng, B.J.: PERCH: a unified framework for disease gene prioritization. Hum. Mutat. **38**(3), 243–251 (2017)
12. Alirezaie, N., Kernohan, K.D., Hartley, T., Majewski, J., Hocking, T.D.: ClinPred: prediction tool to identify disease-relevant nonsynonymous single-nucleotide variants. Am. J. Hum. Genet. **103**(4), 474–483 (2018)
13. Shihab, H.A., et al.: An integrative approach to predicting the functional effects of non-coding and coding sequence variation. Bioinformatics **31**(10), 1536–1543 (2015)
14. Rentzsch, P., Witten, D., Cooper, G.M., Shendure, J., Kircher, M.: CADD: predicting the deleteriousness of variants throughout the human genome. Nucleic Acids Res. **47**(D1), D886–D894 (2019)
15. Ioannidis, N.M., et al.: REVEL: an ensemble method for predicting the pathogenicity of rare missense variants. The Am. J. Hum. Genet. **99**(4), 877–885 (2016)
16. Raimondi, D., et al.: DEOGEN2: prediction and interactive visualization of single amino acid variant deleteriousness in human proteins. Nucleic Acids Res. **45**(W1), W201–W206 (2017)
17. Quang, D., Chen, Y., Xie, X.: DANN: a deep learning approach for annotating the pathogenicity of genetic variants. Bioinformatics **31**(5), 761–763 (2014)
18. Sundaram, L., et al.: Predicting the clinical impact of human mutation with deep neural networks. Nat. Genet. **50**(8), 1161–1170 (2018)
19. Zhang, H., Xu, M.S., Fan, X., Chung, W.K., Shen, Y.: Predicting functional effect of missense variants using graph attention neural networks. Nat. Mach. Intell. **4**(11), 1017–1028 (2022)
20. Cheng, J., et al.: Accurate proteome-wide missense variant effect prediction with AlphaMissense. Science **381**(6664), eadg7492 (2023)
21. Brandes, N., Goldman, G., Wang, C.H., Ye, C.J., Ntranos, V.: Genome-wide prediction of disease variant effects with a deep protein language model. Nat. Genet. **55**(9), 1512–1522 (2023)
22. Jumper, J., et al.: Highly accurate protein structure prediction with AlphaFold. Nature **596**(7873), 583–589 (2021)
23. Varadi, M., et al.: AlphaFold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. Nucleic Acids Res. **50**(D1), D439–D444 (2022)
24. Schmidt, A., Röner, S., Mai, K., Klinkhammer, H., Kircher, M., Ludwig, K.U.: Predicting the pathogenicity of missense variants using features derived from AlphaFold2. Bioinformatics **39**(5), btad280 (2023)
25. Vogan, K.: Improved pathogenicity prediction using primate genomics. Nat. Genet. **55**(7), 1082 (2023)
26. UniProt: the universal protein knowledgebase in 2023. Nucleic Acids Res. **51**(D1), D523–D531 (2023)
27. Landrum, M.J., et al.: ClinVar: improvements to accessing data. Nucleic Acids Res. **48**(D1), D835–D844 (2020)
28. Liu, X., Li, C., Mou, C., Dong, Y., Tu, Y.: DbNSFP v4: a comprehensive database of transcript-specific functional predictions and annotations for human nonsynonymous and splice-site SNVs. Genome medicine **12**, 1–8 (2020)

Check for updates

Improved Inapproximability Gap and Approximation Algorithm for Scaffold Filling to Maximize Increased Duo-Preservations

Jinting Wu and Haitao Jiang(✉)

School of Computer Science and Technology, Shandong University, Qingdao, Shandong, China
wujinting@mail.sdu.edu.cn, htjiang@sdu.edu.cn

Abstract. Scaffold filling is a critical step in DNA assembly. In the scaffold filling problem, we are given a reference (complete) genome, and a scaffold composed of contigs which are matched to fix positions on the reference genome, as well as some unmatched fragments, the purpose is to insert the unmatched fragments between the contigs in the scaffold, such that the resulting genome is similar to the reference genome. Let M be a one-to-one matching between common letters of the scaffold and the reference genome. A duo-preservation is an ordered pair of consecutive letters in the scaffold, which are matched to two consecutive letters in the reference genome based on M. The problem of scaffold filling to maximize increased duo-preservations is described as: given an incomplete scaffold with some fragments missing and a reference genome, inserting the missing fragments back into the incomplete scaffold to maximize the number of increased duo-preservations between the filled scaffold and the reference genome. In [19], this problem was shown to be MAX-SNP-complete and can not be approximated within $\frac{16263}{16262}$. In this paper, we firstly improve the inapproximability gap to $\frac{2363}{2362}$, then we devise a new approximation algorithm with an approximation factor of $\frac{3}{2}+\epsilon$ by a local search method. The running time of the approximation algorithm is $O(n^{O(\frac{1}{\epsilon})})$, where ϵ is an arbitrary small constant. Finally, we apply our algorithm to simulated genomic data, yielding the average approximation factors of 1.034 for $s=1$ and 1.030 for $s=2$, where s denotes the range of local search, indicating the maximum number of elements substituted in the current solution per iteration. The experimental findings show that our algorithm's approximation factor primarily reflects a theoretical worst-case. However, practical datasets rarely encounter such extremes cases. Therefore, our approximation algorithm shows exceptional performance in real-world datasets compared with the theoretical bound.

Keywords: Scaffold Filling · Inapproximability Gap · Approximation Algorithm

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2024
W. Peng et al. (Eds.): ISBRA 2024, LNBI 14956, pp. 76–88, 2024.
https://doi.org/10.1007/978-981-97-5087-0_7

1 Introduction

Scaffold filling is to fill the missing letters or fragments into the incomplete genome scaffold through combinatorial optimization techniques and methods to make the genome sequence be complete and similar to the reference genome. Currently, the dramatic decrease in the cost of genome sequencing has increased the range of organisms available for genome analysis. However, the cost of finishing has not dropped at the same rate as the expense of random sequencing. The final releases of the genomes are usually in the form of scaffolds or contigs. The use of draft genomes will have impacts on the study of the genomes. On the one hand, it makes many analyses and interpretations tentative and prone to error. On the other hand, it leads to particular problems in the comparative study of gene order. Therefore, scaffold filling becomes an important task before conducting the whole genome analysis [21].

Naturally, the purpose of scaffold filling is to obtain a comparatively complete genome which is much more similar to the reference genome. There are many commonly used methods to measure the similarity between two genomes viewed as strings. They include rearrangement distance [12], exemplar breakpoint distance [22], breakpoint distance [15], minimum common string partition distance [10], maximum common adjacencies [17] and so on. For genomes without repeated genes (generally referred to as a permutation), it was shown that it is polynomial-time solvable by any measure even in the two-sided case where both the input scaffolds, being a reference to each other, are incomplete permutations [13].

When the reference genomes and scaffolds contain gene repetitions, the problem becomes much harder. Unfortunately, computing some distances between two genomes are inherently difficult. Bryant showed that the calculation of the exemplar breakpoint distance between two genomes is NP-hard [5]. Later, Blin *et al.* proved that the exemplar breakpoint distance problem cannot be approximated at all as soon as both genomes contain duplicates [3]. The minimum common string partition problem is also NP-hard even if each letter appears at most twice in either string [10]. Therefore, exemplar breakpoint distance and minimum common string partition are not usually used directly for scaffold filling problems. The number of breakpoints and common adjacencies are polynomial-time computable, they can be used as the standard measurement for the sequence similarity comparison after the scaffold filling. The sum of the number of common adjacencies and breakpoints is one less than the length of the genome. Therefore, the minimum breakpoint distance problem and the maximum common adjacency problem are complementary. The number of common adjacencies was first used by Angibaud *et al.* to compare the structure of genome [1], and has been applied to scaffold filling since 2010.

Given two genomes A and B, we say that uv is a common adjacency when uv is a substring of length two in A and uv or vu also appears in B. So, the common adjacencies of A and B are defined as maximum multi-set of common adjacencies, provided that each substring of length two in A and B appears in at most one common adjacency. Jiang *et al.* extended the common adjacency from permutations to genomes with replicated genes, and study the problem

of scaffold filling to maximize the number of common adjacencies. Though the problem is still NP-hard, it shows good computational properties in both aspects of approximation and parameterized tractability [6,14,16]. In order to obtain better approximations, Liu *et al.* studied the problem systematically [17,18]. The number of common adjacencies shows a good similarity between two genomes, but it ignores the fact that even if all the substrings of length two in the two genomes are common adjacencies, the two genomes may also be different. For example, we have two genomes "$A = axbcxd$" and "$B = axcbxd$". Though all the substrings of length two in A and B are common adjacencies, it is obvious that A is not identical to B. Thus, in [19], *duo-preservation* is proposed. The *duo-preservation* is the complementary measurement of the well-known common substring partition distance of two sequences, which is well-studied in [4,7,9,11].

An ordered pair of consecutive letters in a string is called a duo. A duo is preserved by a common partition, if the pair resides inside a common substring of the partition. The preserved duos are also called duo-preservations in a common partition. The sum of the number of duo-preservations and common partition is exactly the length of the strings. Thus, the complement problem of MCSP, the maximum duo-preservations problem (*abbr.* MDPP) is proposed, which aims to find a common partition with the maximum number of duo-preservations. MDPP has attracted much interest in the recent years, since it admits better approximations compared with MCSP [8,23].

A duo-preservation is different from a common adjacency because it implies a one-to-one correspondence between the genes. For example, let $A = abcdb$, $B = abdbc$, then ab, bc and db are all common adjacencies. But under the common partition, $A = ab|c|db$ and $B = ab|db|c$, only ab and db are duo-preservations. We can observe that the letters $'b'$ of ab and bc are the same in A, but not in B, which is not allowed by the definition of duo-preservation.

While genome sequencing, prior to scaffold filling, the scaffold contains many large contigs, each of which has been matched to a fixed segment on the reference genome. The main task of scaffold filling is to fill the missing fragments to proper positions without changing the contig matching already existing. In [19], Ma *et al.* adopted duo-preservation as the measurement to evaluate a scaffold filling, and proposed the scaffold filling to maximize increased duo-preservations problem (*abbr.* SF-MIDP). In this paper, we improve the inapproximability gap and devise new approximation algorithm for this problem with better approximation factor. The main contributions of this paper are as follows. (1) We prove that SF-MIDP can not be approximated with in $\frac{2363}{2362}$. (2) We present an approximation algorithm of factor $\frac{3}{2} + \epsilon$ by a local search method, which runs in $O(n^{O(\frac{1}{\epsilon})})$ time, where ϵ is an arbitrary small constant.

The rest of the paper is organized as follows. In Sect. 2, we give the necessary definitions for the problems and the corresponding basic properties. In Sect. 3, we prove that the SF-MIDP is MAX SNP-complete, and give an inapproximability gap for it. In Sect. 4, we devise an approximation algorithm for SF-MIDP. Finally, we conclude the paper in Sect. 6.

2 Preliminaries

Firstly, we review some necessary definitions, which are also defined in [19]. Given the alphabet Σ, a string G is called a sequence if its elements form a multiset of Σ. In this paper, we focus on sequences with gene (letter) repetitions. A scaffold is an incomplete sequence with some missing letters. Let $S = s_1 s_2 \dots s_n$ be a scaffold or a genome. The multiset of letters in S is denoted by $c(S)$. For any symbol $x \in \Sigma$, let $g(c(S), x)$ be the number of occurrences of x in the multiset $c(S)$. Two consecutive letters s_i and s_{i+1} in S form an adjacency. The first letter of S is called the *head* of S, which is denoted by $h(S)$, and the last letter of S is called the *tail* of S, which is denoted by $t(S)$. An ordered pair of two adjacent letters in S is called a *duo*. A substring is a consecutive segment of a string. A *k-substring* is a substring with k letters. A *block* is a substring of length at least two. We use $S_1|S_2$ to denote the *concatenation* of the two substrings S_1 and S_2.

Given two genomes A and B on Σ, where $g(c(A), x) \geq g(c(B), x)$ for all $x \in \Sigma$, let $M = \{I_1, I_2, \dots, I_j\}$ be a multiset of j blocks, in which all blocks are pairwise disjoint in both A and B, then M is represented as a *block-matching* of A and B, and for $1 \leq i \leq j$, the block I_i in A and the block I_i in B are matched with each other. If a letter occurs in a block of a block-matching, we say it is *matched*, and otherwise, it is *unmatched*. For a matched letter x under some block-matching M, we use $M(A, x)$ and $M(B, x)$ to denote the specific matched couple in A and B respectively.

If a duo is a substring of some block under a block-matching, we say it is *preserved* by the block-matching, and is represented as a *duo-preservation*. Otherwise, it is called a *breakpoint* under this block-matching. For a block $I \in M$, the set of duo-preservations preserved by I is denoted by $dp(I)$. Let $dp(M)$ denote the set of duo-preservations under M. Then, $dp(M) = \cup_{I \in M} dp(I)$. The set of breakpoints in A (resp. B) under the block-matching M is denoted by $bp(M, A)$ (resp. $bp(M, B)$). From the above definitions, a duo in A or B would either be a duo-preservation or a breakpoint under some block-matching, so the sum of the number of duo-preservations and the number of breakpoints in A (resp. B) is $n - 1$ (resp. $m - 1$), where n(resp m) is the length of A (resp. B). Note that, under any block-matching, the set of duo-preservations in A is the same as that in B, and the set of matched letters in A is the same as that in B.

For two block-matchings M and M' both with respect to A and B, we say that M' *contains* M, if $M' \neq M$, and all the blocks of M are disjoint substrings of blocks of M'. We use $M' \supset M$ to denote that M' contains M, and $M' \supseteq M$ to denote that M' contains M or $M' = M$. A block-matching is *maximal* if no other block-matching contains it.

For two multisets $c(A)$ and $c(B)$, we use $X = c(A) - c(B)$ to denote the multiset of missing letters which satisfies that $x \in X$ if and only if $g(c(A), x) > g(c(B), x)$, and $g(X, x) = g(c(A), x) - g(c(B), x)$. Then, we present the problem investigated in this paper formally as follows.

Definition 1. The scaffold filling to maximize increased duo-preservations problem (abbreviated as *SF-MIDP*) [19].

Instance: *A reference genome* $A = \#\#a_1 \dots a_n\$\$$ *and an incomplete scaffold* $B = \#\#b_1 \cdots b_m\$\$$*, the corresponding multi-set of missing letters* $X = c(A) - c(B)$*, a maximal block-matching* M *with respect to* A *and* B*.*

Solution: *A complete genome* B' *by inserting all the letters of* X *into the breakpoints of* B *under* M*, and a new block-matching* $M' \supseteq M$*.*

Measure: *Number of increased duo-preservations, that is number of duo-preservations based on* M' *minus number of duo-preservations based on* M*.*

Similar to [19], we also add ## and $$ to the two ends of the genomes respectively, so that each letter can form two duos together with the letters appearing on its right side and left side.

3 An Improved Inapproximability Gap for SF-MIDP

In this section, we show that the SF-MIDP problem can not be approximated with in $\frac{2363}{2362}$ by a L-reduction from the independent set problem on 3-bounded graphs (where the degree of each vertex is bounded by three).

We show that the SF-MIDP is MAX SNP-complete, which makes a polynomial time approximation scheme nearly impossible. To complete this proof, we construct an L-reduction from a variation of a known MAX SNP-complete problem (the bounded degree Independent Set problem), to the SF-MIDP problem. Also, we give an inapproximability gap for SF-MIDP.

A graph is called k-bounded graph if the degree of each vertex is at most k. The maximum independent set problem on 3-bounded graph (abbreviate as MIS-3) is shown to be MAX SNP-complete in [2]. M.Chlebík and J.Chlebíková obtained a stronger hardness result.

Lemma 1. *MIS-3 can not be approximated within* $\frac{139}{138}$ [20].

Theorem 1. *It is NP-hard to approximate SF-MIDP within a factor of* $\frac{2363}{2362}$.

4 Approximation Algorithms for SF-MIDP

In this section, we present a $\frac{3}{2} + \epsilon$-approximation algorithm for the SF-MIDP problem. Recall that the basic approximation algorithm BasicInsert(A,B,X,M) in [19], guarantees that the number of new duo-preservations is exactly equal to the number of letters inserted.

Lemma 2. *Algorithm BasicInsert(*A,B,X,M*) increases* $|X|$ *duo-preservations by inserting* $|X|$ *letters into* B *and runs in* $O(n^2)$ *time, where* X *is the multi-set of the missing letters* [19].

Next, we introduce the good k-substring, which is also described in [19]. A substring $S = x_1 \cdots x_k$, $(x_i \in X,\ 1 \leq i \leq k)$, is called a *good* k-substring of a block-matching $M' \supset M$, if $b_i b_{i+1}$ is a breakpoint of B under M and $b_i | x_1 \cdots x_k | b_{i+1}$ turns into a block or a substring of some block in M' after

inserting S in between b_i and b_{i+1}. we can see that it will bring $k+1$ new duo-preservations by inserting a *good* k-substring into a breakpoint of B under some block-matching $M' \supseteq M$.

Let B^* be the optimal resulting scaffold and M^* be the corresponding block-matching with respect to A and B^*. Let μ_k^* and ν_k^* denote the number of good and not good k-substrings in the optimal scaffold filling, respectively, where $k = 1, 2, \ldots$, then Ma *et al.* showed the following upper bound of the optimal solution.

Lemma 3. $|dp(M^*)| - |dp(M)| \leq \frac{3}{2}|X| + \frac{1}{2}\mu_1^*$ [19].

From Lemma 3, the algorithm will reach a better approximation factor, if it can obtain more good 1-substrings. But some of these good k-substrings cannot co-exists based on a block matching M with each other. Given two scaffolds A and B, let B_1, B_2 be different resulting scaffolds after inserting some letters of X into B, and M_1 (resp. M_2) be the block-matching with respect to A and B_1 (resp. B_2). Then a letter x in A or B is *mismatched* under M_1 and M_2, if $M_1(A,x) = M_2(A,x)$ & $M_1(B_1,x) \neq M_2(B_2,x)$ or $M_1(B_1,x) = M_2(B_2,x)$ & $M_1(A,x) \neq M_2(A,x)$. Suppose that $S_1 = x_1 \cdots x_k$ is a good k-substring of M_1 and $S_2 = y_1 \cdots y_{k'}$ is a good k'-substring of M_2, which are filled into $b_j b_{j+1}$ and $b_{j'} b_{j'+1}$ in B respectively, the corresponding matched substrings in A are $a_i a_{i+1} \cdots a_{i+k+1}$ and $a_{i'} a_{i'+1} \cdots a_{i'+k'+1}$. Then, S_1 and S_2 *conflict* with each other, if (1) $c(S_1) \cap c(S_2) \neq \emptyset$, or (2) at least one of $a_i, b_j, b_{j+1}, a_{i+k+1}$ is *mismatched* under M_1 and M_2. In [19], the following Lemma show the upper bound of the number of good 1-substrings that conflict with a good 1-substring.

Lemma 4. *Any good k-substring conflicts with at most $k+5$ good substrings of M^** [19].

[H] **Algorithm 1.** Local Search Scaffold Filling(A,B,X,M,s)

Input: reference scaffold: A; incomplete scaffold: B; X: $c(A) - c(B)$; M: a maximal block-matching with respect to A and B; a positive integer s.
Output: B', M'.

1: Identify the matched and unmatched letters of A and B under M respectively.
2: Describe A as $A = a_1 a_2 \cdots a_n$.
3: $P \leftarrow \emptyset$, $H \leftarrow \emptyset$, $B'' \leftarrow B$, $M'' \leftarrow M$, $X'' \leftarrow X$.
4: **for** (each $x \in X''$) **do**
5: **if** ($x = a_i$ is a good 1-substring) **then**
6: $H \leftarrow H + \{x\}$.
7: **end if**
8: **end for**
9: Find an initial set P from H in which pairings do not conflict.
10: **for** (k from 1 to s) **do**
11: **while** (There are a set Z composed of $k+1$ new good 1-substrings in H after deleting a subset Y from P, where $|Y| = k$) **do**

12: $P \leftarrow P - Y + Z$.
13: **end while**
14: **end for**
15: Update B'', X'' and M'' by inserting the items of P to B''.
16: B', $M' \leftarrow$ BasicInsert(A,B'',X'',M'').
17: **return** B',M'.

4.1 An Approximation Algorithm for the SF-MIDP

In this subsection, we propose a new approximation algorithm for SF-MIDP. The main idea of our algorithm is searching for more good 1-substrings by a local search method. There are three stages in our algorithm: the first stages use a greedy method, which provides an initial good 1-substrings set for the local search stage; then, the algorithm improves the good 1-substrings set by a local search method, during which, the algorithm obtains a larger good 1-substrings set by replacing at most s good 1-substrings iteratively, then updates the solution by inserting these good 1-substrings; finally, the algorithm calls Algorithm 1 to insert the remaining letters of X. The pseudo-code of the algorithm is shown in Algorithm 1.

We still use B' and M' to denote the scaffold and the block-matching returned by Algorithm 1, respectively. Also, we use $M^{''}$ to denote the block-matching obtained after running Step 14 in Algorithm 1. Let μ_1' denote the number of good 1-substrings obtained by the local search method in Algorithm 1. Then, we have,

Lemma 5. $|dp(M')| - |dp(M)| = |X| + \mu_1'$.

Proof. Since inserting a good 1-substring increases two duo-preservations, then $|dp(M'')| - |dp(M)| = 2\mu_1'$. Moreover, it follows from Lemma 2 that $|dp(M')| - |dp(M'')| = |X| - \mu_1'$. Thus, $|dp(M')| - |dp(M)| = |X| + \mu_1'$.

4.2 Proof of the Approximation Ratio

In this subsection, we prove that our algorithm achieves an approximation ratio of $\frac{3}{2} + \epsilon$.

