MutTMpredictor: Robust and accurate cascade XGBoost classifier for prediction of mutations in transmembrane proteins

Fang Ge a, Yi-Heng Zhu a, Jian Xu a, Arif Muhammad a,d, Jiangning Song a,b,c,∗, Dong-Jun Yu a,a

a School of Computer Science and Engineering, Nanjing University of Science and Technology, 200 Xiaolinwei, Nanjing 210094, China
b Monash Biomedicine Discovery Institute and Department of Biochemistry and Molecular Biology, Monash University, Melbourne, VIC 3800, Australia
c Monash Centre for Data Science, Faculty of Information Technology, Monash University, Melbourne, VIC 3800, Australia
d School of Systems and Technology, Department of Informatics and System, University of Management and Technology, Lahore, 54770, Pakistan

Abstract
Transmembrane proteins have critical biological functions and play a role in a multitude of cellular processes including cell signaling, transport of molecules and ions across membranes. Approximately 60% of transmembrane proteins are considered as drug targets. Missense mutations in such proteins can lead to many diverse diseases and disorders, such as neurodegenerative diseases and cystic fibrosis. However, there are limited studies on mutations in transmembrane proteins. In this work, we first design a new feature encoding method, termed weight attenuation pos session-specific scoring matrix (WAPSSM), which builds upon the protein evolutionary information. Then, we propose a new mutation prediction algorithm (cascade XGBoost) by leveraging the idea learned from consensus predictors and gcForest. Multi-level experiments illustrate the effectiveness of WAPSSM and cascade XGBoost algorithms. Finally, based on WAPSSM and other three types of features, in combination with the cascade XGBoost algorithm, we develop a new transmembrane protein mutation predictor, named MutTMpredictor. We benchmark the performance of MutTMpredictor against several existing predictors on seven datasets. On the 546 mutations dataset, MutTMpredictor achieves the accuracy (ACC) of 0.9661 and the Matthew’s Correlation Coefficient (MCC) of 0.8950. While on the 67,584 dataset, MutTMpredictor achieves an MCC of 0.7523 and area under curve (AUC) of 0.8746, which are 0.1625 and 0.0801 respectively higher than those of the existing best predictor (fathmm). Besides, MutTMpredictor also outperforms two specific predictors on the Pred-MuHTP datasets. The results suggest that MutTMpredictor can be used as an effective method for predicting and prioritizing missense mutations in transmembrane proteins. The MutTMpredictor webserver and datasets are freely accessible at http://csbio.njust.edu.cn/bioinf/muttmpredictor/ for academic use.

© 2021 The Author(s). Published by Elsevier B.V. on behalf of Research Network of Computational and Structural Biotechnology. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

1. Introduction
With the development and application of next-generation sequencing technology, a large amount of genetic mutation data has been detected, which can be utilized to study the correlation between human genetics and diseases [1]. The rapid identification of pathogenic genetic mutations can help understand the pathogenesis of diseases, which can also contribute to the early detec-
tion of disease and timely treatment [2]. However, it is time-consuming and laborious to distinguish disease-associated mutations from neutral ones using traditional methods. Thus, developing computational techniques to address this is desirable [3].

Missense mutation (MM) is one kind of genetic variants and many methods have been developed to predict disease-associated MM. According to the features utilized, Marwa et al. divided these methods into two categories, i.e. individual and consensus-based predictors [4]. More specifically, there are three subcategories for the individual predictors, including: (1) sequence-based, such as PROVEAN [5] and SIFT [6]; (2) structure-based, for example, SDM [7] and APOGEE [8], and (3) integrated (sequence & structure), for instance, SNAP [9], PolyPhen-1 [10], and PolyPhen-2 [11,12]. Consensus-based predictors often integrate the outputs of several individual predictors (such as PredictSNP [13] and Meta-SNP [14]) and are superior to individual ones in mutation effect prediction [13,15,16].

Among the above predictors, PolyPhen-1 [10] distinguishes specific types of proteins and uses the transmembrane hidden Markov model [17] for membrane region prediction. However, most predictors are not designed for transmembrane proteins which have diverse important function roles, such as cell signaling, cell adhesion, and energy generation [18-22]. It is reported that approximately 60% of membrane proteins are considered as drug targets [23].