Let P be the set of all good 1-substrings obtained by local search method in Algorithm 1, and P^* be the set of good 1-substrings in the optimal solution. To analyze the numerical relationship between μ_1^* and μ_1', consider the auxiliary bipartite graph $G_1 = (V_1, E_1)$ and $V_1 = V' + V^*$. Every vertex in V' denotes a good 1-substring of P. Every vertex in V^* denotes a good 1-substring of P^*. If a good 1-substring is in both P and P^*, then it corresponds to two vertices (one is in V' and the other is in V^*), which are connected by an edge in E_1. If the good 1-substring u ($u \in P$) conflicts with the good 1-substring v ($v \in P^*$), then there is an edge $(u, v) \in E_1$. Thus, we have $\mu_1' = |V'|$ and $\mu_1^* = |V^*|$.

According to Lemma 4, every vertex in V_1 is of degree at most six. Let μ_{1j}^* denote the number of vertices of degree j in V^*, where $j = 1, 2, 3, 4, 5, 6$.

Lemma 6. $\frac{\mu_1'}{\mu_1^*} \geq \frac{s+1}{3s+6}$.

Proof. Every vertex in V^* has degree at least one, since all the good 1-substrings of the optimal solution would either be in P or conflict with some good 1-substrings in P. Thus,

$$\mu_1^* = \mu_{11}^* + \mu_{12}^* + \mu_{13}^* + \mu_{14}^* + \mu_{15}^* + \mu_{16}^*. \tag{1}$$

In order to facilitate the analysis, we preprocess the graph G_1. For each vertex of degree d in V', we add $6-d$ vertices of degree one as its neighbors. We use V_0^* to denote the set of all the added vertices, and let $\mu_{10}^* = |V_0^*|$. Now, each vertex in V' is of degree six. Then, we partition the six neighbors of each vertex in V' into three disjoint pairs, remove all the vertices of V' and edges, and connect the two vertices of each pair via a new edge. We denote the new graph by $G_2 = (V_2, E_2)$, where $V_2 = V_0^* \cup V^*$. It is easy to see that every vertex in V_0^* has degree one, and every vertex in V^* has the same degree as G_1. Since all the degrees of the vertices in V^* and V_0^* are contributed by the vertices in V', and each vertex in V' has degree six, we can obtain the following equation:

$$6\mu_1' = \mu_{10}^* + \mu_{11}^* + 2\mu_{12}^* + 3\mu_{13}^* + 4\mu_{14}^* + 5\mu_{15}^* + 6\mu_{16}^*. \tag{2}$$

From Eq. (1) and Eq. (2), we have,

$$\frac{6\mu_1'}{\mu_1^*} = \frac{\mu_{10}^* + \mu_{11}^* + 2\mu_{12}^* + 3\mu_{13}^* + 4\mu_{14}^* + 5\mu_{15}^* + 6\mu_{16}^*}{\mu_{11}^* + \mu_{12}^* + \mu_{13}^* + \mu_{14}^* + \mu_{15}^* + \mu_{16}^*}. \tag{3}$$

Suppose there are k connected components in graph G_2. For each connected component $G_i^c = (V_i^c, E_i^c)$ $(1 \leq i \leq k)$, we use t_{i0} to represent the number of added vertices in G_i^c and t_{ij} to represent the number of vertices of degree j in $V_i^c \cap V^*$, $1 \leq j \leq 6$. Then,

$$\mu_{1r}^* = \sum_{i=1}^{k} t_{ir}, for \quad r = 0, 1, 2, 3, 4, 5, 6, \tag{4}$$

$$|V_i^c| = t_{i0} + t_{i1} + t_{i2} + t_{i3} + t_{i4} + t_{i5} + t_{i6}, \tag{5}$$

$$|E_i^c| = (t_{i0} + t_{i1} + 2t_{i2} + 3t_{i3} + 4t_{i4} + 5t_{i5} + 6t_{i6})/2. \tag{6}$$

Because G_i^c is a connected component of G_2, then $|E_i^c| \geq |V_i^c| - 1$, from Eq. (5) and Eq. (6), we have,

$$t_{i0} + t_{i1} \leq 2 + t_{i3} + 2t_{i4} + 3t_{i5} + 4t_{i6}. \tag{7}$$

Moreover, if $t_{i0} + t_{i1} = 1 + t_{i3} + 2t_{i4} + 3t_{i5} + 4t_{i6}$, we will have, $2|E_i^c| = 1 + 2t_{i2} + 4t_{i3} + 6t_{i4} + 8t_{i5} + 10t_{i6}$, which is impossible since the left side is even and

the right side is odd. Thus, we only need to consider the following two cases: (1) $t_{i0}+t_{i1} \le t_{i3}+2t_{i4}+3t_{i5}+4t_{i6}$; and (2) $t_{i0}+t_{i1} = 2+t_{i3}+2t_{i4}+3t_{i5}+4t_{i6}$.

Case (1): $t_{i0}+t_{i1} \le t_{i3}+2t_{i4}+3t_{i5}+4t_{i6}$. Since $t_{i0} \ge 0$, we have,

$$\frac{t_{i0}+t_{i1}+2t_{i2}+3t_{i3}+4t_{i4}+5t_{i5}+6t_{i6}}{t_{i1}+t_{i2}+t_{i3}+t_{i4}+t_{i5}+t_{i6}} \ge \frac{t_{i0}+t_{i1}+2t_{i2}+2t_{i3}+2t_{i4}+2t_{i5}+2t_{i6}+t_{i0}+t_{i1}}{t_{i1}+t_{i2}+t_{i3}+t_{i4}+t_{i5}+t_{i6}} \ge 2 > \frac{2\,s+2}{s+2}. \quad (8)$$

Case (2): $t_{i0}+t_{i1} = 2+t_{i3}+2t_{i4}+3t_{i5}+4t_{i6}$. We have two subcases.

Case (2.1): $t_{i0} \ge 1$, i.e., $t_{i1} \le 1+t_{i3}+2t_{i4}+3t_{i5}+4t_{i6}$. Then,

$$\frac{t_{i0}+t_{i1}+2t_{i2}+3t_{i3}+4t_{i4}+5t_{i5}+6t_{i6}}{t_{i1}+t_{i2}+t_{i3}+t_{i4}+t_{i5}+t_{i6}} \ge \frac{t_{i0}+t_{i1}+2t_{i2}+2t_{i3}+2t_{i4}+2t_{i5}+2t_{i6}+t_{i1}-1}{t_{i1}+t_{i2}+t_{i3}+t_{i4}+t_{i5}+t_{i6}} \ge 2 > \frac{2\,s+2}{s+2}. \quad (9)$$

Case (2.2): $t_{i0} = 0$, i.e., $t_{i1} = 2+t_{i3}+2t_{i4}+3t_{i5}+4t_{i6}$. According to Eq. (5), (6), we have $|V_i^c| = |E_i^c|+1$. Since $t_{i0} = 0$, $V_i^c \subseteq V^*$. Because each edge in G_i^c are constructed by a pair of neighbors of a vertex in V', but there are three edges in G_2 constructed according to each vertex in V', thus, all the edges in G_i^c are constructed by at most $|E_i^c|$ vertices in V'. Since Algorithm 1 terminates, it must fulfill that $|V_i^c| \ge s+2$, i.e., $t_{i1}+t_{i2}+t_{i3}+t_{i4}+t_{i5}+t_{i6} \ge s+2$. Thus,

$$\begin{aligned}&\frac{t_{i0}+t_{i1}+2t_{i2}+3t_{i3}+4t_{i4}+5t_{i5}+6t_{i6}}{t_{i1}+t_{i2}+t_{i3}+t_{i4}+t_{i5}+t_{i6}}\\ =\ &\frac{t_{i1}+2t_{i2}+2t_{i3}+2t_{i4}+2t_{i5}+2t_{i6}+t_{i1}-2}{t_{i1}+t_{i2}+t_{i3}+t_{i4}+t_{i5}+t_{i6}}\\ =\ &2-\frac{2}{t_{i1}+t_{i2}+t_{i3}+t_{i4}+t_{i5}+t_{i6}} \ge \frac{2\,s+2}{s+2}.\end{aligned} \quad (10)$$

From Eq. (3), (4), (8), (9) and Eq. (10), we have,

$$\frac{6\mu_1'}{\mu_1^*} = \frac{\sum_{i=1}^{k}(t_{i0}+t_{i1}+2t_{i2}+3t_{i3}+4t_{i4}+5t_{i5}+6t_{i6})}{\sum_{i=1}^{k}(t_{i1}+t_{i2}+t_{i3}+t_{i4}+t_{i5}+t_{i6})} \ge \frac{2\,s+2}{s+2}. \quad (11)$$

Equivalently, we have $\frac{\mu_1'}{\mu_1^*} \ge \frac{s+1}{3s+6}$.

Theorem 2. *Algorithm 1 guarantees an approximation ratio of* $\frac{3}{2}+\epsilon$ *and runs in* $O(n^{O(\frac{1}{\epsilon})})$ *time.*

Proof. As for the approximation factor, from Lemma 3, 5, 6, let $\epsilon = \frac{3}{8s+14}$, thus,

$$\begin{aligned}\frac{|dp(M^*)|-|dp(M)|}{|dp(M')|-|dp(M)|} &\le \frac{\frac{3}{2}|X|+\frac{1}{2}\mu_1^*}{|X|+\frac{s+1}{3\,s+6}\mu_1^*} = \frac{\frac{6\,s+12}{4\,s+7}\times[\frac{4\,s+7}{4\,s+8}(|X|-\mu_1^*)+\frac{4\,s+7}{3\,s+6}\mu_1^*]}{(|X|-\mu_1^*)+\frac{4\,s+7}{3\,s+6}\mu_1^*}\\ &\le \frac{6\,s+12}{4\,s+7} = \frac{3}{2}+\frac{3}{8\,s+14} = \frac{3}{2}+\epsilon.\end{aligned} \quad (12)$$

To identify the matched letters and unmatched letters in A and B, Algorithm 1 has to scan A and B, which needs $O(n)$ time, where n is the length of A. The first for-loop runs at most $|X| < n$ rounds. In each round, the algorithm checks whether a specific letter could become a good 1-substring, which takes $O(n^2)$ time, since it should find out every appearance of this specific letter in A, and then scan B to find a proper position to fill that letter. Thus, the first for-loop takes $O(n^3)$ time. Besides, we can know that $|H| < n^3$. Then, it takes $O(n^4)$ time to find an initial set P, because it needs to go through every element in H and determine if that element conflicts with an existing element in P, and $|P| < |X|$. The second for-loop runs at most s rounds. In the worst case, *i.e.*, $k = s$, the inner while-loop runs at most $|X|$ rounds because $|P|$ is going to increase by at least 1 for each replacement and $|P|$ cannot exceed $|X|$. In each replacement, there are $C^s_{|P|}$ combinations to delete s good 1-substrings from P and $C^{s+1}_{|H|}$ combinations to increase $s+1$ good 1-substrings from H. Besides, it will take no more than $O(n)$ time to judge the conflict relationship. Thus, the second for-loop takes $O(n^{4s+5})$ time. Later, it will take $O(n)$ time to update B and the block-matching. Finally, it will take $O(n^2)$ time to call Algorithm 1 to fill the the remaining missing letter. Consequently, the time complexity of Algorithm 1 is $O(n^{4s+5})$, *i.e.*, $O(n^{O(\frac{1}{\epsilon})})$.

5 Experimental Results

Algorithm 1 uses the local search method, so a large value of s will lead to huge time complexity. Thus, we implement Algorithm 1 when $s = 1$ and $s = 2$. In [1], Ma *et al.* proposed algorithm $BasicInsert(A, B, X, M)$ of factor 2 and algorithm $ScaffoldFilling(A, B, X, M)$ of factor $\frac{12}{7}$. We also implement these two algorithms. We applied our algorithm to the simulated genomic data. Compared with the optimal solution and Ma's algorthms. Algorithm 1 shows much better performance. The algorithm implemented above is freely available at https://github.com/WJTing/SF-MIDP.

On simulated genomic data, we set the length of reference genomes to be from 100 to 1000. For each setting, we acquire multiple independent random instances and calculate the average results. Because the process of obtaining the optimal solution of SF-MIDP problem is very complicated, we give the optimal solution of each sample through certain calculation in the process of generating the sample. In other words, during data generation, we possess prior knowledge regarding the form in which the optimal solution will manifest. As shown in Fig. 1 (a), we can see the superiority of Algorithm 1 clearly. Our algorithm approximates the optimal solution more closely than Ma's algorithm. For Algorithm 1, the performance at $s = 2$ is slightly better than the performance at $s = 1$. We conclude that Algorithm 1 performs much better than others, that's because more good 1-substrings are obtained. Besides, The increase of s expands the scope of local search, and on this basis more good 1-substrings can be obtained, then, the performance at $s = 2$ is slightly better than the performance at $s = 1$.

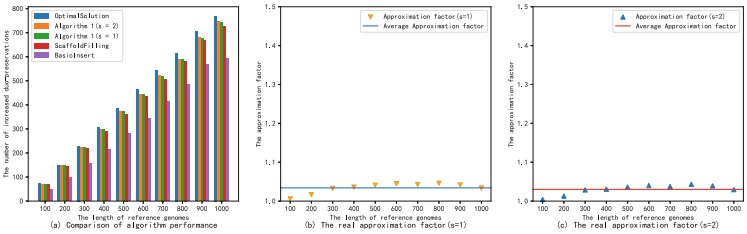

Fig. 1. Experimental results

In Fig. 1(b),(c), when $s = 1$, the real approximation factor of Algorithm 1 ranges from 1.004 to 1.050, and is 1.034 on average, when $s = 2$, the real approximation factor of Algorithm 1 ranges from 1.004 to 1.050, and is 1.030 on average. In conclusion, the real approximation factor is much better than the theoretical approximation factor, which indicates that the approximation factor of our algorithm represents merely a theoretical worst-case. In practical data, the worst-case nearly occurs. Consequently, our designed approximation algorithm demonstrates outstanding performance in real-world datasets.

6 Conclusion

In this paper, we investigate the scaffold filling to maximize increased duo-preservations problem(SF-MIDP). We improve the inapproximability gap from $\frac{16263}{16262}$ to $\frac{2363}{2362}$, and improve the approximation factor from $\frac{12}{7}$ to $\frac{3}{2} + \epsilon$ by a local search method. The algorithm shows much better performance on simulated genomic data. The main idea of our algorithm is still to find more good 1-substrings via local search method. We only consider good 1-substrings in our algorithm, because the approximation factor can not be better than $\frac{3}{2}$, if the number of good 1-substrings is less than a fraction of $\frac{1}{3}$ of the optimal solution. Thus, to improve the approximation factor, firstly, we need to find more good 1-substrings. Also, it is worth improving the inapproximability gap for SF-MIDP.

References

1. Angibaud, S., Fertin, G., Rusu, I., Thévenin, A., Vialette, S.: On the approximability of comparing genomes with duplicates. J. Graph Algor. Appl. **13**(1), 19–53 (2009)
2. Berman, P., Fujito, T.: On approximation properties of the independent set problem for low degree graphs. Theory Comput. Syst. **32**(2), 115–132 (1999)
3. Blin, G., Fertin, G., Sikora, F., Vialette, S.: The ExemplarBreakpointDistance for non-trivial genomes cannot be approximated. In: Das, S., Uehara, R. (eds.) WALCOM 2009. LNCS, vol. 5431, pp. 357–368. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00202-1_31

4. Blum, C., Lozano, J.A., Davidson, P.P.: Iterative probabilistic tree search for the minimum common string partition problem. In: Proceedings of International Workshop on Hybrid Metaheuristics (HM 2014), pp. 145–154 (2014)
5. Bryant, D.: The Complexity of Calculating Exemplar Distances, pp. 207–211. Springer, Dordrecht (2000). https://doi.org/10.1007/978-94-011-4309-7_19
6. Bulteau, L., Carrieri, A.P., Dondi, R.: Fixed-parameter algorithms for scaffold filling. Theor. Comput. Sci. **568**, 72–83 (2015)
7. Bulteau, L., Komusiewicz, C.: Minimum common string partition parameterized by partition size is fixed-parameter tractable. In: Proceedings of the 25th annual ACM-SIAM symposium on Discrete algorithms (SODA 2014), pp. 102–121 (2014)
8. Dudek, B., Gawrychowski, P., Ostropolski-Nalewaja, P.: A family of approximation algorithms for the maximum duo-preservation string mapping problem. In: Proceedings of 28th Annual Symposium on Combinatorial Pattern Matching (CPM 2017). LIPIcs, vol. 78, pp. 10:1–10:14 (2017)
9. Ferdous, S., Rahman, M.S.: Solving the minimum common string partition problem with the help of ants. Math. Comput. Sci. **11**(2), 233–249 (2017)
10. Goldstein, A., Kolman, P., Zheng, J.: Minimum common string partition problem: hardness and approximations. In: Fleischer, R., Trippen, G. (eds.) ISAAC 2004. LNCS, vol. 3341, pp. 484–495. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30551-4_43
11. Goldstein, I., Lewenstein, M.: Quick greedy computation for minimum common string partition. Theoret. Comput. Sci. **542**, 98–107 (2014)
12. Hannenhalli, S.: Polynomial-time algorithm for computing translocation distance between genomes. Disc. Appl. Math. **71**(1), 137–151 (1996)
13. Jiang, H., Fan, C., Yang, B., Zhong, F., Zhu, D., Zhu, B.: Genomic scaffold filling revisited. In: Proceedings of the 27th Annual Symposium on Combinatorial Pattern Matching (CPM 2016), vol. 15, pp. 1–13 (2016)
14. Jiang, H., Ma, J., Luan, J., Zhu, D.: Approximation and nonapproximability for the one-sided scaffold filling problem. In: Xu, D., Du, D., Du, D. (eds.) COCOON 2015. LNCS, vol. 9198, pp. 251–263. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21398-9_20
15. Jiang, H., Zheng, C., Sankoff, D., Zhu, B.: Scaffold filling under the breakpoint distance. In: Tannier, E. (ed.) RECOMB-CG 2010. LNCS, vol. 6398, pp. 83–92. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-16181-0_8
16. Jiang, H., Zhong, F., Zhu, B.: Filling scaffolds with gene repetitions: maximizing the number of adjacencies. In: Proceedings of the 22nd Annual Symposium on Combinatorial Pattern Matching (CPM 2011), pp. 55–64 (2011)
17. Liu, N., Jiang, H., Zhu, D., Zhu, B.: An improved approximation algorithm for scaffold filling to maximize the common adjacencies. IEEE-ACM Trans. Comput. Biol. Bioinform. **10**(4), 905–913 (2013)
18. Liu, N., Zhu, D., Jiang, H., Zhu, B.: A 1.5-approximation algorithm for two-sided scaffold filling. Algorithmica **74**(1), 91–116 (2016)
19. Ma, J., Jiang, H., Zhu, D., Yang, R.: Algorithms and hardness for scaffold filling to maximize increased duo-preservations. IEEE/ACM Trans. Comput. Biol. Bioinf. **19**(4), 2071–2079 (2022). https://doi.org/10.1109/TCBB.2021.3083896
20. Chlebík, M., Chlebíková, J.: Approximation hardness for small occurrence instances of NP-hard problems. In: Petreschi, R., Persiano, G., Silvestri, R. (eds.) CIAC 2003. LNCS, vol. 2653, pp. 152–164. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-44849-7_21

21. Muñoz, A., Zheng, C., Zhu, Q., Albert, V.A., Rounsley, S., Sankoff, D.: Scaffold filling, contig fusion and comparative gene order inference. BMC Bioinform. **11**, 304 (2010)
22. Nguyen, C.T., Tay, Y.C., Zhang, L.: Divide-and-conquer approach for the exemplar breakpoint distance. Bioinformatics **21**(10), 2171–2176 (2005)
23. Xu, Y., Chen, Y., Lin, G., Liu, T., Luo, T., Zhang, P.: A $(1.4 + \varepsilon)$-approximation algorithm for the 2-max-duo problem. In: Proceedings of 28th International Symposium on Algorithms and Computation (ISAAC 2017). LIPIcs, vol. 92, pp. 66:1–66:12 (2017)

Residual Spatio-Temporal Attention Based Prototypical Network for Rare Arrhythmia Classification

Zeyu Cao[1], Fengyi Guo[1(✉)], Ying An[2], and Jianxin Wang[1]

[1] Hunan Provincial Key Lab on Bioinformatics, School of Computer Science and Engineering, Central South University, Changsha 410083, China
234703015@csu.edu.cn

[2] Big Data Institute, Central South University, Changsha 410083, China

Abstract. Arrhythmia is a common cardiovascular disease that requires early detection and treatment to improve prognosis. Electrocardiogram (ECG) is an important tool for diagnosing and monitoring heart health. However, existing ECG diagnostic methods suffer from incomplete capture of spatio-temporal features and poor recognition ability for rare categories. In this paper, we introduce few-shot learning for ECG signals and propose a Residual Spatio-Temporal Attention based Prototypical Network (RSTA-ProtoNet) for rare arrhythmia classification. In the model, a spatio-temporal attention residual network is constructed as the backbone network. This network uses interleaved temporal and spatial attention encoders for extracting spatio-temporal features of ECG. Meanwhile, we build a few-shot learning framework based on prototypical networks to classify rare arrhythmia classes. This meta-training framework can learn useful features for classifying rare categories even with extremely limited samples of rare diseases. We evaluate our method on a large public ECG dataset, and the N-way K-shot experimental results demonstrate that RSTA-ProtoNet outperforms the state-of-the-art approaches in rare arrhythmia classification.

Keywords: Arrhythmia classification · Attention mechanism · Few Shot Learning · Meta-learning

1 Introduction

Arrhythmia is a common issue in cardiology, and many cardiac diseases are developed from benign arrhythmias [19]. Therefore, the timely detection of arrhythmias is crucial for the prevention of heart diseases. ECG is a medical examination method that records the electrical activity of the heart, widely used in clinical diagnosis. It provides physicians with a way to observe the electrophysiological characteristics of the heart. ECG can quickly assess the functional status of the heart and check for the presence of arrhythmias. Prolonged diagnosis may

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2024
W. Peng et al. (Eds.): ISBRA 2024, LNBI 14956, pp. 89–101, 2024.
https://doi.org/10.1007/978-981-97-5087-0_8

lead to observation fatigue in doctors, resulting in the omission of key information. Therefore, computer-aided diagnosis of arrhythmias based on ECG is a challenging and promising task.