Mutations can affect proteins on several levels. They can disrupt structural stability, affect folding/modular degradation, and lead to improper transportation or the emergence of toxic conformations [24]. This influence manner also appears in membrane proteins. Specifically, the α-helix subsequence and helicity in membrane proteins are critical in interacting with lipids and/or other helices to form a higher structure [25]. This procedure is called folding stage. In order to ensure that membrane protein is assembled correctly, cells have already evolved coordination and quality control systems, namely protein deposition networks [26,27]. However, even with the above systems, assembly efficiencies of many proteins at normal body temperature are still less than 50% [28], leading to the destruction of protein deposition network [25]. In membrane protein mutation research field, formation of Tertiary (misfolding) and Quaternary (abnormal oligomerization) structures are defined as mismatches [25]. Species with misassembled mutations may eventually cause disease by affecting the normal flow of proteins, leading to a decrease in the number of functional proteins on target membrane; the retention of toxic functional gain proteins in endoplasmic reticulum; and/or overwhelming estrogen receptors. The quality control component then triggers an unfolded protein response and apoptosis [29]. On the other hand, mismatched membrane proteins may reach their target membrane, resulting in abnormal function of target and further leading to diseases, such as cardiopathies, neurological diseases, and cancer [30,31]. Furthermore, even one single missense mutation occurring on the critical site of α-helix can be deleterious to protein folding and/or its biological function [25]. Thus, the study of transmembrane protein mutations is conducive to a clearer understanding of its functional mechanism, which is also essential for diagnosing and treating specific diseases.

Recently, a variety of database or methods have been developed to store or predict mutations in transmembrane proteins. The MuthTP (mutations in human transmembrane proteins) database were developed to deposit and retrieve mutations in such proteins [32]. Besides, BorodaTM [33], Pred-MuthTP [31], mCSM-membrane [34], and TMSNP [35] are specific predictors for mutations prediction in such proteins. Specifically, in 2019, Kuldalai-samy et al. collected MM in transmembrane proteins from HumSavar (http://www.uniprot.org/docs/humsavvar), SwissVar [36], 1000 Genomes [37], ExAC [38], COSMIC [39], and ClinVar [40]. After selecting, mapping, and extracting procedures, MuthTP was constructed, which stored MM, insertion, and deletion mutations [32]. Furthermore, BorodaTM was developed, which was the first method specifically designed for mutations prediction in transmembrane proteins [33]. The training and test proteins in BorodaTM [33] all have 3D structures in the PDB database [41]. Pred-MuthTP [31] was proposed and four specific datasets were available for mutation prediction at its website. In 2020, mCSM-membrane was developed, which utilized graph-based signatures, protein geometry, and physicochemical properties to predict the pathogenicity of mutations [34]. However, mCSM-membrane could only predict mutations in proteins with known 3D structures in PDB database [41]. In 2021, TMSNP database was constructed, which comprised 196,705 non-pathogenic, 2,624 pathogenic, and 437 likely pathogenic mutations, respectively in transmembrane proteins [35].

Although numerous methods have been developed, the concept and characteristics of “mutation microenvironment information” are not considered generally, which may be useful for improving the prediction performance. Second, few methods take the outputs of individual predictors, which can make the model more robust. Third, the feature reutilized in gForest [42] and DenseNet [43] can be leveraged to significantly improve the model's prediction performance. However, such “gained feature” is not used in most methods.

In this study, some further improvements are made with respect to the following main aspects. Firstly, we propose a weight attenuation feature based on evolutionary information, termed WAPSSM (weight attenuation position-specific scoring matrix). This feature extracts the microenvironment information, along with different weights to measure different influence on the mutation site. Second, we leverage the idea inspired by consensus predictors that take the outputs of individual predictors as part of the feature vector. Third, we utilize the previous level's output as the input to the following level, which is learned from gForest [42]. Building upon such advantages, we propose the cascade XGBoost-based algorithm and develop a powerful predictor, termed MutTMPredictor, for improving prediction of transmembrane protein mutations. Extensive benchmarking experiments demonstrate the effectiveness of WAPSSM and cascade XGBoost algorithm. Moreover, performance comparison with several existing predictors on six different datasets and blind test using a third-party dataset show that MutTPredictor is effective for predicting mutations in transmembrane proteins.

2. Mutation datasets and feature representation

2.1. Benchmark datasets

In this work, we utilized seven datasets to evaluate and compare the performance of different predictors. (1) 546 mutations dataset: we collected this dataset from BorodaTM [33], which comprised 154 neutral and 392 disease-associated missense mutations in 64 transmembrane proteins. Notably, these proteins have 1 to 13 transmembrane α-helices and known 3D structures in the PDB [41]. (2) Pred-MuthTP dataset: mutations in Pred-MuthTP [31] were collected from MuthTP [32] by retaining mutations present in at least two databases and removing the sequence redundancy using CD-HIT [44]. In addition, three sub-datasets (i.e. “Cytoplasmic or inside”, “Membrane”, and “Extracellular or outside”) were constructed based on different topological regions where the mutations were located. (3) 67,584 mutations dataset: the original dataset comprised 29,033 disease-associated and 38,580 neutral missense mutations. Some proteins and mutations were deleted, because the wild-type amino acids at given positions
did not match those in UniProt [45]. Accordingly, we removed 13 disease-associated and 116 neutral mutations. (4) TMSNP database: TMSNP [35] stored a certain set of membrane proteins taken from TMSNP data- base: TMSNP [35] stored a certain set of membrane proteins taken from UniProt [45]. It retrieved the disease-causing/pathogenic mutations occurring in transmembrane helical regions from ClinVar [40] and SwissVar [36] and nonpathogenic mutations/allele frequency from GnomAD [46] and ClinVar [40]. A detailed summary of these seven benchmark datasets is provided in Table 1.

### 2.2. Feature representation and selection

In this work, each mutation was represented by a feature vector in a multi-dimensional information space. Herein, we extracted four different types of features, including features collected from BorodaTM [33], features based on evolutionary information, outputs of four individual mutation analysis tools, and outputs of three sub_XGBoost models. A detailed description of these features is provided in the following subsections.

#### 2.2.1. Features collected from BorodaTM

BorodaTM [33] utilized CompoMug [47] to extract features for each mutation, including protein sequence-based, structure-based, and energy-based features. Specifically, as for sequence-based characteristics, 12 types of physicochemical properties (such as isoelectric point, ZIMj680104 and net charge, KLEP840101) were extracted from AAindex [48]. Besides, structure-based characteristics mainly contain secondary structure, residue solvent exposure, and number of residues contact in 5Å proximity. Energy-based characteristics mainly comprise Van-der-Waals, entropic, and energy within 5Å centered by the mutant site. We collected the above feature descriptors from BorodaTM [33], named as “Original”. A detailed list of “Original” features can be found in Supplementary Table S1.

#### 2.2.2. Gaussian WAPSSM

Some characteristics have been frequently used for representing proteins, e.g. position-specific scoring matrix (PSSM) [49]. Numerous research works have proven its discriminative capability for sequence classification problems in bioinformatics, such as protein-DNA binding site prediction [50] and transmembrane protein prediction [51]. We also used PSSM as part of the feature vector. PSI-BLAST [49] was utilized to generate the PSSM characteristics by searching each query against the SWISS-PROT database [45]. Specifically, PSI-BLAST [49] can generate multiple sequence alignment (MSA) to retrieve the biological evolutionary information of the closest relatives for the query protein, by setting the e-cutoff value to 1e-3, number of iterations to 3, and substitution score matrix to BLOSUM62 [52].

On the basis of evolutionary information (i.e. PSSM), we need to take the following two aspects into consideration. First, the characteristic of a mutation site \( i \) should consist of its own and neighboring residues (i.e. the “microenvironment”). Second, different neighboring residues may have a diverse impact on the mutation site \( i \). That is, a neighboring residue located further away from the centered residue \( i \) would have a lesser impact. In contrast, those located in the closer proximity would have a more significant impact. In light of above two aspects, we developed a weight attenuation PSSM (named WAPSSM) extraction algorithm by combining the original PSSM matrix and the concept of weight attenuation.

The obtained wPSSM \( \mathbf{w} \) comprises three parts of weighted sub-vectors, including \( [w^k \cdot e_{i,k}, \ldots, w^i \cdot e_{i-1,k}, \ldots, w^k \cdot e_{i-1,k}]_{1:k} \), \( w^k \cdot e_{i,k} \), and \( [w^k \cdot e_{i,k}, \ldots, w^i \cdot e_{i-1,k}, \ldots, w^k \cdot e_{i-1,k}]_{1:k} \), where \( w^k \cdot e_i \) represents the PSSM feature of the mutation site \( i \) itself, whereas \( [w^k \cdot e_{i,k}, \ldots, w^i \cdot e_{i-1,k}, \ldots, w^k \cdot e_{i-1,k}]_{1:k} \) are the weighted local microenvironment PSSM features before and after the mutation site \( i \). According to our preliminary analyses, herein we set \( k = 3 \).