Deep learning techniques have been widely applied to the detection and diagnosis of arrhythmias [1,6,15]. Some researchers have also utilized the popular attention mechanism in deep learning to extract features from ECG signals. This approach has gained widespread application due to its ability to enhance feature representation and extract relevant information from ECG signals. Che et al. [3] introduced a CNN-based framework with an embedded transformer network to enhance ECG signal classification by capturing temporal information. Jin et al. [10] employed a dual-attention convolutional long short-term memory network to detect intra-patient and inter-patient atrial fibrillation. Yao et al. [21] proposed an attention-based time-incremental convolutional neural network model for multi-class arrhythmia detection from 12-lead varied-length ECGs. However, these studies have focused solely on either the temporal or spatial information of the ECG, limiting the model's feature extraction capabilities. Therefore, we propose a spatio-temporal attention residual network to fully extract the spatio-temporal information of ECG.

Additionally, in reality, the scarcity of samples for some rare diseases leads to deep learning models being biased towards more common categories and underperforming on less frequent ones. Some studies use methods such as SMOTE [16], Focal Loss [11], and GAN [4] to address class imbalance problems. However they are not very effective in recognizing extremely rare categories. Few-shot learning, as an effective method, has been widely applied to the identification of scarce data. For example, Gupta et al. [7] proposes a few-shot learning algorithm based on similarity learning for time series classification using a Siamese Convolutional Neural Network. Liu et al. [14] proposed the Meta Siamese Network for heart pairs matching in few-shot learning, utilizing a residual network for feature extraction. However, the existing research on rare arrhythmias classification is still very limited. Therefore, we propose a meta-training framework based on prototypical networks for the identification of rare arrhythmia samples.

In order to fully extract effective features from both temporal and spatial dimensions of ECG and achieve efficient identification of rare arrhythmia classes, we propose a Residual spatio-temporal Attention based Prototypical Network (RSTA-ProtoNet). The main contributions of this paper are summarized as follows:

(1) We introduce a residual spatio-temporal attention feature extractor composed of spatial attention encoders and temporal attention encoders, capable of capturing the information across both the temporal and spatial dimensions of ECG.
(2) We propose a few-shot learning framework based on prototypical networks, which employs a residual spatio-temporal attention feature extractor as the backbone network for meta-learning. This approach effectively detects extremely rare arrhythmias in cases with limited training samples.

2 Methods

The overall framework of our model is illustrated in Fig. 1. Firstly, we construct a residual spatio-temporal attention feature extractor for ECG feature extraction and employ a fully connected layer for the classification of majority class diseases, simultaneously achieving pre-training of the backbone network model. Secondly, we construct few-shot tasks, specifically targeting rare class diseases to construct prototypical networks for meta-learning. Finally, we conduct meta-testing on rare class diseases.

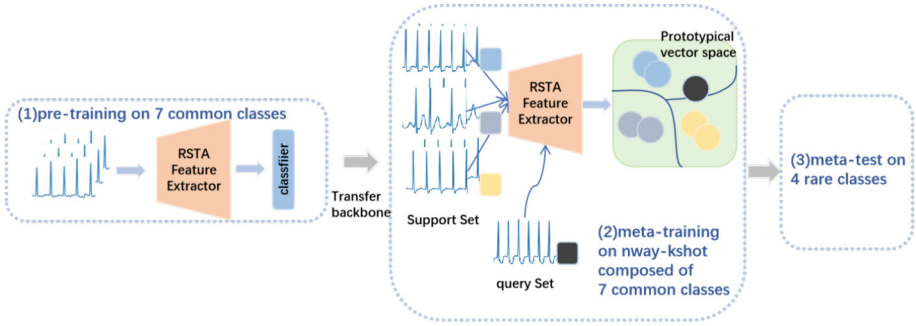

Fig. 1. Architecture of RSTA-ProtoNet for rare arrhythmia classification

2.1 Residual Spatio-Temporal Attention Feature Extractor

The structure of the residual spatio-temporal attention feature extractor (RSTA) is shown in Fig. 2. This architecture interweaves multiple spatial and temporal attention encoders, effectively extracting spatio-temporal features of the ECG signals.

Spatial Attention Encoder: The Squeeze-and-Excitation Residual Block (SE-ResBlock) is constructed to extract the spatial dimension information of the ECG signals, which we refer to as the spatial attention encoder. Each SE-ResBlock consists of two one-dimensional convolutional layers, followed by a batch normalization layer and an SE spatial channel attention module [9]. The SE spatial channel attention module can adaptively adjust the importance of each channel, emphasizing features useful for the classification task while suppressing unimportant information, thereby enhancing the feature's adaptiveness and spatial attention. To introduce nonlinearity and enhance the model's expressive capacity, A ReLU activation function is added after the first convolutional layer of each SE-ResBlock. By constructing multiple SE-ResBlock layers with different numbers of channels and strides, we can gradually elevate the abstraction level of features, enabling the model to capture more complex and deep information. SE-ResNet inherits the advantages of ResNet [8] in terms of ease of optimization and deep construction, and further enhances the adaptive adjustment capability of features by integrating the SE spatial channel attention module.

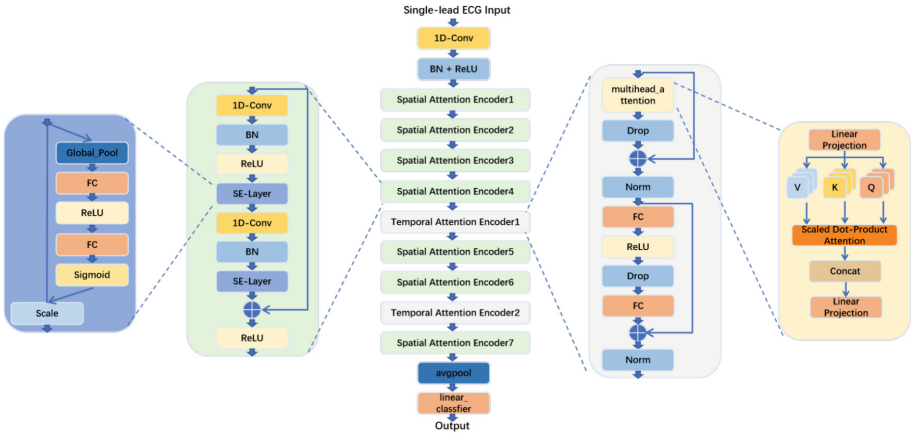

Fig. 2. Architecture of the residual spatio-temporal attention feature extractor

Temporal Attention Encoder: The Transformer encoder layer [20] is constructed to extract temporal dimension information from ECG signals, which we refer to as the temporal attention encoder. Specifically, each temporal attention encoder is composed of two residual structures. Using residual structures can effectively reduce the number of model parameters and alleviate the training difficulty of deep networks. In the first residual structure, the multi-head attention module from the Transformer is introduced, which extends the self-attention mechanism, allowing the model to simultaneously focus on different positions and different representational subspaces in the sequence. A dropout layer is added after the multi-head attention module to prevent overfitting. In the second residual block, a feedforward network consisting of two linear transformation layers and a ReLU activation function is used to further process and optimize the features processed by the self-attention mechanism.

2.2 Meta Training Based on Prototype Network

To address the classification and diagnosis of rare arrhythmia samples, we frame the learning task of rare arrhythmias as an N-way K-shot classification problem [17]. In this setup, a support set S is given which contains N classes each with K samples, *i.e.*, $S = \{(x_1, y_1), (x_2, y_2), ..., (x_{N\times K}, y_{N\times K})\}$, where x_i is a sample and y_i is the corresponding class label. During the meta-training phase, we continually construct N-way K-shot tasks within common arrhythmia classes, using the prototypical network to fine-tune the model, enabling it to rapidly adapt and generalize to learning tasks with a small number of samples. For each class c, we compute its prototype vector p_c, which is the average of the features of all samples in the support set belonging to the same class:

$$p_c = \frac{1}{K} \sum_{(x_i, y_i) \in S_c} f(x_i), \tag{1}$$

where S_c is the set of samples in the support set belonging to class c, and $f(\cdot)$ is the residual spatio-temporal attention feature extractor, used to map the input sample x_i into the feature space. The Euclidean distance is used as the metric to measure the distance between each sample x in the query set and prototype vectors:

$$d(x, p_c) = \sqrt{\sum_{i=1}^{D} (f(x) - p_c)^2}. \tag{2}$$

Then, the model parameters are optimized using the negative log-likelihood loss:

$$L = -\log p(y|x) = -\log \frac{\exp(-d(x, p_y))}{\sum_{c=1}^{N} \exp(-d(x, p_c))}, \tag{3}$$

where $P(y|x)$ is the probability distribution predicted by the model, and y is the true label of sample x.

2.3 Meta Test

During the N-way K-shot testing process, we first select K samples from N rare classes as the support set and input them into the residual spatio-temporal attention feature extractor to obtain the embeddings of the support set $f(X_i), i \in \{1, 2, ..., N \times K\}$. Then, we average these embeddings by class to obtain N prototype feature vector $p_1, p_2, ..., p_N$. We randomly draw Q ECG signal samples from each of the N classes as the query set, obtaining the embeddings of the queries $Q_j = f(X_j), j \in \{1, 2, ..., N \times Q\}$. The final class label for each query is:

$$y_j = \arg\min_i \{d(p_i, Q_j) | i = 1...N\}. \tag{4}$$

3 Experiments and Results

3.1 Dataset

We utilize the 12-lead ECG dataset published by Chapman University and Shaoxing People's Hospital, which contains 12-lead ECG signals from 10,646 patients with a sampling rate of 500 Hz, featuring 11 different rhythms [24]. We filter out incomplete or invalid records that only contain zeros. The final dataset includes 10,588 ECG records: the common classes include 438 atrial flaps (AF), 1,780 atrial fibrillation (AFIB), 397 sinusoidal irregular rhythms (SI), 3,888 sinus bradycardia (SB), 1,825 sinus rhythm (SR), 1,564 sinus tachycardia (ST), and 544 supraventricular tachycardia (SVT). The rare classes include 121 atrial tachycardia (AT), 16 atrioventricular node reentrant tachycardia (AVNRT), 8 atrioventricular reentrant tachycardia (AVRT), and 7 sinus atrium to atrial wandering rhythm (SAAWR). We conduct the following experiments using Lead II.

3.2 Experiment Settings

All experiments and methods in this paper are implemented in Python 3.8.15 and PyTorch 1.9.0, using the AdamW optimizer with a learning rate of 0.00001 and a weight decay of 0.000001. The hardware environment includes an Intel(R) Xeon(R) Gold 6230 CPU @ 2.10 GHz and eight GeForce RTX 2080 Ti graphics cards.

We divide the common classes in a 9:1 ratio to create the training and validation sets for pre-training and meta-training. The network parameters that perform best on the validation set are selected for meta-testing. The rare classes are designated as the meta-testing set for few-shot learning to evaluate the model's rare arrhythmia classification performance. For the few-shot learning experiments, we conduct meta-learning using different N-way K-shot configurations. For instance, the 4way-5shot setup involves testing four rare classes with five samples per class. Due to the limited number of samples, we keep the number of queries as small as possible and perform a large number of meta-tasks during testing to ensure the stability of the experimental results. During the meta-training phase, 200 meta-tasks are executed in each epoch. The results are obtained by repeating the experiment 10 times with 100 meta-tasks each, and they are evaluated using a 95% confidence interval.

3.3 Performance Comparison with Other ECG Few-Shot Methods

In this section, we compare our RSTA-ProtoNet model with several other methods that employ few-shot learning approaches for ECG classification. The methods listed below are meta-trained on common classes and subjected to N-way-K-shot meta-testing on rare classes. The methods for comparison are as follows:

PM-CNN ProtoNet [12]: It utilizes a parallel multi-scale convolutional prototype network for few-shot ECG classification, capturing features at various scales of the ECG signal.
Siamese-CNN [13]: It utilizes a Siamese network with two shared-weight one-dimensional convolutional neural networks to extract and compare feature vectors from input signal pairs for ECG classification.
Relation-SCNN [7]: It is based on a Siamese convolutional framework and combines support and query sets through shared-weight networks into a relation module, using similarity learning for ECG feature learning.

From Table 1, it can be observed that our RSTA-ProtoNet model outperforms other few-shot learning methods in all configurations. Specifically, RSTA-ProtoNet achieves the highest accuracy in both 4-way and 2-way settings. This demonstrates the effectiveness of our model in leveraging the prototype network for few-shot ECG classification, especially in scenarios with limited samples. The comparison highlights the robustness of RSTA-ProtoNet in capturing discriminative features from ECG signals and its ability to generalize well to new classes. Overall, the results validate the superiority of our approach in addressing the challenges of few-shot ECG classification, particularly in the context of rare arrhythmia diagnosis.

Table 1. Comparison results with other methods. The performance is regarded as mean accuracies (%) with 95% confidence interval.

Methods	4way-1shot	4way-5shot	2way-1shot	2way-5shot
PM-CNN ProtoNet [12]	60.14 ± 0.11	71.04 ± 0.16	80.85 ± 0.12	88.20 ± 0.17
Siamese-CNN [13]	62.21 ± 0.21	72.14 ± 0.17	82.72 ± 0.21	90.26 ± 0.11
Relation-SCNN [7]	62.78 ± 0.18	72.50 ± 0.15	83.16 ± 0.09	90.64 ± 0.24
RSTA-ProtoNet	$\mathbf{63.42 \pm 0.15}$	$\mathbf{73.75 \pm 0.11}$	$\mathbf{83.48 \pm 0.16}$	$\mathbf{91.25 \pm 0.14}$

3.4 Ablation Experiments

To study the impact of various modules in the model on the final performance, we compared RSTA-ProtoNet with several variants as follows:

RSTA-ProtoNet (no_SE_Trans): It is a feature extractor composed of 7 residual blocks by removing the SE spatial channel attention module and the multi-head attention module.
RSTA-ProtoNet (no_Trans): It is a feature extractor composed of 7 residual blocks by removing the Transformer Encoder layer while only retaining SE spatial channel attention module into each residual block to enhance channel-wise feature recalibration.
RSTA-ProtoNet (no_SE): It is a feature extractor composed of 7 residual blocks by removing the SE spatial channel attention module while only retaining the Transformer Encoder layer inserted after the fourth and sixth residual blocks to capture long-range dependencies within the ECG signal.

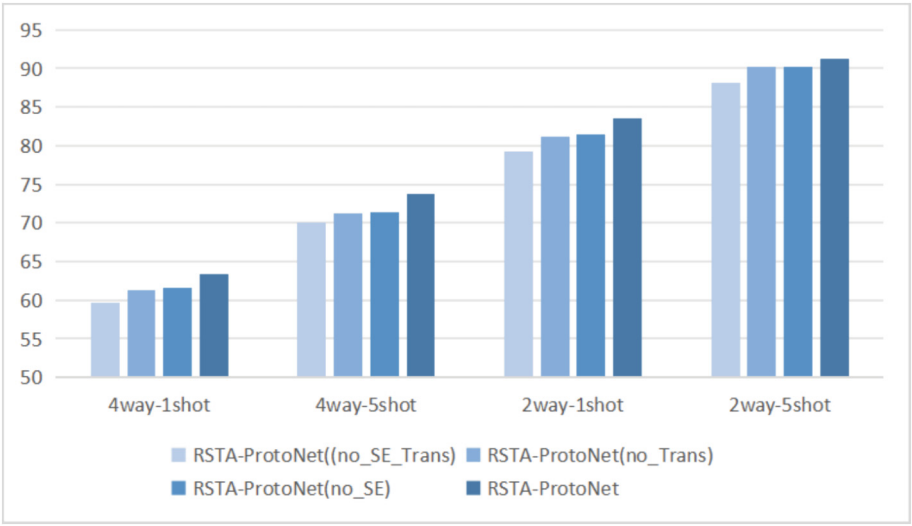

Fig. 3. Experimental results of proposed model and its variants.

As can be seen from Fig. 3, the RSTA-ProtoNet achieved the best performance across different N-way K-shot rare arrhythmia classification tasks. RSTA-ProtoNet (no_SE_Trans) exhibited the poorest performance across all few-shot tasks, significantly lagging behind other models that integrate attention mechanisms. This also demonstrates that the superior performance of the RSTA-ProtoNet model can be attributed to its comprehensive approach to capturing both spatial and temporal features of the ECG signal. The inclusion of SE spatial channel attention module within the residual blocks enhances the model's ability to recalibrate channel-wise features, allowing it to focus on the most relevant spatial characteristics of the ECG signal. This is particularly useful for identifying subtle patterns associated with different types of arrhythmias. Furthermore, the integration of Transformer Encoder layers after the fourth and sixth blocks enables the model to capture long-range dependencies within the ECG signal. This temporal feature extraction is crucial for understanding the dynamics of the heart's electrical activity over time, which is a key aspect of arrhythmia classification. By combining these spatialtemporal attention mechanisms, the RSTA-ProtoNet model can effectively identify and distinguish between various arrhythmias, leading to improved classification performance. This experiment demonstrates the importance of considering both spatial and temporal dimensions in ECG signal analysis for accurate arrhythmia detection.

3.5 Feature Extractor Performance on Common Classes Comparison with Baselines

To validate the effective ECG feature extraction capability of our residual spatio-temporal Attention Feature Extractor, we used the common classes for a 7-class classification task, following the strategy adopted by most researchers [22]. The baseline methods are as follows:

HA-ResNet [6]: It performs dimensional elevation on the network to retrieve potential hidden information in two-dimensional space, enhancing the model's ability to extract meaningful features from ECG signals.
CNN+LSTM [22]: It comprises six convolutional layers and a 128-unit LSTM block, this model effectively captures both local and global features of ECG signals for sequential learning.
ResNet50+LR [23]: This method employs wavelet transform and grayscale conversion of ECG signals to fine-tune a pre-trained ResNet-50 model, with logistic regression as the meta-learner in a stacking integration approach.
MPFNet [5]: It combines one-dimensional raw signals and two-dimensional transformed RP images through an interactive module, enhancing arrhythmia classification by leveraging multi-perspective feature fusion.
HIT [2]: It utilizes a Homeomorphically Irreducible Tree (HIT) based directed acyclic graph for feature generation and maximum absolute pooling for signal decomposition, this method employs SVM for arrhythmia classification.

Table 2 presents the comparison results of our proposed RSTA and other baseline methods on the arrhythmia classification task. It can be observed our

model achieves superior results in both F1 score and accuracy, with an F1 score of 94.60% and an accuracy of 94.81%, outperforming all other models compared. This also fully demonstrates from the results that our model effectively utilizes spatio-temporal attention to extract features from ECG data.

Table 2. Performance comparison between RSTA feature extractor and existing advanced methods in 7-class classification task. The performance is regarded as mean accuracies (%) with 95% confidence interval.

Methods	Acc	Precision	Recall	F1
HA-ResNet+RP [6]	88.24 ± 0.01	87.60 ± 0.04	88.21 ± 0.02	88.03 ± 0.06
HIT [2]	93.04 ± 0.02	90.21 ± 0.01	81.02 ± 0.02	85.34 ± 0.01
CNN+LSTM [22]	92.21 ± 0.01	80.30 ± 0.02	80.22 ± 0.01	80.21 ± 0.02
ResNet50+LR [23]	93.91 ± 0.06	93.72 ± 0.01	94.02 ± 0.02	93.61 ± 0.03
MPFNet [5]	94.30 ± 0.01	94.01 ± 0.02	94.30 ± 0.01	94.21 ± 0.02
RSTA	$\mathbf{94.81 \pm 0.02}$	$\mathbf{94.62 \pm 0.03}$	$\mathbf{94.81 \pm 0.01}$	$\mathbf{94.60 \pm 0.09}$

3.6 Meta Test with Different Classifier

We employ features extracted by the RSTA feature extractor for a 7-class classification task without meta-training, and directly conduct meta-testing to compare the performance differences. Following the methods in [18], we use several classic classification methods: Nearest Neighbor, Cosine Similarity, Logistic Regression, and Prototypical Networks. The experimental results are shown in Table 3. It can be observed that the prototypical network performs excellently in different N-way K-shot tasks, especially in the 2way-5shot and 4way-5shot tasks, with accuracies of 90.11% and 70.82% respectively, surpassing other meta-classifiers. This result further confirms the effectiveness and superiority of the prototypical network in handling few-shot learning tasks.

Table 3. Performance comparison different classifier. The performance is regarded as mean accuracies (%) with 95% confidence interval.

classification methods	4way-1shot	4way-5shot	2way-1shot	2way-5shot
Nearest Neighbor	58.31 ± 0.21	67.62 ± 0.14	80.26 ± 0.12	88.63 ± 0.07
Cosine Similarity	59.05 ± 0.18	68.20 ± 0.05	81.53 ± 0.02	87.36 ± 0.11
Logistic Regression	62.23 ± 0.11	69.33 ± 0.07	82.24 ± 0.16	89.47 ± 0.15
Prototypical Networks	$\mathbf{62.72 \pm 0.16}$	$\mathbf{70.82 \pm 0.11}$	$\mathbf{82.48 \pm 0.15}$	$\mathbf{90.11 \pm 0.13}$

3.7 Visualization Analysis

To further demonstrate the ability of RSTA-ProtoNet to capture key ECG features, we use GRAD-CAM to visualize the attention distribution of our model on several types of ECG arrhythmias. The results are shown in Fig. 4. We select three types of arrhythmia ECGs as examples, including atrial fibrillation (AFIB), sinus bradycardia (SB), and sinus rhythm (SR).

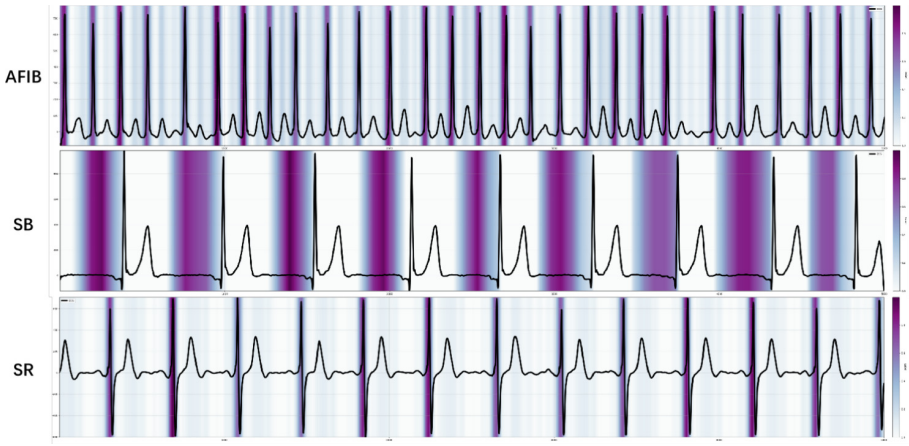

Fig. 4. Heatmaps of different heart rhythms.

It can be observed that the model pays strong attention to the QRS complex, P wave, and T wave intervals in the ECG signal, especially under different types of arrhythmias, the focus is different, which is very consistent with the doctor's diagnostic thinking. For example, for atrial fibrillation, it can be seen from the heatmap that the purple is most intense at the position of the P wave, which is also consistent with the recognition of atrial fibrillation in the ECG in the absence of the P wave and the irregularity of the heart rate; for sinus bradycardia, the model focuses on the slowing of the heart rate and the changes in the intervals of the P wave, QRS complex, and T wave; for sinus rhythm, the model focuses on the normal morphology and intervals of the P wave, QRS complex, and T wave. This detailed attention helps to improve the performance of the model in multi-class arrhythmia classification tasks.

From the heatmaps, it can be seen that our proposed RSTA-ProtoNet based on the SE spatial channel attention module and the multi-head attention module further enhances the model's ability to recognize key features in ECG signals, while improving the feature extraction of spatial channels and the capture of long-range dependencies in time series data. This method not only improves the accuracy and robustness of arrhythmia classification but also provides strong support for the classification of arrhythmias.

4 Conclusion

This paper presents a residual spatio-temporal attention based prototypical network combining residual spatio-temporal attention feature extractor and prototypical network, aimed at improving the classification performance of rare arrhythmias in ECG signals. By applying attention mechanisms in spatial channels and temporal sequences, the model can capture the latent features of ECG signals. For the small number of samples of rare arrhythmias in the dataset, we adopt a few-shot learning approach and evaluate different N-way K-shot strategies for rare classes through meta-testing. Experimental results demonstrate that our model can fully exploit the features of ECG signals, and using only single-lead ECG data, it can effectively enhance the classification accuracy on the Chapman ECG dataset compared to other state-of-the-art methods. In future work, we will focus on exploring the integration of different attention mechanisms to further improve the model's classification capability, and develop few-shot learning methods more suitable for the characteristics of ECG signals. Additionally, we plan to validate and enhance our approach using a broader range of datasets.

Acknowledgments. This work was supported in part by the National Key Research and Development Program of China (No. 2021YFF1201200), the Science and Technology Major Project of Changsha (No. kh2402004). This work was also carried out in part using computing resources at the High Performance Computing Center of Central South University.

References

1. Ahmad, Z., Tabassum, A., Guan, L., Khan, N.M.: ECG heartbeat classification using multimodal fusion. IEEE Access **9**, 100615–100626 (2021)
2. Baygin, M., Tuncer, T., Dogan, S., Tan, R.S., Acharya, U.R.: Automated arrhythmia detection with homeomorphically irreducible tree technique using more than 10,000 individual subject ECG records. Inf. Sci. **575**, 323–337 (2021)
3. Che, C., Zhang, P., Zhu, M., Qu, Y., Jin, B.: Constrained transformer network for ECG signal processing and arrhythmia classification. BMC Med. Inform. Decis. Mak. **21**(1), 184 (2021)
4. Dissanayake, T., Fernando, T., Denman, S., Sridharan, S., Fookes, C.: Generalized generative deep learning models for biosignal synthesis and modality transfer. IEEE J. Biomed. Health Inform. **27**(2), 968–979 (2022)
5. Guan, Y., An, Y., Guo, F., Wang, J.: MPFNet: ECG arrhythmias classification based on multi-perspective feature fusion. In: Guo, X., Mangul, S., Patterson, M., Zelikovsky, A. (eds.) Bioinformatics Research and Applications, ISBRA 2023. LNCS, vol. 14248, pp. 85–96. Springer, Singapore (2023). https://doi.org/10.1007/978-981-99-7074-2_7
6. Guan, Y., An, Y., Xu, J., Liu, N., Wang, J.: HA-ResNet: residual neural network with hidden attention for ECG arrhythmia detection using two-dimensional signal. IEEE/ACM Trans. Comput. Biol. Bioinf. **20**(6), 3389–3398 (2022)

7. Gupta, P., Bhaskarpandit, S., Gupta, M.: Similarity learning based few shot learning for ECG time series classification. In: 2021 Digital Image Computing: Techniques and Applications (DICTA), pp. 1–8. IEEE (2021)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
9. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7132–7141 (2018)
10. Jin, Y., Qin, C., Huang, Y., Zhao, W., Liu, C.: Multi-domain modeling of atrial fibrillation detection with twin attentional convolutional long short-term memory neural networks. Knowl. Based Syst. **193**, 105460 (2020)
11. Jothiaruna, N., et al.: SSDMNV2-FPN: a cardiac disorder classification from 12 lead ECG images using deep neural network. Microprocess. Microsyst. **93**, 104627 (2022)
12. Li, Z., Zhang, H.: Parallel multi-scale convolution based prototypical network for few-shot ECG beats classification. In: 2022 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI), pp. 1–4. IEEE (2022)
13. Li, Z., Wang, H., Liu, X.: A one-dimensional Siamese few-shot learning approach for ECG classification under limited data. In: 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), pp. 455–458. IEEE (2021)
14. Liu, Z., Chen, Y., Zhang, Y., Ran, S., Cheng, C., Yang, G.: Diagnosis of arrhythmias with few abnormal ECG samples using metric-based meta learning. Comput. Biol. Med. **153**, 106465 (2023)
15. Niu, J., Tang, Y., Sun, Z., Zhang, W.: Inter-patient ECG classification with symbolic representations and multi-perspective convolutional neural networks. IEEE J. Biomed. Health Inform. **24**(5), 1321–1332 (2019)
16. Rai, H.M., Chatterjee, K.: Hybrid CNN-LSTM deep learning model and ensemble technique for automatic detection of myocardial infarction using big ECG data. Appl. Intell. **52**(5), 5366–5384 (2022)
17. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. In: Advances in Neural Information Processing Systems, vol. 30 (2017). https://arxiv.org/abs/1703.05175
18. Tian, Y., Wang, Y., Krishnan, D., Tenenbaum, J.B., Isola, P.: Rethinking few-shot image classification: a good embedding is all you need? In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020, Part XIV. LNCS, vol. 12359, pp. 266–282. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58568-6_16
19. Tse, G.: Mechanisms of cardiac arrhythmias. J. Arrhythmia **32**(2), 75–81 (2016)
20. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017). https://arxiv.org/abs/1706.03762
21. Yao, Q., Wang, R., Fan, X., Liu, J., Li, Y.: Multi-class arrhythmia detection from 12-lead varied-length ECG using attention-based time-incremental convolutional neural network. Inf. Fus. **53**, 174–182 (2020)
22. Yildirim, O., Talo, M., Ciaccio, E.J., San Tan, R., Acharya, U.R.: Accurate deep neural network model to detect cardiac arrhythmia on more than 10,000 individual subject ECG records. Comput. Meth. Program. Biomed. **197**, 105740 (2020)

23. Yoon, T., Kang, D.: Multi-modal stacking ensemble for the diagnosis of cardiovascular diseases. J. Pers. Med. **13**(2), 373 (2023)
24. Zheng, J., Zhang, J., Danioko, S., Yao, H., Guo, H., Rakovski, C.: A 12-lead electrocardiogram database for arrhythmia research covering more than 10,000 patients. Sci. Data **7**(1), 48 (2020)

SEMQuant: Extending Sipros-Ensemble with Match-Between-Runs for Comprehensive Quantitative Metaproteomics

Bailu Zhang[1], Shichao Feng[1], Manushi Parajuli[1], Yi Xiong[2], Chongle Pan[2,3(✉)], and Xuan Guo[1(✉)]

[1] Department of Computer Science and Engineering, University of North Texas, Denton, TX 76207, USA
{bailuzhang,fengfeng}@my.unt.edu, xuan.guo@unt.edu
[2] School of Biological Sciences, University of Oklahoma, Norman, OK 73019, USA
{yixiong,cpan}@ou.edu
[3] School of Computer Science, University of Oklahoma, Norman, OK 73019, USA

Abstract. Metaproteomics, utilizing high-throughput LC-MS, offers a profound understanding of microbial communities. Quantitative metaproteomics further enriches this understanding by measuring relative protein abundance and revealing dynamic changes under different conditions. However, the challenge of missing peptide quantification persists in metaproteomics analysis, particularly in data-dependent acquisition mode, where high-intensity precursors for MS2 scans are selected. To tackle this issue, the match-between-runs (MBR) technique is used to transfer peptides between LC-MS runs. Inspired by the benefits of MBR and the need for streamlined metaproteomics data analysis, we developed SEMQuant, an end-to-end software integrating Sipros-Ensemble's robust peptide identifications with IonQuant's MBR function. The experiments show that SEMQuant consistently obtains the highest or second highest number of quantified proteins with notable precision and accuracy. This demonstrates SEMQuant's effectiveness in conducting comprehensive and accurate quantitative metaproteomics analyses across diverse datasets and highlights its potential to propel advancements in microbial community studies. SEMQuant is freely available under the GNU GPL license at https://github.com/Biocomputing-Research-Group/SEMQuant.

Keywords: Metaproteomics · Match-Between-Runs · Label-Free Quantification · Mass Spectrometry

B. Zhang and S. Feng—These authors contributed equally to this work.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-981-97-5087-0_9.

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2024
W. Peng et al. (Eds.): ISBRA 2024, LNBI 14956, pp. 102–115, 2024.
https://doi.org/10.1007/978-981-97-5087-0_9

1 Introduction

Metaproteomics using high-throughput liquid chromatography-mass spectrometry reveals the entire complement and quantity of proteins expressed by the microbial communities. This large-scale characterization of proteins offers crucial insights into functional activities [1], metabolic pathways [33], and community interactions [24]. Quantitative metaproteomics further enhances our understanding of the microbiomes by measuring the relative abundance of proteins to provide insights into the dynamic changes in protein expression under different conditions, such as environmental perturbations [17,38], disease biomarker study [27,37], or experimental treatments [20]. Mass spectrometry-based metaproteomics involves complexities spanning diverse disciplines. Proteins obtained from environmental samples undergo enzymatic digestion with trypsin, which yields numerous peptides, that are subsequently separated via liquid chromatography and analyzed using tandem mass spectrometry (MS/MS). The resulting MS/MS data are then searched against the *in-silico* digested peptide sequences from protein databases of the microbial community to generate a list of peptide-spectrum-matches (PSMs). The abundance of identified proteins is determined by summing the abundance of their identified peptides, obtained from the extracted ion chromatography of the corresponding MS1 scans of PSMs. However, in data-dependent acquisition (DDA) mode, the biased intensity-based sampling of precursor ions for MS/MS scans may result in some peptides not being detected in some LC-MS/MS runs, which may lead to missing peptide quantification. These missing values diminish the effectiveness and accuracy of label-free quantitative proteomics. The match-between-runs (MBR) technique has emerged as a prevalent strategy for addressing this issue. It involves transferring peptides identified in one run to another through inference based on various factors such as m/z, charge state, and retention time (RT).

Two major label-free quantification (LFQ) algorithms utilize the match-between-runs (MBR) approach to address missing values in quantification with DDA data. The first algorithm, MaxLFQ [3], was designed to transfer identified peptide peaks to unsequenced or unidentified peptides in other similar LC-MS runs by matching their mass and RTs. However, transferring peptide peaks among multiple runs using MaxLFQ may result in increased false positives. To overcome this obstacle, IonQuant [35] improves the accuracy and sensitivity of MaxLFQ by controlling the false discovery rate (FDR) of MBR using a mixture model. Moreover, there is a growing trend to streamline proteomics data analysis into a complete pipelines that integrate data processing, peptide identification, filtering, and quantification. For example, MaxLFQ and IonQuant are embedded into MaxQuant [28] and FragPipe [11], respectively. Similarly, software packages such as OpenMS [23], Skyline [21], Proteome discoverer [18], and AlphaPept [15,26] offer integrated core functionalities to support end-to-end quantitative metaproteomics data analysis. In addition, our previous work, called Sipros-Ensemble [8], has demonstrated improved peptide identification results in recent studies [6,7,12,30,31] by integrating multiple score functions [5,13,16,25]. Sipros-Ensemble incorporated ProRata [19,32] to provide LFQ to users.

Unfortunately, ProRata lacks the match-between-runs (MBR) function, which leads to significant missing quantities of identified proteins.

Motivated by the increasing demand to streamline the analysis of vast metaproteomics data and to leverage the superior peptide identifications from Sipros-Ensemble, we have developed a software called SEMQuant. This software extends the quantification capabilities of Sipros-Ensemble by incorporating the MBR function of IonQuant. The experimental results demonstrate that SEMQuant with MBR enabled consistently achieves the highest or second-highest number of quantified proteins with improved accuracy and precision (expressed as coefficient of variance (CV)) among the benchmarking datasets of various microbial complexity. These findings emphasize the potential of SEMQuant to provide more comprehensive and accurate quantitative metaproteomics analysis results.

2 Methods

2.1 Overview of SEMQuant

The SEMQuant workflow overview is illustrated in Fig. 1. It offers a command-line interface designed to streamline the quantitative analysis of metaproteomics data. This integration includes various tools and computational frameworks, such as Raxport for extracting MS1 and MS2 scans, Sipros-Ensemble for peptide and protein identification, and IonQuant with MBR enabled for peptide and protein quantification. In a standard proteomics data analysis workflow using SEMQuant, the process begins with searching MS2 scans against the protein database using Sipros-Ensemble, followed by SEMQuant establishing a separate workspace for each LC-MS run. In this workspace, PSM candidates are sampled and trained by a semi-supervised logistic regression model to re-score the entire set of PSM candidates. The FDR controlling and peptide assembly functions are executed individually for each LC-MS run. Subsequently, SEMQuant generates a configuration file for IonQuant based on the settings of Sipros-Ensemble, and converts the result file containing identified proteins with corresponding PSMs (formatted as "pro2psm.txt") for each run into compatible input files following the format required by IonQuant. These input files include a PSM identification file in XML format and two tab-delimited files containing identified PSMs and peptides with necessary features. IonQuant then traces peaks based on the m/z values of each PSM from corresponding scans in MS1 and transfers peptide identification to other runs. Finally, the quantification results are presented in tab-delimited files comprising all combined identified peptides and proteins across all LC-MS runs.

2.2 Implementation and Software Test

SEMQuant is primarily developed using Python, with additional components implemented in Bash (Unix shell) and tested on the Linux operating system.

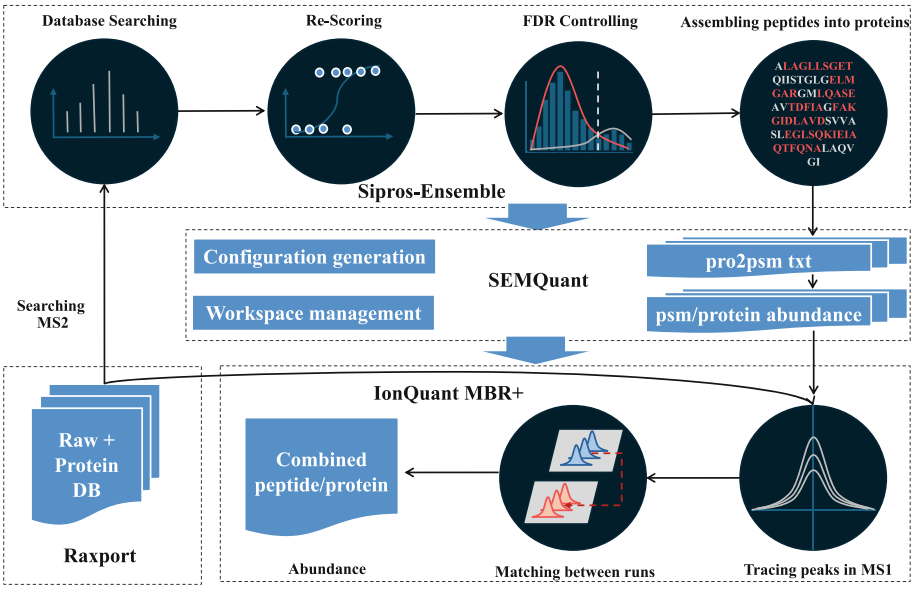

Fig. 1. Overview of the SEMQuant workflow.

Computational tools incorporated into SEMQuant were developed using different programming languages. Raxport relies on the .NET Framework, which can be supported by the Mono platform on Linux based system. The search engine of Sipros-Ensemble was designed using C++, while functionalities such as PSM re-scoring, FDR control, and peptide assembly are executed in Python 2.7. Conversely, IonQuant is built using Java. The environment management for the computational tools mentioned above and workspace management for each LC-MS run are handled using Bash (Unix shell), while the file conversion and configuration generation from Sipros-Ensemble to IonQuant are managed by the Python scripts.

We tested SEMQuant using the OpenMP version of Sipros-Ensemble v1.2 and IonQuant v1.10.12 on two computing platforms: a desktop computer with an Ubuntu 22.04 system equipped with a single 2.3 GHz Intel(R) Xeon(R) Gold 5118 24-core CPU and 32 GB memory and a Linux-based computing node in a supercomputer at the Texas Advanced Computing Center (TACC) featuring two 2.45 GHz AMD EPYC 7763 64-core CPUs and 256 GB memory.

3 Experiments and Results

3.1 Evaluation Measures

We assessed the performance of LFQ results using the "totalPeakArea" in ProRata's output files and the "MaxLFQ intensities" in FragPipe and SEMQuant's output files as the protein abundance measures. In the tabular quantification

result file of FragPipe, there are different measures presented as the quantification result. For a fair comparison, we used the column "MaxLFQ intensities" to conduct post-analysis following experimental design [35] in all the benchmarking datasets. All these results are used to generate the following evaluation measures:

- **Number of quantified proteins**: For each benchmarking dataset, quantified proteins were counted by excluding the identified proteins with zero intensities across all LC-MS runs.
- **Precision**: Median coefficient of variation (CV) across replicates, as designed in the previous study [36]. The lower median CV indicates better precision.
- **Accuracy**: It is determined by comparing the estimated ratio to the ground-truth ratio for the same organism under two different experimental conditions. This measure is particularly applicable to datasets whose proteomes were mixed at the known ratios.

3.2 Benchmark Datasets and Experiment Design

The performance of SEMQuant was evaluated through benchmarking against eight publicly accessible datasets and one in-house dataset. The public datasets comprised one from mixed proteomes of two organisms [14], five from the yeast-UPS1 (Universal Proteomics Standard) mixed proteomes [22], and two from mock microbial communities [10]. Additionally, there was one in-house dataset obtained from a mixed culture comprising four bacterial species.

The two-organism dataset, encompassing 40 runs, was measured on an Orbitrap Fusion Lumos mass spectrometer (Thermo Fisher Scientific). It consists of 20 runs of a mixture of H. sapiens and S. cerevisiae proteomes and 20 runs containing only H. sapiens proteins. This dataset served to assess the number of false positives in transferred peptides across LC-MS runs. The yeast-UPS1 datasets, which mixed yeast proteins with UPS1 concentrations of 2, 4, 10, 25, and 50 fmol/μL, were generated by LTQ Orbitrap Velos mass spectrometers (Thermo Fisher Scientific). These datasets, aimed at evaluating precision and accuracy, include three replicates for each UPS1 concentration and preserve known abundance ratios between any two replicates. For example, in the two replicates, where yeast proteins were mixed with UPS1 at concentrations of 4 fmol/μL and 2 fmol/μL, the expected abundance ratios for yeast and UPS1 proteins are 1:1 and 2:1, respectively. Two datasets from mock communities were utilized to evaluate the performance of SEMQuant on metaproteome samples. Originally labeled as "UNEVEN" in their initial studies, these datasets feature 32 species with varying cell counts and protein biomass levels. In our analysis, one of the mock community datasets is referred to as "Mock-F", comprising 12 fractions analyzed using 2D LC-MS/MS technology. The other dataset, labeled simply as "Mock", includes a single fraction and was processed without the use of 2D LC-MS/MS technology.

The in-house dataset consisted of a mixed culture of four bacterial strains: *Roseburia intestinalis DSM 14610*, *Roseburia hominis DSM 16839*, *Bifidobacterium adolescentis DSM 20083*, and *Bifidobacterium longum subsp. longum*

DSM 20219. The data were acquired on an Orbitrap Exploris 480 (Thermo Fisher Scientific), with three technical replicates.

For the two-organism dataset, yeast-UPS1 datasets, the Mock-F dataset, and the Mock dataset, the protein databases were directly obtained from their original studies or public databases. However, for the in-house dataset, the protein database was generated by translating microbial genomes from NCBI [34] using Prodigal [9].

3.3 Parameters for Benchmarking Software

SEMQuant was compared with Sipros-Ensemble with ProRata [19], and FragPipe with IonQuant. The following software versions were used: Sipros-Ensemble [8] (version 1.2), FragPipe (version 21.1) with MSFragger [11] (version 4.0), Philosopher [29] (version 5.1.0), and IonQuant [36] (version 1.10.12). We maintained the parameters the same for a fair comparison between SEMQuant, Sipros-Ensemble with ProRata, and FragPipe with IonQuant.

The precursor fragment mass tolerance was set to 0.05 Da, and the fragment mass tolerance was set to 0.01 Da. Mass calibration and parameter optimization were activated. The peptide mass range was set from 700 to 7000 Da, and the peptide length was set from 7 to 60. Trypsin/P was employed for the digestion enzyme, with a maximum missed cleavages were set to 1. Oxidation of methionine and acetylation of the protein N-termini were chose as variable modifications, while carbamidomethylation of cysteine was chose as a fixed modification. A maximum of three variable modifications per peptide were permitted. Additionally, for FragPipe, mass calibration and parameter optimization were activated, while all other settings remained at default values.

We estimated the identification false-discovery rate in all datasets using the target-decoy approach [4]. Peptide-spectrum matches (PSMs), peptides, and proteins were filtered at 1% identification FDR.