#### Gaussian WAPSSM algorithm

**Input**: the original PSSM \( n \times 20 \) matrix and half of the microenvironment size \( k \)

**Step 1**: Use the sigmoid function \( h(x) = \frac{1}{1 + e^{-x}} \) to transform the original PSSM element values into range (0, 1). Thereafter, we obtain the PSSM matrix:

\[
\mathbf{e}_i = [e^1_{i,1}, \ldots, e^{20}_{i,1}],
\]

where \( \mathbf{e}_i = [e^1_{i,1}, \ldots, e^{20}_{i,1}] \) and \( n \) is the protein sequence length. Herein, we used the numerical codes 1, 2, 3, ... , 20 to represent 20 native amino acid types.

**Step 2**: For each mutation site \( i \), collect its microenvironment-related local PSSM feature, formulated as \( \mathbf{imPSSM}_{(2k+1) 	imes 20} \) where \( 2k + 1 \) is the number of residues in the microenvironment centered at the mutation site \( i \).

**Step 3**: In order to measure the impact of residues within different distances on the mutation site, the Gaussian weight vector \( \mathbf{w} \) is utilized to measure the degree of such attenuation. Herein, we set

\[
\mathbf{w} : h(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \quad (\mu = 0, \sigma = 1).
\]

**Step 4**: Building upon Step 3,

\[
\mathbf{w} = [w^k \ldots, w^i \ldots, w^k \ldots]_{1:20}
\]

i.e. \( w^k = h(-k), \ldots, w^i = 1, \ldots, w^k = h(k) \) can be obtained.

**Step 5**: Equip the local feature \( \mathbf{imPSSM}_{(2k+1) 	imes 20} \) with the Gaussian weight vector \( \mathbf{w} \). We would obtain the WAPSSM feature \( \mathbf{wpPSSM} = [w^k \cdot \mathbf{e}_{i,k}, \ldots, w^i \cdot \mathbf{e}_{i,k}, \ldots, w^k \cdot \mathbf{e}_{i,k}] \).

**Output**: WAPSSM feature

2.2.3. Outputs of four individual mutation analysis tools

In this work, we took advantage of consensus methods by taking the outputs of several individual mutation analysis tools as part of the feature vector. Specifically, we first fed the mutations into PROVEAN [5], PolyPhen-2 [12], and fathmm [53] web servers. Subsequently following the prediction, we downloaded the respective output files and extracted the results predicted by each tool. This kind of feature is named as “Individuals’ output”.

2.2.4. Outputs of three sub_XGBoost models and the cascade XGBoost algorithm

In the gcForest work, Zhou et al. designed a cascade forest algorithm, which took the outputs of the previous level as part of the

### Table 1

| Order | Name              | Number of mutations (number of proteins with mutations) | Note                          |
|-------|-------------------|--------------------------------------------------------|-------------------------------|
| 1     | 546 mutations     | 392 (31)                                               | From BorodaTM [33]            |
| 2     | Whole data*       | 11,846 (1,014)                                         | From Pred-MutHTP [31]         |
| 3     | Cytoplasmic       | 4,416 (625)                                            | From Pred-MutHTP [31]         |
| 4     | Membrane          | 2,421 (454)                                            | From Pred-MutHTP [31]         |
| 5     | Extracellular      | 4,948 (677)                                            | From Pred-MutHTP [31]         |
| 6     | TMSNP             | 29,020 (2,581)                                         | From TMSNP [35]               |

Note: For each dataset the number of proteins with mutations is given in parenthesis. Whole data* (Pred-MutHTP): all mutations in human transmembrane proteins are considered.
input for the following level [42]. Herein, we also collected the outputs of three sub_XGBoost models, which were utilized in the cascade XGBoost algorithm, as follows:

### Cascade XGBoost algorithm

**Input:** original feature, PSSM, and outputs of four mutation analysis tools

**Step 1:** Utilizing the Gaussian WAPSSM algorithm, we can obtain the WAPSSM feature vector, named as \( f^1 \).

**Step 2:** Collect the original features in BorodaTM [33], labeled as \( f^2 \).

**Step 3:** Set different weights \( c_1, c_2, c_3, c_4 \) to outputs of PROVEAN, SIFT, fathmm, and PolyPhen-2. Thereafter, we can obtain a new feature vector \( f^3 \). That is,

\[
\begin{align*}
    f^3 &= [c_1 \times P_{\text{PROVEAN}}; c_2 \times P_{\text{SIFT}}; c_3 \times P_{\text{fathmm}}; c_4 \times P_{\text{PolyPhen-2}}].
\end{align*}
\]