3.4 Assessment of the False Positives of Transferred Peptides Using the Two-Organism Dataset

The results of quantified peptides across 40 LC-MS runs for the two-organism dataset are summarized in Table 1. This dataset includes 20 runs from an *H. sapiens*-only proteome sample, labeled as "Sample H", and another 20 runs from a mixed culture of *H. sapiens* and *S. cerevisiae*, labeled as "Sample HY". Peptides of *S. cerevisiae* transferred from "HY" to "H" samples were categorized as false positives for calculating the false positive rate (FPR), which is the ratio of *S. cerevisiae* peptides found in "Sample H" to the total number of *S. cerevisiae* peptides identified. As shown in Table 1 and a more straightforward Venn diagram in Supplementary Figures S10, FragPipe and SEMQuant identified more peptides across runs with the MBR enabled. Specifically, SEMQuant outperformed FragPipe in identifying unique peptides for different species. SEMQuant identified 25% more *H. sapiens* peptides and 8.5% more *S. cerevisiae* peptides, albeit with a slight increase in FPR, which stood at 1.25% with MBR mode and 1.33% without MBR mode.

Table 1. Peptides quantified by FragPipe and SEMQuant in analyzing the two-organism dataset.

	FragPipe		SEMQuant	
Total unique H. sapiens peptides[a]	47,526		59,601	
Sample H, MBR−	25,936±495	(54.60%)	22,899±868	(38.42%)
Sample HY, MBR−	25,630±727	(53.90%)	22,816±1,350	(38.28%)
Sample H, MBR+	37,781±443	(79.50%)	34,012±751	(57.07%)
Sample HY, MBR+	37,444±495	(78.80%)	33,967±709	(53.99%)
Total unique S.cerevisiae peptides	4,620		5,014	
Sample H, MBR−	23±7	(0.50%[d])	92±20	(1.83%)
Sample HY, MBR−	2,578±84	(55.80%)	2,127±125	(42.42%)
Sample H, MBR+	113±13	(2.40%)	183±17	(3.65%)
Sample HY, MBR+	3,713±58	(78.50%)	3,187±78	(63.56%)

[a] Peptides were classified as unique *S. cerevisiae* peptides if they exclusively matched proteins from *S. cerevisiae*. Conversely, peptides were identified as unique *H. sapiens* peptides if this criterion was not met.
[b] "MBR+" and "MBR−" indicate that the analysis was performed with and without match-between-runs (MBR), respectively.
[c] The entry value is denoted by $X \pm Y(Z)$, where X represents the average number of unique peptides per run, Y indicates the variation in counts across 20 runs, and Z reflects the ratio of the average count to the total counts.
[d] The numbers underscored are the False Positive Rates.

3.5 Assessment of the Identification and Quantification Results Using the Yeast-UPS1 Datasets

The identification and quantification results for five yeast-UPS1 dataset are shown in Table 2 and Table 3. On average, On average, Sipros-Ensemble outperformed FragPipe by identifying 24.8% more PSMs, 17.3% more peptides, and 8.9% more proteins, offering a more comprehensive profile for the quantification process. As for the quantification results in Table 3, SEMQuant, with the match-between-runs (MBR) feature enabled, consistently reported the highest number of quantified proteins, marking an increase ranging from 1.4% to 10.0% compared to the second best. It also showed superior precision, with improvements between 0.28% and 0.76% in median CV values compared to the second best. Even with MBR disabled, SEMQuant maintained the comparable precision level. In contrast, when Sipros-Ensemble was paired with ProRata, it quantified more proteins than other benchmarked tools. However, the median CV for this combination was over twice as high as others.

We employed the yeast-UPS1 datasets to evaluate the precision of label-free quantification (LFQ) using FDR-controlled match-between-runs (MBR). These datasets include five different UPS1 concentrations (2, 4, 10, 25, and 50 fmol/μL), with each concentration replicated three times. We calculated the protein expression ratios for every possible pairwise comparison among different

Table 2. Benchmarking of identification performance using yeast-UPS1 datasets.

Datasets[a]	2fmol	4fmol	10fmol	25fmol	50fmol
Search[b]	# PSM identifications at PSM FDR 1%				
SE	23,366	24,698	20,438	20,115	18,637
FP	18,888	20,099	15,978	16,039	15,037
	# Peptide identifications at Peptide FDR 1%				
SE	8,033	8,241	7,272	6,464	5,854
FP	6,782	7,019	6,166	5,546	5,036
	# Protein identifications at Protein FDR 1%				
SE	1,666	1,671	1,331	1,202	1,052
FP	1,551	1,522	1,214	1,100	970

[a] Datasets: Yeast proteome mixed with different concentrations (2fmol, 4fmol, 10fmol, 25fmol, 50 fmol) of UPS proteome.
[b] Search engines: SE, Sipros-Ensemble; FP, FragPipe.

Table 3. Benchmarking of quantification performance using yeast-UPS1 datasets.

Pipeline[a]	MBR[b]	2fmol[c]	4fmol	10fmol	25fmol	50fmol
SEMQuant	+	**940 (3.97%)**	**976 (5.78%)**	**801 (12.91%)**	**712 (7.90%)**	**620 (6.36%)**
SEMQuant	−	813 (4.25%)	863 (6.15%)	648 (13.37%)	592 (8.34%)	528 (7.12%)
SE + PR	NaN	927 (12.43%)	950 (13.18%)	668 (30.86%)	625(24.34%)	533 (21.59%)
FragPipe	+	801 (4.92%)	855 (6.23%)	728 (14.20%)	656 (9.86%)	573 (8.45%)
FragPipe	−	654 (5.64%)	694 (6.68%)	540(17.12%)	513(12.27%)	455 (10.72%)

[a] Pipeline: End-to-end pipeline from raw MS files to quantified proteins. SE, Sipros-Ensemble; PR, ProRata.
[b] MBR: "+" means MBR was enable; "−" means MBR was disable; "NaN" means MBR was not available.
[c] Datasets: Yeast proteome mixed with different content (2fmol, 4fmol, 10fmol, 25fmol, 50 fmol) of UPS proteome.
[d] The entry value is denoted by $X(Y)$, where X represents the number of quantified proteins and Y indicates median coefficient of variants. The identification results were filtered at PSM/peptide/protein level FDR 1%.
[e] The best entry was bold and the second-best entry was underlined.

concentrations, such as 2 vs 4, 2 vs 10, 2 vs 25, etc. Given the predetermined mixed ratio of these proteomes, we assessed the accuracy of the LFQ algorithm by comparing the estimated ratios to the actual values. The comparison between 2 and 4 fmol/μL was shown in Fig. 2. As expected, both SEMQuant and FragPipe identified a greater number of proteins with MBR enabled. Protein expression ratios were close to the expected ratios for both FragPipe and SEMQuant, indicating their accurate quantification performance. Further details on the additional pairwise comparisons are provided in the Supplementary Figures (S1-S9).

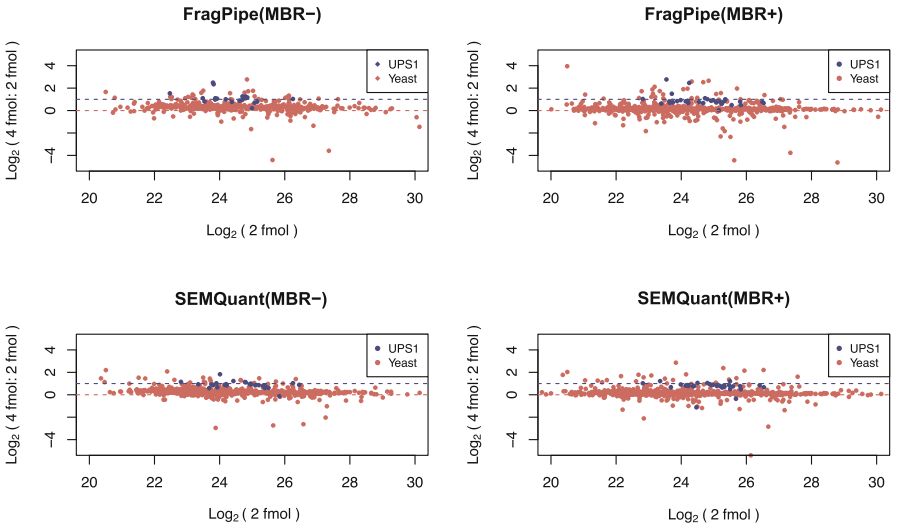

Fig. 2. Ground-truth protein quantification results from Sipros Ensemble and FragPipe using yeast-UPS1 datasets. The known concentrations of UPS1 proteins are 2 $fmol/\mu L$ and 4 $fmol/\mu L$. "MBR-" indicates the match-between-runs (MBR) disabled, and "MBR+" indicates the MBR enabled. Yeast proteins are shown in red and UPS1 proteins are in purple. The horizontal colored dashed lines indicate the expected protein express ratios (0 for yeast and 1 for UPS1).

3.6 Assessment of the Identification and Quantification Results Using the In-House Dataset of a Four-Bacteria Mixed Culture

The identification and quantification results are shown in Table 4. In our analysis, SEMQuant demonstrated superior performance in protein identification when compared to FragPipe, achieving a 6% increase in PSMs, a 1.9% increase in identified peptides, and a modest reduction of 2.5% in identified proteins compared to the second best and/or the best. Notably, when integrated with ProRata, Sipros-Ensemble identified the highest number of proteins, though this came at the cost of precision, as reflected in a median coefficient of variation (CV) of 26.13%. SEMQuant showcased precision comparable to other methods in our benchmarking study. These findings underscore the efficacy of SEMQuant in processing and analyzing complex datasets derived, highlighting its potential as a valuable tool in metaproteomics research.

3.7 Assessment of the Identification and Quantification Results Using Two Mock Community Datasets

The mock community datasets, designed to emulate complex microbial communities, were used in benchmarking the performance of SEMQuant against other platforms. The analysis focused on both identification and quantification metrics across the "Mock-F" and "Mock" datasets, as detailed in Table 5 and Table 4.

Table 4. Benchmarking of identification and quantification performance using the mixed culture dataset of four bacteria.

Pipeline[a]	MBR[b]	#PSMs	#peptides	#proteins	#Quantified proteins(Median CV)
SEMQuant	+	82,190	23,030	2,958	1,970 (6.82%)
SEMQuant	−				1,965 (6.67%)
SE + PR	NaN				**2,038** (26.13%)
FragPipe	+	77,525	22,596	3,035	1,959 (6.66%)
FragPipe	−				1,924 (**6.55%**)

[a]Pipeline: End-to-end pipeline from raw files to quantified proteins. SE, Sipros-Ensemble; PR, ProRata.
[b]MBR: "+" means MBR was enable; "−" means MBR was disable; "NaN" means MBR was not available.
[c]The identification results were filtered at PSM/peptide/protein level FDR 1%.
[d]The best entry was bold and the second-best entry was underlined.

For the "Mock-F" dataset, despite SEMQuant identifying 5.3% fewer PSMs, it generated a greater number of peptides and proteins compared to FragPipe. Similarly, in the "Mock" dataset, SEMQuant outperformed in identifying peptides and proteins but with 2.2% fewer PSMs. The characteristic feature of 2D fractionation led to minimal peptide overlap between fractions, limiting the utility of MBR for cross-fraction peptide quantification. This limitation was evident in the significant disparity between identified and quantified proteins, with quantified proteins constituting less than 30% of those identified in the "Mock-F" dataset. In the contrast, the "Mock" dataset showed a protein coverage exceeding 92%.

Despite SEMQuant with MBR enabled quantified 2.9% fewer proteins than FragPipe with MBR enabled in the "Mock" dataset, it demonstrated a superior median coefficient of variation (CV) that was 4% lower, indicating more precise quantification results. Although Sipros-Ensemble, when combined with ProRata, quantified a significant number of proteins, the precision significantly decreased, with a median CV more than 40% lower than SEMQuant's. These findings underscore SEMQuant's adeptness with MBR in analyzing metaproteomics data (Table 6).

4 Conclusion

We developed a software called SEMQuant, which incorporates Sipros-Ensemble and IonQuant with MBR functionality. This software underwent benchmarking datasets featuring diverse biological experimental conditions, such as mixed organism proteomes with known ratios, metagenome-assembled metaproteomes, and (meta)proteomes with different types of replicates. The experimental results demonstrate that SEMQuant consistently delivers robust performance in terms of the number of quantified proteins, accuracy, and precision.

Table 5. Benchmarking the identification and quantification performance using the "Mock-F" dataset with 12 fractions.

Pipeline[a]	MBR[b]	#PSMs	#peptides	#proteins	#Quantified proteins(Median CV)
SEMQuant	+	144,626	41,169	10,244	**3,302** (61.34%)
SEMQuant	−				3,298 (**60.98%**)
SE + PR	NaN				3,126 (97.03%)
FragPipe	+	152,754	41,152	10,011	3,150 (65.54%)
FragPipe	−				2,595 (63.51%)

[a]Pipeline: End-to-end pipeline from raw files to quantified proteins. SE, Sipros-Ensemble; PR, ProRata.
[b]MBR: "+" means MBR was enable; "−" means MBR was disable; "NaN" means MBR was not available.
[c]The identification results were filtered at PSM/peptide/protein level FDR 1%.
[d]The best entry was bold and the second-best entry was underlined.

Table 6. Benchmarking the identification and quantification performance using the"Mock" dataset.

Pipeline[a]	MBR[b]	#PSMs	#peptides	#proteins	#Quantified proteins(Median CV)
SEMQuant	+	776,700	69,412	11,094	9,817 (**20.96%**)
SEMQuant	−				9,539 (23.77%)
SE + PR	NaN				9,804 (63.94%)
FragPipe	+	794,464	60,212	10,693	**10,118** (24.96%)
FragPipe	−				9,211 (25.10%)

[a]Pipeline: End-to-end pipeline from raw files to quantified proteins. SE, Sipros-Ensemble; PR, ProRata.
[b]MBR: "+" means MBR was enable; "−" means MBR was disable; "NaN" means MBR was not available.
[c]The identification results were filtered at PSM/peptide/protein level FDR 1%.
[d]The best entry was bold and the second-best entry was underlined.

5 Data Availability

The raw MS data and protein databases are available from PRIDE repository with the following dataset identifiers PXD014415 (two-organism), PXD002099 (yeast and UPS1), and PXD006118 (mock uneven U1). The four-bacterial dataset is currently in-house. The datasets produced in this study can be found in the GitHub repository located at https://github.com/Biocomputing-Research-Group/SEMQuant. Additionally, the manuscript and its supporting information files include all other related data. For the two-organism dataset, a protein sequence database of reviewed *H. sapiens* (UP000005640) and *S. cerevisiae* (UP000002311) from UniProt [2] (reviewed sequences only) were used. The yeast-ups1 protein sequence database from UniProtKB/Swiss-Prot (accessed 110615), with UPS1 and cRAP (the common Repository of Adventitious

Proteins, accessed 110403) protein sequences combined. The protein database for the mock uneven dataset is included in the PRIDE repository (PXD006118).

Acknowledgments. This work was supported by the National Library of Medicine, the National Center for Complementary & Integrative Health, and the National Institute of General Medical Sciences of the National Institutes of Health [R15LM013460 and R01AT011618]. The authors acknowledge the department of Research Computing Services at The University of North Texas for providing High Performance Computing resources that have contributed to the research results reported within this paper. URL: https://research.unt.edu/research-services/research-computing.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Abiraami, T., Singh, S., Nain, L.: Soil metaproteomics as a tool for monitoring functional microbial communities: promises and challenges. Rev. Environ. Sci. Bio/Technol. **19**(1), 73–102 (2020)
2. Consortium, U.: Uniprot: a worldwide hub of protein knowledge. Nucleic Acids Res. **47**(D1), D506–D515 (2019)
3. Cox, J., Hein, M.Y., Luber, C.A., Paron, I., Nagaraj, N., Mann, M.: Accurate proteome-wide label-free quantification by delayed normalization and maximal peptide ratio extraction, termed maxlfq. Molec. Cellular Proteomics **13**(9), 2513–2526 (2014)
4. Elias, J.E., Gygi, S.P.: Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. Nat. Methods **4**(3), 207–214 (2007)
5. Eng, J.K., Hoopmann, M.R., Jahan, T.A., Egertson, J.D., Noble, W.S., MacCoss, M.J.: A deeper look into comet-implementation and features. J. Am. Soc. Mass Spectrom. **26**(11), 1865–1874 (2015)
6. Feng, S., et al.: Metalp: an integrative linear programming method for protein inference in metaproteomics. PLoS Comput. Biol. **18**(10), e1010603 (2022)
7. Feng, S., Sterzenbach, R., Guo, X.: Deep learning for peptide identification from metaproteomics datasets. J. Proteomics **247**, 104316 (2021)
8. Guo, X., et al.: Sipros ensemble improves database searching and filtering for complex metaproteomics. Bioinformatics **34**(5), 795–802 (2018)
9. Hyatt, D., Chen, G.L., LoCascio, P.F., Land, M.L., Larimer, F.W., Hauser, L.J.: Prodigal: prokaryotic gene recognition and translation initiation site identification. BMC Bioinform. **11**, 1–11 (2010)
10. Kleiner, M., et al.: Assessing species biomass contributions in microbial communities via metaproteomics. Nat. Commun. **8**(1), 1558 (2017)
11. Kong, A.T., Leprevost, F.V., Avtonomov, D.M., Mellacheruvu, D., Nesvizhskii, A.I.: Msfragger: ultrafast and comprehensive peptide identification in mass spectrometry-based proteomics. Nat. Methods **14**(5), 513–520 (2017)
12. Li, J., Xiong, Y., Feng, S., Pan, C., Guo, X.: Cloudproteoanalyzer: scalable processing of big data from proteomics using cloud computing. Bioinform. Adv. vbae024 (2024)

13. Li, Y., Wang, H., Sun, S., Buckles, B.: Integrating multiple deep learning models to classify disaster scene videos. In: 2020 IEEE High Performance Extreme Computing Conference (2020)
14. Lim, M.Y., Paulo, J.A., Gygi, S.P.: Evaluating false transfer rates from the match-between-runs algorithm with a two-proteome model. J. Proteome Res. **18**(11), 4020–4026 (2019)
15. Liu, Z., Zhang, S., Garrigus, J., Zhao, H.: Genomics-GPU: a benchmark suite for GPU-accelerated genome analysis. In: 2023 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), pp. 178–188. IEEE (2023)
16. Ma, Z.Q., et al.: Idpicker 2.0: improved protein assembly with high discrimination peptide identification filtering. J. Proteome Res. **8**(8), 3872–3881 (2009)
17. Mikan, M.P., et al.: Metaproteomics reveal that rapid perturbations in organic matter prioritize functional restructuring over taxonomy in western arctic ocean microbiomes. ISME J. **14**(1), 39–52 (2020)
18. Orsburn, B.C.: Proteome discoverer-a community enhanced data processing suite for protein informatics. Proteomes **9**(1), 15 (2021)
19. Pan, C., et al.: Prorata: a quantitative proteomics program for accurate protein abundance ratio estimation with confidence interval evaluation. Anal. Chem. **78**(20), 7121–7131 (2006)
20. Pan, S., et al.: Gut microbial protein expression in response to dietary patterns in a controlled feeding study: a metaproteomic approach. Microorganisms **8**(3), 379 (2020)
21. Pino, L.K., Searle, B.C., Bollinger, J.G., Nunn, B., MacLean, B., MacCoss, M.J.: The skyline ecosystem: informatics for quantitative mass spectrometry proteomics. Mass Spectrom. Rev. **39**(3), 229–244 (2020)
22. Pursiheimo, A., et al.: Optimization of statistical methods impact on quantitative proteomics data. J. Proteome Res. **14**(10), 4118–4126 (2015)
23. Röst, H.L.: Openms: a flexible open-source software platform for mass spectrometry data analysis. Nat. Methods **13**(9), 741–748 (2016)
24. Shrestha, H.K., et al.: Metaproteomics reveals insights into microbial structure, interactions, and dynamic regulation in defined communities as they respond to environmental disturbance. BMC Microbiol. **21**, 1–17 (2021)
25. Shteynberg, D., Nesvizhskii, A.I., Moritz, R.L., Deutsch, E.W.: Combining results of multiple search engines in proteomics. Molec. Cellular Proteomics **12**(9), 2383–2393 (2013)
26. Strauss, M.T., et al.: Alphapept: a modern and open framework for ms-based proteomics. Nat. Commun. **15**(1), 2168 (2024)
27. Thuy-Boun, P.S., et al.: Quantitative metaproteomics and activity-based protein profiling of patient fecal microbiome identifies host and microbial serine-type endopeptidase activity associated with ulcerative colitis. Molec. Cellular Proteomics **21**(3) (2022)
28. Tyanova, S., Temu, T., Cox, J.: The maxquant computational platform for mass spectrometry-based shotgun proteomics. Nat. Protoc. **11**(12), 2301–2319 (2016)
29. da Veiga Leprevost, F., et al.: Philosopher: a versatile toolkit for shotgun proteomics data analysis. Nat. Methods **17**(9), 869–870 (2020)
30. Wang, D., et al.: Cross-feedings, competition, and positive and negative synergies in a four-species synthetic community for anaerobic degradation of cellulose to methane. MBio **14**(2), e03189-22 (2023)

31. Wang, S., Feng, S., Pan, C., Guo, X.: Finefdr: fine-grained taxonomy-specific false discovery rates control in metaproteomics. In: 2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 287–292. IEEE (2022)
32. Wang, Y., Ahn, T.H., Li, Z., Pan, C.: Sipros/prorata: a versatile informatics system for quantitative community proteomics. Bioinformatics **29**(16), 2064–2065 (2013)
33. Wang, Y., Zhou, Y., Xiao, X., Zheng, J., Zhou, H.: Metaproteomics: a strategy to study the taxonomy and functionality of the gut microbiota. J. Proteomics **219**, 103737 (2020)
34. Wheeler, D.L., et al.: Database resources of the national center for biotechnology information. Nucleic Acids Res. **36**(suppl_1), D13–D21 (2007)
35. Yu, F., Haynes, S.E., Nesvizhskii, A.I.: Ionquant enables accurate and sensitive label-free quantification with FDR-controlled match-between-runs. Molec. Cellular Proteomics **20** (2021)
36. Yu, F., Haynes, S.E., Teo, G.C., Avtonomov, D.M., Polasky, D.A., Nesvizhskii, A.I.: Fast quantitative analysis of timsTOF PASEF data with MSFragger and IonQuant. Molec. Cellular Proteomics **19**(9), 1575–1585 (2020)
37. Zhang, L., et al.: Islet autoantibody seroconversion in type-1 diabetes is associated with metagenome-assembled genomes in infant gut microbiomes. Nat. Commun. **13**(1), 3551 (2022)
38. Zhu, Y., Liu, N., Yang, Q.: A new approximation algorithm for genomic scaffold filling based on contig. In: 2023 IEEE International Conference on E-health Networking, Application & Services (Healthcom), pp. 72–77. IEEE (2023)

PrSMBooster: Improving the Accuracy of Top-Down Proteoform Characterization Using Deep Learning Rescoring Models

Jiancheng Zhong[1], Chen Yang[1], Maoqi Yuan[1], and Shaokai Wang[2(✉)]

[1] College of Information Science and Engineering, Hunan Normal University, Changsha 410081, China

[2] David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON N2L 3G1, Canada

shaokai.wang@uwaterloo.ca

Abstract. The aim of top-down mass spectrometry-based proteoform identification and characterization is to achieve optimal alignment between mass spectra and proteoforms. Consequently, the accuracy of identification results is crucial. Proteins with multiple primary structure alterations generate various proteoforms, leading to a combinatorial explosion due to their vast numbers. Furthermore, there is no gold set as a reference. So, enhancing the accuracy of identification results remains challenging. We propose a novel rescoring algorithm, PrSMBooster, which employs an ensemble approach. This approach utilizes non-deep models such as XGBoost, Decision Trees, and SVM as weak learners to extract latent features from proteoform spectrum matches. Ultimately, the deep learning model ResNeXt is used for final rescoring. We applied the PrSMBooster rescoring model to 47 independent cross-species datasets. Our comparison with the identification algorithm TopPIC demonstrates that PrSMBooster scores more accurately. In the vast majority of datasets, PrSM increases were observed at 1% FDR. Our findings indicate that PrSMBooster enhances scoring accuracy, reveals more identification results, and exhibits strong generalization capabilities.

Keywords: Proteoform Characterization · Rescoring Model · Deep Learning

1 Introduction

Proteoform characterization is crucial for understanding the diversity of protein functions and their roles in biological processes [1, 2]. A protein combined with multiple primary structure alterations (PSA) come into various proteoforms, leading to the combinational explosion due to its vast quantities. So accurately matching MS/MS spectra with candidate proteoform poses a significant challenge in this field.

Currently, various tools have emerged in the field of top-down proteoform characterization. These methods can be broadly categorized into two types: derivative proteoform database characterization and blind search methods for PSA. The former involves

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2024
W. Peng et al. (Eds.): ISBRA 2024, LNBI 14956, pp. 116–129, 2024.
https://doi.org/10.1007/978-981-97-5087-0_10

searching an extended protein sequence databases containing actual proteoforms by combining theoretical protein sequence libraries with preset modification information. All these methods have advanced the development of TD proteoforms characterization. The ProSight PTM [3] system developed by Zamdborg and others allows users to quantitatively test hypotheses about protein modifications (PTMs) and compare them with intact protein and fragment ion mass data. Karabacak proposed the BIG Mascot search engine [4], which improved fragmentation and data processing methods. Théberge and others developed a top-down analysis platform using the LTQ-Orbitrap mass spectrometer, combined with the BUPID top-down software algorithm [5], to identify and characterize protein variants. Li Li and Zhixin Tian proposed the ProteinGoggle search engine [6], which integrates the iMEF algorithm, allowing the direct use of isotopic envelopes (iEs) in raw mass spectrometry data for the identification of biological molecules, avoiding errors in traditional deisotoping steps, and providing comprehensive quality control for matching precursor and product ions. Solntsev and others enhanced the Global PTM Discovery (G-PTM-D) workflow through the MetaMorpheus software tool [7], using multi-enzyme search and improved algorithms, significantly increasing the speed and accuracy of post-translational modification identification. Toby and others developed the TDPortal software [8], which uses high-resolution Fourier transform mass spectrometry data for protein identification.

On the other hand, PSA blind search methods directly compare experimental mass spectra with theoretical spectra constructed from protein reference sequences. For example, S. Tsai et al. have presented a Precursor Ion Independent Top-Down Algorithm (PIITA) for the de novo sequencing of peptides from top-down tandem mass spectra [9], enabling the identification of proteins by comparing deconvoluted and deisotoped observed spectra against all possible theoretical tandem mass spectra with-in a genomic sequence database. Ari M. Frank et al. proposed a spectral alignment method [10], an algorithm that efficiently finds the optimal alignment path between mass spectra and protein sequences using dynamic programming techniques. It can handle the identification of protein forms with numerous modifications and can be extended to analyze isotope protein form mixtures. Xiaowen Liu et al. also proposed the MS-Align algorithm [11], capable of searching for unknown PTMs. In the paper, they also presented a method for assessing the statistical significance of top-down protein identification. Based on this algorithm, Xiaowen Liu et al. developed MASH Suite Pro [12], which integrates multiple functions into a user-friendly, customizable interface, greatly simplifying and accelerating the interpretation of high-resolution top-down proteomics data. The team continued to improve the algorithm, proposing MS-Align-E [13], capable of identifying expected and unexpected PTMs in hyper-modified proteins. MS-Align-E uses spectral alignment techniques to analyze intact proteins and reveal PTM patterns across the entire protein. They eventually developed an integrated tool, TopPIC [14], which integrates algorithms for protein filtering, spectral alignment, E-value computation, and Bayesian models to characterize unknown amino acid mutations and modifications. It efficiently identifies and characterizes complex protein forms with unknown primary structure changes, such as amino acid mutations and post-translational modifications.

In the field, there are also methods applying machine learning techniques, such as pTop [15] developed by Sun et al., which uses a machine learning model integrating various mass spectrometry features. It employs a SVM for online training to more accurately detect precursor ions to assist identification.

Some researchers have proposed graph-based algorithms, such as a strategy developed by Kira Vyatkina et al. [16], which introduced the concept of T-Bruijn graphs adapted from A-Bruijn graphs widely used in genomics. The subsequent paper used T-Bruijn graphs to compute and assemble de novo amino acid sequences generated by the Twister approach [17]. Jungkap Park et al. developed an open-source software package called Informed-Proteomics and proposed a new graph-based method—the sequence graph—which efficiently explores the combinatorial space of possible proteoforms [18]. Kou et al. proposed a mass graph-based protein form identification tool, TopMG [19], which transforms the protein form identification problem into a mass graph alignment problem and proposed a dynamic programming algorithm to solve it.

Most of methods typically depend on their built-in scoring algorithms, like the MIScore method of bayesian models, spectral probability methods, and dynamic programming. However, for spectra with a high degree of heterogeneity or complexity, it may be difficult to find an accurate match. The confidence scores and rankings offered by proteoform identification algorithms may not precisely correspond to all PrSMs within the target-decoy search strategy. Certain valuable PrSMs might be overlooked by the target-decoy strategy owing to their lower rankings. To address this limitation, we propose the PrSMBooster model, which utilizes an integrated machine learning approach for rescoring. Our model aims to address this by assigning higher scores to PrSMs with substantial MS-based evidence, thereby improving their rankings within the target-decoy search strategy. A comparative analysis was conducted on 47 cross-species datasets to compare the rescoring results obtained using PrSMBooster with those from TopPIC. This comparison revealed differences in PrSM counts, validating the effectiveness of PrSMBooster in improving the accuracy of characterization results.

2 Method

The primary objective of algorithms for proteoform characterization using mass spectrometry technology is typically to identify PrSMs. Our rescoring model is intended to enhance the post-processing phase of this characterization process.

Our approach employs an ensemble deep learning framework for post-processing rescoring. The goal is to maximize the rankings of valuable matches identified as targets while pushing decoy matches towards the lower end of the score rankings. The PrSMBooster method consists of two main components: a feature extraction module and a deep learning-based rescoring module. A schematic representation of the PrSMBooster pipeline is illustrated in Fig. 1.

2.1 Basic Feature Extraction

The input data utilized in PrSMBooster consists of the intermediate results produced by the identification algorithm TopPIC. It is essential to emphasize that, at this stage, the

FDR mechanism in TopPIC remains inactive, resulting in PrSMs without information regarding target and decoy status. Parsing these PrSMs involves extracting basic features:

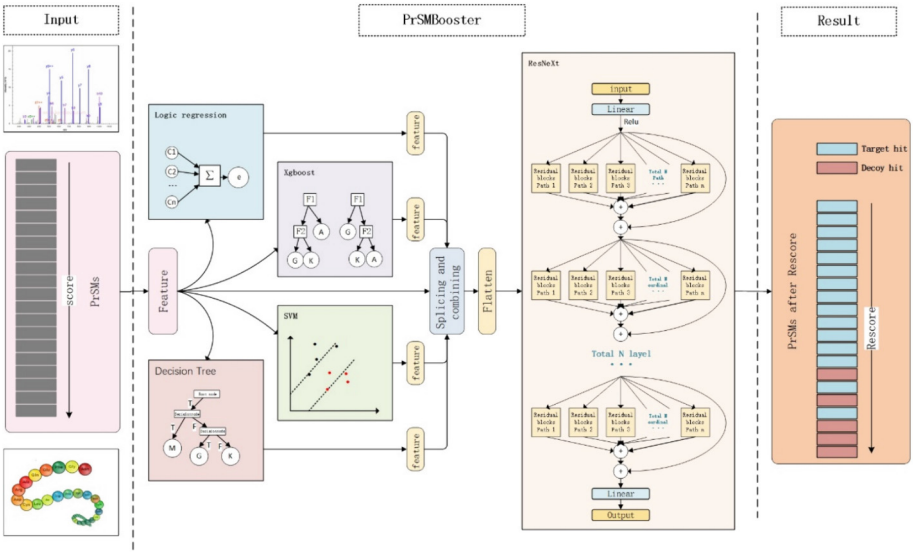

Fig. 1. The Pipeline of PrSMBooster

1. Number of Matched Peaks: This represents the count of peaks in the experimental mass spectrum that correspond to those in the theoretical mass spectrum. A higher number of matched peaks generally indicates more precise identification and characterization.
2. Quantity of Matched Spectral Fragments: This reflects the process where ion fragments, corresponding to the amino acid sequence and potential modifications, are generated during matching. These fragments are then compared to the theoretical fragments derived from known proteins.
3. Count of Variable and Unknown PTMs: This indicates that proteoforms may contain multiple modification sites, affecting the position and intensity of peaks in the mass spectrum.
4. Original Rank Indicator: This serves as a measure of certainty and reliability for the PrSMs reported by TopPIC, akin to a p-value.

2.2 Rescoring Model

PrSMBooster employs an ensemble learning approach to extract latent features from basic features using four classical machine learning algorithms. Once all features are extracted, they are input into the ResNeXt model for rescoring.

Logistic regression, XGBoost, decision trees, and support vector machines are adopted to contribute diverse explanatory capabilities to PrSMBooster by leveraging

multiple models. The integration of multiple models reduces the risk of overfitting associated with individual models and enhances the learning capacity of PrSMBooster across various aspects of the data. Moreover, the ensemble of multiple models facilitates a balance between their respective strengths and weaknesses, allowing PrSMBooster to maintain robustness even in scenarios where individual models may perform suboptimally.

Latent Feature Extraction Module. The inputs for the experiments include basic features such as Number of Matched Peaks, Quantity of Matched Spectral Fragments, Count of Variable and Unknown PTMs, and Original Rank Indicator (i.e., the p-value), a metric used by TopPIC to assess the correctness of PrSMs. These features are combined into a 1-dimensional vector and fed into four classical machine learning models for prediction. The predictions serve as latent features of PrSMs. Here is a detailed description of the four models used:

1. Logic regression. Logistic regression uses the basic features for prediction and outputs a probability value for PrSMs. The parameters are set as follows: the fit_intercept is set to True, meaning the model will calculate the intercept term even if the features are not centered; normalize to False, indicating that the regression variables will not be normalized before performing regression, which typically adjusts features to have a mean of 0 and a variance of 1; copy_X to True, meaning the input data will be copied, ensuring the original data is not modified; n_jobs to None, meaning only one CPU core will be used for computation; and positive to False, meaning the regression coefficients are not constrained to be positive. In the specific experiments, the four basic features are finally predicted to one value after using logic regression that will be used as a potential feature of PrSMs.
2. XGBoost. XGBoost predicts the four basic features as a single value to be used as a latent feature of PrSMs. A parameter values too large for the learning rate and tree depth in XGBoost may lead to overfitting. Similarly, setting the values of learning rate and tree depth too small may result in underfitting. For the experiments, the parameters were set as follows: max_depth was set to 5, limiting the maximum depth of the trees to control model complexity; learning_rate was set to 0.1, scaling the contribution of each tree to prevent overfitting; n_estimators was set to 160, indicating the number of boosting rounds; and silent was set to True, suppressing the output of the model to reduce verbosity.
3. Decision trees. A decision tree is a classical predictive model where the experimental input basic feature predicts a value as a potential feature. For the experiments, the maximum depth of the tree was set to 4, which limits the depth of the tree to prevent overfitting. The criterion used for making decisions was the Gini coefficient, which measures the quality of splits. All other parameters were kept as the default parameters of sklearn, ensuring the standard implementation was used without additional tuning.
4. SVM. SVM is a classical supervised model for solving binary and multiclassification problems and is one of the most popular models in machine learning. The parameters were set as follows: the kernel was set to poly, using a polynomial kernel to capture non-linear relationships; C was set to 0.1, controlling the trade-off between achieving a low training error and a low testing error; gamma was set to 1, defining the influence of a single training example; probability was set to True, enabling probability estimates;

and max_iter was set to 10000, setting the maximum number of iterations for the solver. SVM has good robustness and it is advantageous to use it for extracting potential features.

Final Rescoring Module Uses ResNeXt. The final comprehensive scoring is conducted using ResNeXt. ResNeXt is a deep neural network architecture that builds upon the ResNet architecture [20]. The residual connections in ResNeXt enables the deep learning model to capture nonlinear relationships comprehensively to generate a scoring function for proteoform identification. The advantage of ResNeXt' s group operation over normal ResNet lies in its ability to reduce computational complexity and enhance training speed under identical parameter conditions.

The latent features extracted by the four classical machine learning models and basic features are stacked together as input of ResNeXt. So the ResNeXt operates with an input dimensionality of 8, and the cardinality is set to 8. Cardinality denotes the number of parallel groups within each residual block in ResNeXt, determining the quantity of features selected in a group channel. Each group is composed of a stack of 9 layers of residual blocks. Residual learning can be applied to multiple layers of layers. The residual block on ResNet is defined as follows:

$$y = F(x, W + x) \tag{1}$$

where x is input layer; y is output layer; and F function is represented by the residual map. In addition, each ResNet block consists of two layers (for ResNet-18 and ResNet-34 networks) or three layers (for ResNet-50 and ResNet-101 networks). In this study,

we used the three-layer residual block model. The configuration includes 64 neurons in each layer, and input layers of ResNeXt and residual blocks directly use RelU as an activation functionwith.

ResNeXt employs the Adam Optimizer as its optimizer, and sets batch_size to 100. Compared to the traditional stochastic gradient descent (SGD) optimiser, Adam has the advantages of being able to adaptively adjust the learning rate, adaptive momentum estimation, and relatively more robust choice of hyperparameters. Finally, we set the initial learning rate to 0.001 and set the epoch for model training to 50.

3 Result and Discussion

3.1 Dataset and Preprocessing

The experimental setup included an AMD R7 8-core processor with a clock speed ranging from 3.80 to 4.60 GHz and 32 GB of memory. Python 3.6 was used for experiment design and implementation on PyCharm, utilizing the PyTorch framework and classic machine learning libraries like sklearn.

The majority of these datasets, obtained from the European Bioinformatics Institute, comprised raw mass spectrometry source files covering various species, including human, yeast, tenebrio molitor, pisum sativum, Mus musculus mouse, and Arabidopsis thaliana. Zebrafish dataset from the literature [21]. All datasets are illustrated in Table 1. The datasets were identified using TopPIC(1.6.2), with the sizes of the PrSMs

ranging from 9 to 9019. A total of 59 datasets were employed, with 12 datasets allocated for training, derived from zebrafish and human samples, while the remaining 47 datasets constituted the test set. Preprocessing tools such as MSconvert, TopFD(1.6.2), and TopPIC were employed.

3.2 Evaluation Criteria

Our rescored results are finally ranked using the target-decoy strategy. Initially, they undergo sorting based on score, followed by the calculation of the FDR for each PrSM. Various methods exist for FDR calculation, and in this study, we employ the most common approach. The FDR calculation formula for each PrSM is as follows:

$$FDR_{PrSM\,(i)} = \frac{decoy_i}{target_i} \tag{2}$$

here, $decoy_i$ denotes the count of PrSMs matching decoy sequences prior to the ranking position of this PrSM, while $target_i$ represents the count of PrSMs matching target sequences before the ranking position of this PrSM.

Ultimately, PrSMs that match target sequences and have a ranking position below a specific threshold, along with an FDR less than 0.01, are reported as our final results based on the scoring order.

3.3 Comparison of PrSM Results Before and After Rescoring

The post-processing rescoring method yielded satisfactory performance enhancements across various datasets. We conducted experiments on 47 datasets, noting significant effects. Our use of cross-species input data underscores the versatility of our model, unbounded by species restrictions. Three categories of datasets were divided according to the number of PrSMs of intermediate results reported by TopPIC: those with more than 5000, less than 100, and falling within the range of 100 to 5000.

The Increased Number of PrSMs. Figure 2 illustrates a substantial rise in the number of enhanced outcomes after applying PrSMBooster rescoring, especially prominent in datasets with over 5000+ PrSMs (FB-TO). Across these three large datasets, each initially reporting PrSM counts exceeding 5000, our rescoring algorithm implementation resulted in respective increases of 89, 147, and 162 PrSMs in reported counts.

Some medium-sized datasets also show notable improvement ratios with PrSM-Booster rescoring. Figure 3 displays datasets for Yeast (Yeast) and Tenebrio molitor(TM). In the yeast dataset, except for Yeast_2 and Yeast_5, significant improvements were noted across other datasets. Of these, Yeast_2 had the smallest increase, with 10 enhanced PrSMs, while Yeast_6 had the largest increase, with 87 enhanced PrSMs.

In the medium-sized Tenebrio molitor(TM) dataset, the majority of results showed improvements exceeding 8%. Most datasets reported PrSM counts of at least 30 or more. Notably, datasets TM_1 exhibited the most significant increases, augmenting reported PrSM counts by 104.

Table 1. Description of 59 data sets

Train/Test	Access Number	species	Abbreviations	Raw File name
Train	PXD26137 [22]	human	Human_H	20150927_BM_sort_2E6_CD19_highCD10_techrep01
				20150927_BM_sort_2E6_CD19_highCD10_techrep02
				20150927_BM_sort_2E6_CD19_highCD10_techrep03
				20150927_BM_sort_2E6_CD19_highCD10_techrep04
Train	PXD26141 [22]	human	Human_T	20150930_TDQ_BC_Memory_techrep_01
				20150930_TDQ_BC_Memory_techrep_02
				20150930_TDQ_BC_Memory_techrep_03
				20150930_TDQ_BC_Memory_techrep_04
				20150930_TDQ_BC_Memory_techrep_05
Train	-	Zebrafish	FB_CB	FB_CB_1
				FB_CB_2
				FB_CB_4
Test	-	Zebrafish	FB_TO	FB_TeO_4
				FB_TeO_5
				FB_TeO_6
Test	PXD26178 [22]	Human	Human_L	LCA_RM_20191005_Platelets_F1AB_01
				LCA_RM_20191005_Platelets_F1AB_02
				LCA_RM_20191005_Platelets_F2AB_01
				LCA_RM_20191005_Platelets_F2AB_02

(*continued*)

Table 1. (*continued*)

Train/Test	Access Number	species	Abbreviations	Raw File name
Test	PXD26128 [22]	Human	Human_0	081018_RVG262_PGAFF_RP4H_Neutros_GelFREE_8pF1_FAIMS_CV_-10_018
				081018_RVG262_PGAFF_RP4H_Neutros_GelFREE_8pF1_FAIMS_CV_-20_005
				081018_RVG262_PGAFF_RP4H_Neutros_GelFREE_8pF1_FAIMS_CV_-30_020
				081018_RVG262_PGAFF_RP4H_Neutros_GelFREE_8pF1_FAIMS_CV_-50_019
Test	PXD 29703 [23]	Human	Human_E	Experiment_4_620_F1_01
				Experiment_4_620_F1_02
				Experiment_4_620_F1_03
				Experiment_4_620_F2_01
				Experiment_4_620_F2_02
				Experiment_4_620_F2_03
Test	PXD 42298 [24]	Mus musculus mouse	AH	F1_1
				F1_2
				F2_1
				F2_2
				F3_1
				F4_1
Test	PXD18772 [25]	Tenebrio molitor	TM	ESVO_NSI_5939
				ESVO_NSI_5942
				ESVO_NSI_5951

(*continued*)

Table 1. (*continued*)

Train/Test	Access Number	species	Abbreviations	Raw File name
				ESVO_NSI_5954
				ESVO_NSI_5957
Test	PXD17382 [26]	Pisum sativum	PS	20181010_F1_Ag5_alban001_SA_TCx3_22
				20181010_F1_Ag5_alban001_SA_TCx3_23
				20181010_F1_Ag5_alban001_SA_TCx3_24
				20181010_F1_Ag5_alban001_SA_TCx3_25
				20181010_F1_Ag5_alban001_SA_TCx3_26
				20181010_F1_Ag5_alban001_SA_TCx3_27
				20181010_F1_Ag5_alban001_SA_TCx3_28
Test	PXD34368 [27]	Arabidopsis thaliana	AT	F4_1
				F4_2
				F5_1
				F6_1
				F6_2
				C6
Test	PXD46651 [28]	Yeast	Yeast	Yeast_3
				Yeast_5
				Yeast_6
				Yeast_7
				Yeast_8
				Yeast_9

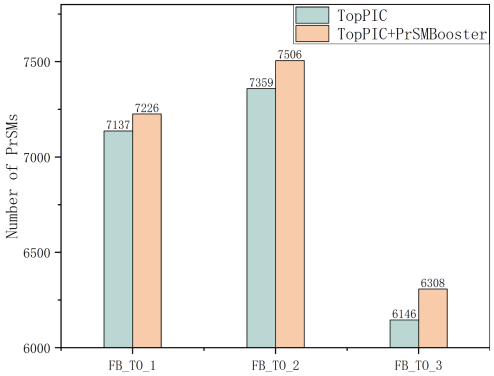

Fig. 2. The number of PrSMs reported in the zebrafish dataset (FB_TO)

Figure 4 illustrates the Pisum sativum (PS) dataset, showing a distinct trend: unlike the advantages observed in large and medium-sized datasets in terms of the number of enhancements, small-sized datasets demonstrate more pronounced advantages in improvement ratios. Excluding dataset PS_4, all other datasets achieved enhancement rates of up to 10%.

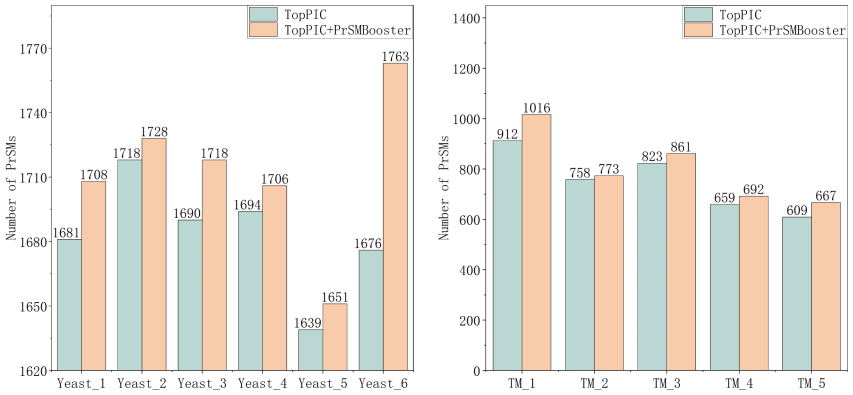

Fig. 3. The number of PrSMs reported in the Yeast dataset (left) and Tenebrio molitor dataset (right)

Analyze the PrSM Improvement Ratio Before and After Using PrSMBooster. We analyze and present our results from an alternative perspective, demonstrating the enhancement and optimization achieved by employing rescoring for individual datasets using the following formula.

$$ratio = \frac{PRSMS_{RE} - PRSMS_{TopPIC}}{PRSMS_{TopPIC}} \tag{3}$$

Here, the original PrSM count denotes the total number of PrSMs reported by the initial identification algorithm. We applied the formula above to assess the improvement

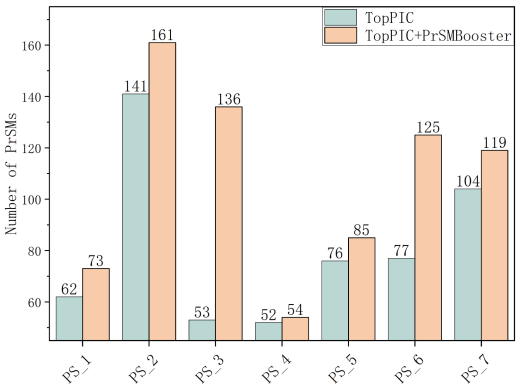

Fig. 4. The number of PrSMs reported in the Pisum sativum dataset (PS)

status of the 47 test datasets, categorizing them based on the extent of enhancement, as depicted in Fig. 5. Remarkably, dataset PS_3 achieved an enhancement ratio of 156%.

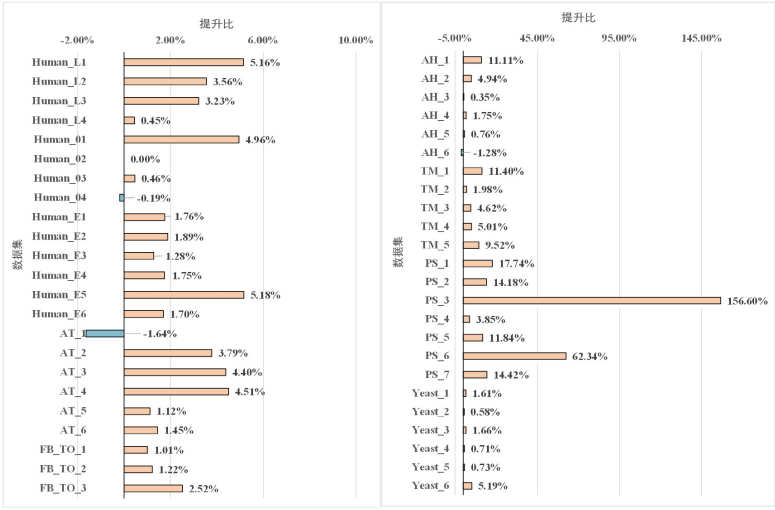

Fig. 5. The improvement ratio of the 47 datasets

4 Conclusion

This study presents PrSMBooster, a novel rescoring algorithm, which leverages classical machine learning models to extract latent features. Subsequently, ResNeXt is employed to integrate both the original and extracted feature information for the final rescoring process. Furthermore, we conducted experiments using PrSMBooster on 47 independent

datasets, demonstrating its capability to augment the number of reported PrSMs. In conclusion, PrSMBooster exhibits promising potential in enhancing the accuracy of proteoform identification and showcasing generalization capabilities.

Acknowledgments. This work has been supported by the National Natural Science Foundation of China, grant no 62372171, the Hunan Provincial Natural Science Foundation of China, grant no. 2023JJ30414. Scientific Research Fund of Hunan Provincial Education Department (No. 23A0100);

Disclosure of Interests. No competing interests.

References

1. Chiribiri, A., Masci, P.G.: From the epicardial vessels to the microcirculation: the coronary vasculature at the crossroad of HFpEF. Cardiovascular Imaging **14**(12), 2334–2336 (2021)
2. Zhong, J., et al.: Proteoform characterization based on top-down mass spectrometry. Brief. Bioinform. **22**(2), 1729–1750 (2021)
3. Zamdborg, L., et al.: ProSight PTM 2.0: improved protein identification and characterization for top-down mass spectrometry. Nucleic Acids Res. **35**(suppl_2), W701–W706 (2007)
4. Karabacak, N.M., et al.: Sensitive and specific identification of wild type and variant proteins from 8 to 669 kDa using top-down mass spectrometry* S. Mol. Cell. Proteomics **8**(4), 846–856 (2009)
5. Théberge, R., Infusini, G., Tong, W., McComb, M.E., Costello, C.E.: Top-down analysis of small plasma proteins using an LTQ-Orbitrap. Potential for mass spectrometry-based clinical assays for transthyretin and hemoglobin. Int. J. Mass Spectrom. **300**(2–3), 130–142 (2011)
6. Li, L., Zhixin, T.: Interpreting raw biological mass spectra using isotopic mass-to-charge ratio and envelope fingerprinting. Rapid Commun. Mass Spectrom. **27**(11), 1267–1277 (2013)
7. Solntsev, S.K., Shortreed, M.R., Frey, B.L., Smith, L.M.: Enhanced global post-translational modification discovery with MetaMorpheus. J. Proteome Res. **17**(5), 1844–1851 (2018)
8. Toby, T.K., et al.: A comprehensive pipeline for translational top-down proteomics from a single blood draw. Nat. Protoc. **14**(1), 119–152 (2019)
9. Tsai, Y.S., et al.: Precursor ion independent algorithm for top-down shotgun proteomics. J. Am. Soc. Mass Spectrom. **20**, 2154–2166 (2009)
10. Frank, A.M., Pesavento, J.J., Mizzen, C.A., Kelleher, N.L., Pevzner, P.A.: Interpreting top-down mass spectra using spectral alignment. Anal. Chem. **80**(7), 2499–2505 (2008)
11. Liu, X., et al.: Protein identification using top-down spectra. Mol. Cell. Proteomics **11**(6), 008524 (2012)
12. Cai, W., et al.: MASH suite pro: a comprehensive software tool for top-down proteomics. Mol. Cell. Proteomics **15**(2), 703–714 (2016)
13. Liu, X., Hengel, S., Wu, S., Tolic, N., Pasa-Tolic, L., Pevzner, P.A.: Identification of ultra-modified proteins using top-down tandem mass spectra. J. Proteome Res. **12**(12), 5830–5838 (2013)
14. Kou, Q., Xun, L., Liu, X.: TopPIC: a software tool for top-down mass spectrometry-based proteoform identification and characterization. Bioinformatics **32**(22), 3495–3497 (2016)
15. Sun, R.X., et al.: pTop 1.0: a high-accuracy and high-efficiency search engine for intact protein identification. Anal. Chem. **88**(6), 3082–3090 (2016)

16. Kira, V., et al.: De novo sequencing of peptides from top-down tandem mass spectra. J. Proteome Res. **14**(11), 4450–4462 (2015)
17. Vyatkina, K., et al.: Top-down analysis of protein samples by de novo sequencing techniques. Bioinformatics **32**(18), 2753–2759 (2016)
18. Park, J., et al.: Informed-Proteomics: open-source software package for top-down proteomics. Nat. Methods **14**(9), 909–914 (2017)
19. Kou, Q., Wu, S., Tolić, N., Paša-Tolić, L., Liu, Y., Liu, X.: A mass graph-based approach for the identification of modified Proteoforms using top-down tandem mass spectra. Bioinformatics **33**(9), 1309–1316 (2017)
20. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
21. Basharat, A.R., Ning, X., Liu, X.: EnvCNN: a convolutional neural network model for evaluating isotopic envelopes in top-down mass-spectral deconvolution. Anal. Chem. **92**(11), 7778–7785 (2020)
22. Melani, R.D., et al.: The blood Proteoform Atlas: a reference map of Proteoforms in human hematopoietic cells. Science **375**(6579), 411–418 (2022)
23. McCool, E.N., et al.: Deep top-down proteomics revealed significant proteoform-level differences between metastatic and nonmetastatic colorectal cancer cells. Sci. Adv. **8**(51), eabq6348 (2022)
24. Wang, Q., Xu, T., Fang, F., Wang, Q., Lundquist, P., Sun, L.: Capillary zone electrophoresis-tandem mass spectrometry for top-down proteomics of mouse brain integral membrane proteins. Anal. Chem. **95**(34), 12590–12594 (2023)
25. Project PXD018772. https://www.ebi.ac.uk/pride/archive/projects/PXD018772. Accessed 20 march 2024
26. Albanese, P., Tamara, S., Saracco, G., Scheltema, R.A., Pagliano, C.: How paired PSII–LHCII supercomplexes mediate the stacking of plant thylakoid membranes unveiled by structural mass-spectrometry. Nat. Commun. **11**(1), 1361 (2020)
27. Wang, Q., Sun, L., Lundquist, P.K.: Large-scale top-down proteomics of the Arabidopsis thaliana leaf and chloroplast proteomes. Proteomics **23**(3–4), 2100377 (2023)
28. Sadeghi, S.A., et al.: Pilot evaluation of the long-term reproducibility of capillary zone electrophoresis–tandem mass spectrometry for top-down proteomics of a complex proteome sample. J. Proteome Res. **23**, 1399–1407 (2024)

FCMEDriver: Identifying Cancer Driver Gene by Combining Mutual Exclusivity of Embedded Features and Optimized Mutation Frequency Score

Sichen Yi[1(✉)] and MinZhu Xie[1,2(✉)]

[1] Key Laboratory of Computing and Stochastic Mathematics (LCSM) (Ministry of Education), School of Mathematics and Statistics, Hunan Normal University, Changsha 410081, China
yisichen1994@163.com, xieminzhu@hunnu.edu.cn

[2] College of Information Science and Engineering, Hunan Normal University, Changsha 410081, China

Abstract. Efficiently identifying cancer driver genes is critical to drug design, cancer diagnosis and treatment. Current unsupervised cancer driver gene prediction approaches mainly exploit mutual exclusivity of mutated driver genes and integrate multi-omics data with gene function networks. Some of them identify driver genes based on the gene features learned by network embedding algorithms. However, these methods are limited to using the mutual exclusivity from original data without considering the mutual exclusivity implanted in the learned features. Additionally, they simply assume that all driver genes have high mutation frequencies. Thus, we propose a novel unsupervised framework FCMEDriver, which utilizes the mutual exclusivity from the learned features and mutation frequency to predict driver genes. In FCMEDriver, a feature clustering algorithm is designed to obtain modules. Based on the modules, our extensive experiments show that the Euclidean distances between learned features are highly related with the mutual exclusivity defined on the original data, and they can reveal more information compared to mutual exclusivity. Thus, we apply the Euclidean distances of learned gene features for each module to calculate a module importance score for each gene. Since the fact that most of driver genes have intermediate mutation frequencies, we design a mutation frequency scoring function for each gene to optimize the existing mutation frequency scoring strategy in which genes with intermediate mutation frequencies are more inclined to obtain similar high scores as those genes with high mutation frequencies. The weighted sum of the module importance score and the mutation frequency score is used to prioritize the genes. The experiment results show that FCMEDriver outperforms other four state-of-the-art methods for cancer driver identification.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-981-97-5087-0_11.

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2024
W. Peng et al. (Eds.): ISBRA 2024, LNBI 14956, pp. 130–141, 2024.
https://doi.org/10.1007/978-981-97-5087-0_11

Keywords: Cancer driver · Clustering algorithm · Mutual exclusivity · embedded feature

1 Introduction

Nowadays, the unsupervised approaches for identifying driver genes can mainly be divided into two classes: methods based on mutation rates and methods based on network [1]. The methods based on mutation rates regard the genes with mutation rates greater than a preset background mutation rate (BMR) as driver genes [2]. These methods include MutSigCV [3] and MuSiC [4]. However, it is difficult to define a reasonable BMR [5], and not all driver genes have a high mutation rate. Network-based methods identify cancer driver genes using gene function networks, which integrate the information of cellular signaling and regulatory pathways, protein-protein or gene-gene interactions. Since the biological functions of cancer cells are always implemented by the interaction of driver genes and some other genes, driver genes are usually enriched in many pathways related to cancer progression [1,5]. HotNet [6] and HotNet2 [7] are two classic network-based methods. Based on the information of mutation frequency of genes and the local topology of gene function network, they use network hot propagation algorithm to detect driver genes. Furthermore, MUFFINN [8] combines the mutation information of individual genes and their neighbors in the gene function network to prioritize genes. Simultaneously, some researchers attempt to integrate multi-omics data with gene function networks to enrich the information of genes. DriverNet [9], DawnRank [10] and Shi et al. [11] integrate gene mutation data and expression data with gene function networks to predict driver genes. EntroRank [1] combines mutation data and subcellular localization data with the gene functional network to realize the driver gene detection.

Based on the confirmation of the research [12] that there are two core properties of driver gene sets corresponding to cancer pathways: (1) high coverage—nearly every patient exhibits at least one mutated driver gene in an important cancer pathway; (2) high mutual exclusivity—mutated driver genes are observed at most once in an important cancer pathway in each patient, Vandin et al. [13] propose a driver gene identification approach Dendrix where they design a weight function that directly measures high coverage and mutual exclusivity to identify the driver gene sets with high weights. Inspired by Dendrix, MeMo [14] first integrates mutation data and expression data into a gene function network to find driver gene modules and then sets an empirically derived p-value to extract the driver gene modules with significant mutual exclusivity as the final identified driver gene modules. MEMCover [15] and CovEx [16] combine mutual exclusivity and mutation frequency to design a trade-off scoring strategy for prioritizing genes.

Besides integrating multi-omics data into gene function networks and exploiting the property of mutual exclusivity, some recent studies have incorporated network embedding algorithms to identify cancer driver genes. RLAG [17] uses network embedding to learn gene feature representations from the gene function network and subcellular localization information. Then it evaluates mutual

exclusivity and mutation frequency for each gene to prioritize potential cancer driver genes. Chu et al. [18] combine the mutation data and the gene function network and learns gene features using a network embedding algorithm, and then use these features to identify cancer driver genes through machine learning algorithms. Doria et al. [19] exploit the gene features of embedding spaces by network embedding to reveal key functional modules associated with cancer.

In general, integrating multi-omics data and adopting new computation technologies could help unsupervised cancer driver gene prediction method achieve better performance, and existing methods have yielded high prediction accuracy. However, as far as we know, current approaches are directly based on original data to define and use mutual exclusivity, and do not explore the important information of mutual exclusivity in embedded features. Meanwhile, the fact that mutation frequencies of most cancer driver genes are concentrated at an intermediate level [20] is not effectively utilized by most existing methods. Herein, we proposed a novel method (called FCMEDriver) to identify cancer driver genes. We first proposed a clustering algorithm based on the cosine similarity to select gene modules with strong associations due to cosine similarity can comprehensively measure the degree of difference of genes in each dimension of embedded features. In each module, we then generated the corresponding subnetwork by mapping the genes in the module to the network and obtained the gene ranging results for each gene set (module) by introducing mutual exclusivity to subnetworks. Since the Euclidean distances of gene embedding features measure the absolute difference of genes in the corresponding dimensions, we adopted the same procedure as above to introduce Euclidean distance instead of mutual exclusivity to obtain another gene ranging results. Meanwhile, we found that these two ranking results have high correlation by calculating Spearman's correlation coefficient, and we verified that the Euclidean distance of the embedded features can show more mutual exclusivity information by the mutual exclusive test. After that, the module importance score based on the Euclidean distance of the embedded features was to evaluate mutual exclusivity of genes in each module. Simultaneously, we designed a novel function to optimize mutation frequency scores for each gene. Finally, FCMEDriver provides a comprehensive evaluation combining the module importance score and mutation frequency score to pick out candidate driver genes. The comprehensive evaluation mainly involves mutual exclusivity, topological centrality of networks, modularity and mutation frequency.

2 Materials and Methods

An overall workflow of FCMEDriver is presented in Fig. 1. We detailed FCMEDriver in the following subsections.

2.1 Datasets and Resources

In this work, the somatic mutation and gene expression data of 230 lung tumor samples, 974 breast tumor samples and 331 prostate tumor samples were

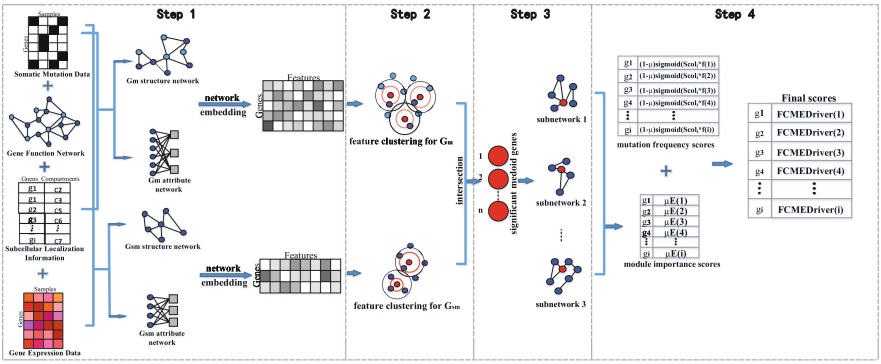

Fig. 1. The workflow of FCMEDriver. Firstly, somatic mutation data, gene expression data and gene function network are combined to screen out specific genes: the genes (called G_m) that mutate in at least one sample are filtered out, and the genes of G_m that are associated with abnormally expressed genes are picked out (called G_{sm}). G_m and G_{sm} are used to construct corresponding gene structure networks and gene attribute networks. Based on these networks, a network embedding algorithm is utilized to learn the corresponding features of G_m and G_{sm}. Secondly, a novel clustering algorithm is performed on the G_m and G_{sm} and we extract the significant medoid genes that present in both the clusters of G_m and G_{sm}. Thirdly, we detect the genes highly correlated with each significant medoid genes to generate a corresponding module and construct a subnetwork for each module and used an entropy-based scoring strategy to calculate an module importance score for each gene in each subnetwork. Finally, FCMEDriver combines the module importance score and the mutation frequency score to prioritize potential cancer driver genes.

downloaded from the TCGA database. The protein subcellular localization information that plays a key role in revealing the cellular functions of proteins (or corresponding genes) was collected from the COMPARTMENTS dataset [21]. There are 11 different subcellular compartments such as cytosol, endoplasmic reticulum and lysosome. The benchmark of cancer driver genes was from the Network of Cancer Genes (NCG6.0) [9]. It includes 2372 cancer driver genes, of which there are 203 driver genes of lung cancer, 221 driver genes of breast cancer and 57 cancer driver genes of prostate cancer. The gene function network containing 14005 genes and 518296 edges was from the protein function interaction networks (Version 2020) based on pathway-informed data analysis [22].

2.2 Networks Construction and Network Embedding

The set of genes that are mutated in at least one sample in the somatic mutation data is denoted by G_m. Previous researches [9–11] have showed that driver genes have a significant impact on the expression of other genes called outlying genes, therefore we first picked out the genes from G_m that are directly associated with outlying genes in the function network, and denoted those genes as G_{sm} that are more likely to be driver genes. Similar to DriverNet [9], we used the

z-score of gene expression data to identify the outlying genes. If the z-score of a gene > 2.0 or < -2.0, the gene is regarded as an outlying gene. Compared to the G_m, G_{sm} doesn't contain any outlying genes. Thus, these associations in the gene function network between the genes in G_{sm} tend to reflect connections between driver genes. Since there are a lot of outlying genes, there are complex associations in the gene function network between the genes in G_m such as the expression impact, the synergistic effect, mutual exclusivity and other biological associations [22].

Since the gene function network is a biomolecular network in which cancer driver genes rarely interact directly with each other, but are more likely to interact with other genes (called heterophilic setting of biomolecular networks) [23]. To reduce the impact of the heterophilic setting of biomolecular network, we construct gene structure networks and gene attribute networks separately. By using network embedding algorithm based on a random-walk for the gene structure network, the structural similarity of gene nodes in the network is measured. The random walk strategy (Node2Vec) [24] with breadth-first sampling (BFS) and depth-first sampling (DFS) has been shown to be effective in capturing the similarity of network neighborhoods of gene nodes. In addition, we introduced subcellular localization information to construct the gene attribute network in which gene nodes establish interactions by localizing to the same subcellular compartments. These two operations can make most of the driver genes that do not interact directly with each other (or are far away from each other) in the biomolecular network have a closer feature representation after network embedding learning. For both G_m and G_{sm}, we constructed gene structure networks and gene attribute networks. The gene structure network of G_m (or G_{sm}) is the subgraph of the gene function network induced by G_m (or G_{sm}) , which is composed of the vertices corresponding to G_m (or G_{sm}) and the edges between the vertices in the gene function network. The gene attribute network of G_m (or G_{sm}) is a bipartite graph. The vertices of left side of the bipartite graph represents the genes and the right side represents the subcellular compartments where the genes are located. A edge between the vertices of a gene and a subcellular compartment means the gene locates at the subcellular compartment. If two genes locate at same subcellular compartments, there may be an interaction between them, and the more subcellular compartments two genes share, the more possibly them interact. The network construction is shown as Step 1 in Fig. 1.

To learn gene features from the gene structure networks and gene attribute networks of G_m and G_{sm}, we used a network embedding algorithm similar to RLAG [17]. The algorithm is based on Node2Vec and consists of two steps: (i) using a random walk to obtain a vector for each gene node in each network (Node2Vec); (ii) using an artificial neural network fed with the vector to learn a gene feature. Specially, we performed multiple random walks on the gene structure networks and the gene attribute networks starting from each gene node, and to make the number of genes in each random walk on the gene structure networks and the gene attribute networks equal, the steps of a random walk on

the gene attribute networks is twice of those on the gene structure networks due to the use of subcellular compartments as a bridge between two genes in the gene attribute networks. For more details of the network embedding algorithm, please refer to RLAG [17]. Therefore, the features of G_m (or G_{sm}) is effectively learned through the construction of networks (gene structure network and gene attribute network of G_{sm}) and network embedding. Based on the learned feature vectors, we can calculate the cosine similarity and the Euclidean distance between two genes.

2.3 Gene Clustering to Detect the Modules of Highly Correlated Genes

We first put forward a clustering algorithm inspired by the VAMB program [25] to detect modules of highly correlated genes based on the above learned features of G_m and G_{sm}. Then, the clusters of G_m and G_{sm} are integrated to detect modules of highly correlate genes.

Clustering G_m and G_{sm} Based on Cosine Similarity. The distance between two genes g_1 and g_2 is defined as 1 minus their cosine similarity and is denoted by $d(g_1, g_2)$. For a given radius r, the neighbors of a gene g are the genes whose distances from the gene is not larger than r, which is denoted by $N(g, r)$, *i.e.* $N(g, r) = \{g_i | d(g, g_i) \leq r\}$. The average distance between gene g and its neighbors is denoted by $d_{\text{avg}}(g, r)$.

Our clustering is a recurring process consists of the following two steps. Step 1: given a suitable radius r, a gene g_i is chosen randomly as the central node (called medoid) and $N(g_i, r)$ forms a cluster C_i. For the cluster, the average distance s between the medoid and other genes in the cluster is calculated, which is $d_{\text{avg}}(g_i, r)$. Step 2: a gene g_j other than g_i in C_i is randomly sampled as a new medoid, and $N(g_j, r)$ corresponds a new cluster C'. For cluster C', the average distance s' between the medoid and other genes in the cluster is calculated, which is $d_{\text{avg}}(g_j, r)$. If $s' < s$, g_i is replaced by g_j, and goto Step 1. Otherwise, Step 2 is repeated until all genes in cluster C have been sampled. When all genes in cluster C have been sampled, let the medoid of C is g, and we obtain a final cluster $N(g, s)$. Remove genes in $N(g, s)$ from the gene set, repeating the above process until there is no gene left. Finally, the clusters that contain only one or two genes are deleted. In the clustering, we followed the principle of high difference between clusters and high closeness of each cluster to determine the values of the radius r for the gene set G_{sm} (G_m) that are set to 0.53 (0,48), 0.5 (0.48), 0.49 (0.47) for lung, breast, prostate cancers datasets, respectively (the details of r decision are in the Supplementary Text S1 and Supplementary Table S1).

Detection of Modules of Highly Correlated Genes. It is possible that a medoid of a gene cluster is a driver gene or a gene that is highly associated with driver genes in the cancer development, and the possibility increases if a gene is both a medoid of a gene cluster of G_{sm} and a medoid of a gene cluster of G_m.

The medoids of $\mathcal{M}_i$ are all genes that play a key role in the networks constructed by G_{sm} and G_m. Based on these genes, in order to explore the mutual exclusivity between driver genes in the embedding space, we focus on finding the modules of the gene set G_{sm}. In detail, let the set of such significant medoid genes be S_m. For each medoid gene g_i in S_m, we detect the genes highly correlated with g_i and construct a corresponding module $\mathcal{M}_i$. For each $\mathcal{M}_i$, we calculate the distances of the genes in the G_{sm} from the medoid gene and take their average value as the threshold, and then remove those genes from the module whose distance to the medoid gene is larger than the threshold. Thus, we measure the degree of association between genes using cosine similarity of feature vectors to show relative differences in direction where those vectors with high similarity are close to each other in most dimensions of feature embedding.

2.4 Module Importance Score with Mutual Exclusivity

Mutual Exclusivity in the Embedding Space. For each module $\mathcal{M}$, mapping the genes in $\mathcal{M}$ to gene function network to construct a subnetwork. For each gene, we count the number of genes with that it has a connection in the subnetworks and is not mutated in the same samples to represent their mutual exclusivity. We then compare the mutual exclusivity of the same genes in each subnetwork and leave only the genes with the maximum mutual exclusivity for a corresponding subnetwork. After above operations (module refinement), the genes of G_{sm} are assigned to the subnetworks where they have the maximum mutual exclusivity and those with a mutual exclusivity of 0 are deleted. All of these genes are then sorted through the mutual exclusivity of corresponding subnetworks (modules).

However, there are two aspects of concern in our study. On the one hand, the detection of modules of highly correlated genes using cosine similarity only focus on the relative differences in the direction of features vectors. It means the cosine similarity actually reflects the difference between the corresponding dimensions without evaluating the absolute numerical difference of the dimensions where some important information may exit. On the other hand, the efficacy of mutual exclusivity heavily relies on the quality of the limited original data features. Therefore, for each gene in each subnetwork, we calculated the sum of Euclidean distance of neighborhood genes that are connected to it to reflect its absolute numerical difference in the subnetworks. Meanwhile, to complete more information about mutual exclusivity, we try to mine mutual exclusivity through the gene features learned by network embedding of multi-omics data and verify our idea by comparing with the mutual exclusivity extracted from the original mutation data. In this process, we first used Euclidean distances to replace the mutual exclusivity of genes and repeated the operations of module refinement and gene ranking. Then, the mutual exclusivity from the original mutation data and Euclidean distances of embedded features to produce two different gene ranking results. Finally, we obtained the Spearman correlation coefficient of 0.9766, 0.9765 and 0.9914 for these two results of lung, breast and prostate cancers respectively by correlation analysis. It shows that the Euclidean

distance of the gene feature representation can measure the mutual exclusivity in the embedding space. Furthermore, we believe that using Euclidean distance to measure the connection of genes in the embedding space has the following two advantages. (i) The Euclidean distance in the embedding space may imply more information of mutual exclusivity due to the fact that the corresponding features is based on multi-omics data learned through network embedding. (ii) The measure of Euclidean distance also considers the influence of heterophilic biomolecular networks in the highly correlated gene modules.

Module Importance Score for Each Genes. Based on the mutual exclusivity, an entropy-based scoring strategy is implemented to measure the module importance of genes in each subnetwork. According to RLAG [17], we calculated the structural entropy (SE) and relative entropy (RE) of each gene to present the genes' centrality and correlations with their neighbors in the subnetworks. The $W_k(i)$ show the importance of the influence of gene i to their neighbors in the subnetwork k. The $SE_k(i)$, $RE_k(ij)$ and $W_k(i)$ are defined as the following formulas.

$$SE_k\left(i\right) = S_k\left(i\right) * log\left(\frac{1}{S_k\left(i\right)}\right) \tag{1}$$

$$RE_k\left(ij\right) = f\left(i\right) * log\left(\frac{f\left(i\right)}{f\left(j\right)}\right) \tag{2}$$

$$W_k\left(i\right) = \sum_{j \neq i} d_e\left(ij\right) * RE_k\left(ij\right) \tag{3}$$

$$d_k\left(ij\right) = \begin{cases} w_{ij}, & e_{ij} \in E_{me} \\ 1, & e_{ij} \in E_{SN}, e_{ij} \notin E_{me} \\ 0, & e_{ij} \notin E_{SN} \end{cases} \tag{4}$$

where $S_k(i)$ is the proportion of the number of the neighbor edges of gene i in the subnetwork k and the number of all edges in the subnetwork. In the Eq. (2), we calculated the $RE_k(ij)$ based on the mutation frequencies ($f(i)$ and $f(j)$) to measure difference degree of gene i to gene j in the subnetwork k. For Eq. (3), the $W_k(i)$ denotes the relationship between gene i and all its neighbors in the subnetwork k. $d_k(ij)$ is the relationship of gene i to gene j in the subnetwork k that is defined by Eq. (4). The $d_k(ij)$ is calculated based on three different types of edge information where e_{ij} represents gene i connect to gene j. E_{SN} is the set of all edges of corresponding subnetwork. In the E_{SN}, we consider the mutual exclusivity of gene sets by searching for the edges between genes that not mutated simultaneously in same samples to define these edges as E_{me}. We calculated the Euclidean distance of feature representations between gene i and j (w_{ij}) to show the mutual exclusivity when they have edge belonging to E_{me}, while $d_k(ij)$ is set to 1 for the other edges in E_{SN}. Obviously, the $d_k(ij)$ corresponding to gene pairs that are not connected in subnetwork k is set to 0. We assumed the driver genes are those with larger difference degree and higher

mutual exclusivity compared to its neighbors. Finally, the module importance score $E(i)$ of gene i is calculated by the formula (5).

$$E(i) = \max\{SE_k(i) * W_k(i) * M(k)\} \tag{5}$$

where $M(k)$ is the ratio of the number of edges in subnetwork k to all edges in the structure network of G_{sm}.

2.5 Comprehensive Scoring to Prioritize Driver Genes

In this section, we combined module importance scores and mutation frequency scores to prioritize potential driver genes.

For the mutation frequency, unlike previous studies [7,17], we held that a gene with a high mutation frequency does not mean it is more likely to be a driver gene. Since most of the cancer driver genes' mutations arise at intermediate frequencies and the high or low mutation frequencies are only a small part [20]. According to this driver mutation property, we designed a scoring strategy in which the mutation frequency values were normalized by a sigmoid function (6). The optimized mutation frequency score not only makes the importance of genes with intermediate mutation frequencies closer to that of genes with high mutation frequencies, but also make the scores of those genes at intermediate mutation frequencies closer to each other. Besides, to show the significance of genes across each tumor sample, we processed and normalized the matrix S by Eq. (7) before using the sigmoid function. The S is a binary mutation matrix of m $*$ n, where m rows represents the number of samples, and n columns ($cols$) represents the number of genes. $S_{j,i} = 1$ when the gene i is mutated in sample j and $S_{j,i} = 0$ otherwise. For each sample (row), we counted the number of genes that mutated in it ($rowSums(S)$) and assigned equal contribution to each gene ($S/rowSums(S)$). For each gene (col), we summed up its contribution in each sample to get its contribution in all mutation samples. Therefore, $Scol$ is a vector of length n where the value of each dimension represents to the total

Table 1. Results of top N genes for the metrics

μ	Lung				Breast				Prostate			
	Precision	Recall	F1score	AUPRC	Precision	Recall	F1score	AUPRC	Precision	Recall	F1score	AUPRC
0	0.250	0.246	0.248	0.0877	0.230	0.208	0.219	0.0901	0.28	0.246	0.262	0.0899
0.1	0.260	0.256	0.258	0.1016	0.230	0.208	0.219	0.0971	0.28	0.246	0.262	0.1150
0.2	0.265	0.261	0.263	0.1151	0.235	0.213	0.223	0.1050	0.30	0.263	0.280	0.1397
0.3	0.265	0.261	0.263	0.1262	0.240	0.217	0.228	0.1124	0.30	0.263	0.280	0.1476
0.4	0.265	0.261	0.263	0.1315	0.245	0.222	0.233	0.1183	0.30	0.263	0.280	0.1607
0.5	**0.280**	**0.276**	**0.278**	**0.1367**	0.240	0.217	0.228	0.1206	0.30	0.263	0.280	0.1639
0.6	0.280	0.276	0.278	0.1339	0.255	0.231	0.242	0.1322	0.32	0.281	0.299	0.1718
0.7	0.270	0.266	0.268	0.1244	0.285	0.258	0.271	0.1480	**0.32**	**0.281**	**0.299**	**0.1769**
0.8	0.250	0.246	0.248	0.1099	**0.290**	**0.262**	**0.276**	**0.1561**	0.32	0.281	0.299	0.1721
0.9	0.210	0.207	0.208	0.0967	0.275	0.249	0.261	0.1533	0.30	0.263	0.280	0.1608
1	0.180	0.177	0.179	0.0881	0.250	0.226	0.238	0.1373	0.28	0.246	0.262	0.1432

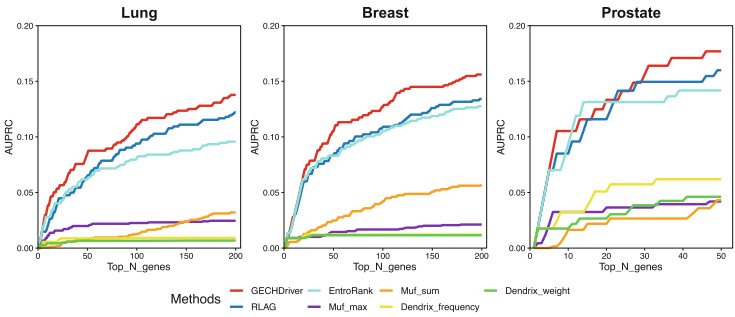

Fig. 2. Comparisons of the performance of each method in predicting three types of cancer based on the AUPRC values for top-ranking genes. The X-axis represents the number of top-ranking genes. The Y-axis represents the score of the given metric.

contribution of each gene in its corresponding mutation samples and the x_i in Eq. (6) is the total contribution of gene i. In particular, the Eq. (6) of sigmoid sets a parameter α of 70.

Then, a weight was assigned to module importance scores and mutation frequency scores to calculate comprehensive scores in the Eq. (8). The weight was to set the parameter μ ranging from 0 to 1 at 0.1 intervals. In this section, we combined the module importance scores with the mutation frequency scores to achieve a comprehensive scoring and ranking of potential driver genes.

$$sigmoid\left(x_i, \alpha\right) = \frac{1}{1 + e^{-\alpha x}} \tag{6}$$

$$Scol = colSums\left(S / rowSums\left(S\right)\right) \tag{7}$$

$$FCMEDriver\left(i\right) = \mu * E\left(i\right) + \left(1 - \mu\right) * sigmoid\left(Scol_i * f\left(i\right), \alpha\right) \tag{8}$$

3 Results and Conclusion

To evaluate the performance in identifying potential driver genes, our method was compared with the four state-of-the-art methods involving Dendrix [13], MUFFINN [8], EntroRank [17] and RLAG [17] to test the performance of our method. The Dendrix is divided into the Dendrix_frequency and Dendrix_weight frameworks by using the different ways of searching the maximum weight subnetworks with mutual exclusiveness and coverage to find driver genes. Since the number of driver genes for lung, breast and prostate cancer in the NGC6.0 benchmark is 203, 221 and 57, the top 200, 200 and 50 genes were chosen respectively from the gene ranking results of every method. The ranking results of three types cancer include 6029, 7105, and 2110 genes, of which the top 200, 200, and 50 are regarded as driver genes (positive samples) and the rest as nondriver genes (negative samples). Because the number of negative genes was much larger than

the number of positive genes, it was more informative to use AUPRC instead of AUC. From the Table 1, the optimal μ is 0.5, 0.8 and 0.7 for lung, breast and prostate cancers to evaluate FCMEDriver method. It not only demonstrates that module importance score plays a significant role in the identification of these three types of cancer driver genes, but also reflects the differences among these cancers. In terms of the benchmark, we used the statistical metrics of Precision, Recall, F1-score and AUPRC to assess the performance of these methods. The Fig. 2 describes the overall metrics of AUPRC for each method on three types of cancer data (The other comparison results see Supplementary Figure S1–S3). In the general, our method FCMEDriver significantly outperforms the other four methods to the three types of cancer data according to the comprehensive metric of AUPRC value. Moreover, we performed enrichment analysis and co-citer analysis (details in Supplementary Text S2-S3).

In this work, we propose a novel method called FCMEDriver to identify cancer driver genes based on integrating multi-omics data, including somatic mutation, gene expression data, gene function network and subcellular localization information. FCMEDriver not only utilizes cosine similarity to pick out modules of highly related genes using a feature clustering algorithm, but also uses Euclidean distance of gene features to explore mutual exclusivity information. Compared with the previous studies, FCMEDriver further focuses on mining mutual exclusivity of embedded features and the optimization of mutation frequency. The results show that our method is the best in identifying cancer driver genes in terms of lung cancer, breast cancer and prostate cancer comparing with other four methods. This not only provides new insights into the studies of identifying cancer driver genes, but also provides a different perspective to reveal the mutual exclusivity of driver genes.

References

1. Song, J., Peng, W., Wang, F.: An entropy-based method for identifying mutual exclusive driver genes in cancer. IEEE/ACM Trans. Comput. Biol. Bioinf. **17**(3), 758–768 (2019)
2. Ding, L., et al.: Somatic mutations affect key pathways in lung adenocarcinoma. Nature **455**(7216), 1069–1075 (2008)
3. Lawrence, M.S., et al.: Mutational heterogeneity in cancer and the search for new cancer-associated genes. Nature **499**(7457), 214–218 (2013)
4. Dees, N.D., et al.: Music: identifying mutational significance in cancer genomes. Genome Res. **22**(8), 1589–1598 (2012)
5. Cheng, F., Zhao, J., Zhao, Z.: Advances in computational approaches for prioritizing driver mutations and significantly mutated genes in cancer genomes. Brief. Bioinform. **17**(4), 642–656 (2016)
6. Vandin, F., Upfal, E., Raphael, B.J.: Algorithms for detecting significantly mutated pathways in cancer. J. Comput. Biol. **18**(3), 507–522 (2011)
7. Leiserson, M.D.M., et al.: Pan-cancer network analysis identifies combinations of rare somatic mutations across pathways and protein complexes. Nat. Genet. **47**(2), 106–114 (2015)

8. Cho, A., Shim, J.E., Kim, E., Supek, F., Lehner, B., Lee, I.: MUFFINN: cancer gene discovery via network analysis of somatic mutation data. Genome Biol. **17**, 1–16 (2016)
9. Bashashati, A., et al.: Drivernet: uncovering the impact of somatic driver mutations on transcriptional networks in cancer. Genome Biol. **13**(12), 1–14 (2012)
10. Hou, J.P., Ma, J.: Dawnrank: discovering personalized driver genes in cancer. Genome Med. **6**(7), 1–16 (2014)
11. Shi, K., Gao, L., Wang, B.: Discovering potential cancer driver genes by an integrated network-based approach. Mol. BioSyst. **12**(9), 2921–2931 (2016)
12. Allan, F., et al.: Comprehensive genomic characterization defines human glioblastoma genes and core pathways. Nature **455**(7216), 1061–1068 (2008)
13. Vandin, F., Upfal, E., Raphael, B.J.: De novo discovery of mutated driver pathways in cancer. Genome Res. **22**(2), 375–385 (2012)
14. Ciriello, G., Cerami, E., Sander, C., Schultz, N.: Mutual exclusivity analysis identifies oncogenic network modules. Genome Res. **22**(2), 398–406 (2012)
15. Kim, Y.A., Cho, D.Y., Dao, P., Przytycka, T.M.: MEMCover: integrated analysis of mutual exclusivity and functional network reveals dysregulated pathways across multiple cancer types. Bioinformatics **31**(12), i284–i292 (2015)
16. Gao, B., Li, G., Liu, J., Li, Y., Huang, X.: Identification of driver modules in pan-cancer via coordinating coverage and exclusivity. Oncotarget **8**(22), 36115 (2017)
17. Peng, W., Yi, S., Dai, W., Wang, J.: Identifying and ranking potential cancer drivers using representation learning on attributed network. Methods **192**, 13–24 (2021)
18. Chu, X., Guan, B., Dai, L., Liu, J., Li, F., Shang, J.: Network embedding framework for driver gene discovery by combining functional and structural information. BMC Genomics **24**(1), 426 (2023)
19. Doria-Belenguer, S., Xenos, A., Ceddia, G., Malod-Dognin, N., Pržulj, N.: A functional analysis of omic network embedding spaces reveals key altered functions in cancer. Bioinformatics **39**(5), btad281 (2023)
20. Lawrence, M.S., et al.: Discovery and saturation analysis of cancer genes across 21 tumour types. Nature **505**(7484), 495–501 (2014)
21. Binder, J.X., et al.: Compartments: unification and visualization of protein subcellular localization evidence. Database **2014** (2014)
22. Guanming, W., Feng, X., Stein, L.: A human functional protein interaction network and its application to cancer data analysis. Genome Biol. **11**(5), 1–23 (2010)
23. Zhang, T., Zhang, S.W., Xie, M.Y., Li, Y.: A novel heterophilic graph diffusion convolutional network for identifying cancer driver genes. Brief. Bioinf. **24**(3), bbad137 (2023)
24. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 855–864 (2016)
25. Nissen, J.N., et al.: Improved metagenome binning and assembly using deep variational autoencoders. Nat. Biotechnol. **39**(5), 555–560 (2021)

Author Index

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2024
W. Peng et al. (Eds.): ISBRA 2024, LNBI 14956, pp. 143–147, 2024.
https://doi.org/10.1007/978-981-97-5087-0