**Step 4:** Feed \( f^3 \) into sub_XGBoost1. Label the outputs as \( O_1^{\text{XGBoost}} \).

**Step 5:** Feed \( f^2 \) into sub_XGBoost2. Label the outputs as \( O_2^{\text{XGBoost}} \).

**Step 6:** Feed \( f^3 \) into sub_XGBoost3. Label the outputs as \( O_3^{\text{XGBoost}} \).

**Step 7:** Concatenate the above feature vectors and mark as \( f^{\text{total}} \). That is,

\[
\begin{align*}
    f^{\text{total}} &= [f^1; f^2; f^3] = O_1^{\text{XGBoost}}; O_2^{\text{XGBoost}}; O_3^{\text{XGBoost}}.
\end{align*}
\]

**Step 8:** Use mRMR [54] to remove the redundant and irrelevant features from \( f^{\text{total}} \). Thereafter, we can obtain feature vector \( f^* \).

**Step 9:** Feed \( f^* \) into the XGBoost model for final prediction \([p_1, p_2]\).

**Output:** \([p_1, p_2]\)

Firstly, for each mutation, we denoted the WAPSSM feature as \( f^1 \), labeled the “Original” feature as \( f^2 \). In addition, we named the mutation analysis outputs as \( P_{\text{PROVEAN}}, P_{\text{SIFT}}, P_{\text{fathmm}}, \) and \( P_{\text{PolyPhen-2}} \) and set different weights \( c_1, c_2, c_3, c_4 \) to the above four outputs (i.e., \( c_1 \times P_{\text{PROVEAN}}; c_2 \times P_{\text{SIFT}}; c_3 \times P_{\text{fathmm}}; c_4 \times P_{\text{PolyPhen-2}} \)), which were concatenated and labelled as \( f^3 \).

Secondly, we fed \( f^1, f^2, f^3 \) into three sub_XGBoost models (namely sub_XGBoost1, sub_XGBoost2, and sub_XGBoost3), and documented the corresponding outputs as \( O_1^{\text{XGBoost}}, O_2^{\text{XGBoost}}, \) and \( O_3^{\text{XGBoost}} \) respectively.

Thirdly, we concatenated the features (i.e., \( f^1, f^2, f^3, O_1^{\text{XGBoost}}, O_2^{\text{XGBoost}}, \) and \( O_3^{\text{XGBoost}} \)) and labeled as \( f^{\text{total}} \). Notably, \( f^{\text{total}} \) may contain redundant and noisy features, which might lead to the decrease of the model performance. Thus, mRMR [54] was applied to rank and select more important features, denoted the selected feature vector as \( f^* \).

Finally, we fed \( f^* \) into the XGBoost model, and denoted the final prediction results as \([p_1, p_2]\), where \( p_1 \) and \( p_2 \) represent the probability of belonging to neutral and disease-associated class.

### 2.2.5 Feature selection using the minimum redundancy maximum relevance algorithm

In order to filter out redundant and identify features that contribute the most to model performance, in this work, we applied the minimum redundancy maximum relevance (mRMR) algorithm [54] to rank and select the most critical features from the extracted feature vector, introduced as follows:

\[
\begin{align*}
    \text{maxD} & = \frac{1}{|S|} \sum_{x_i \in S} I(x_i; C) \\
    \text{minR} & = \frac{1}{|S|} \sum_{x_i, x_j \in S} I(x_i; x_j) \\
    \text{max} & = F(D, R) = F = D - R
\end{align*}
\]

where \( x_i \) and \( x_j \) are two specific feature vectors, \( I(x_i; x_j) \) is the mutual information function, \( S \) is the entire feature set, \( C \) is the mutation class (i.e. disease-associated and neutral mutation), and \( D \) is the correlation of \( x_i \) and \( x_j \). The function \( \text{maxD}(S, C) \) in Eq. (1) means that feature \( x_i \) has the most significant effect on class \( C \). Besides, \( R \) represents the correlation between \( x_i \) and \( x_j \). Function \( \text{minR}(S) \) in Eq. (2) means that the feature vectors \( x_i \) and \( x_j \) have the minimum redundancy. As shown in the formula (3), mRMR can find out a feature subspace, which has the minimum redundancy between features and the maximum relevancy with the mutation class [54].

### 2.3. Performance evaluation

In this work, we utilized two evaluation methods (i.e. randomized 10-fold cross-validation and leave-one-out cross-validation) to assess the XGBoost model and compare MutTMPredictor with other machine learning methods and existing predictors.

1. **Randomized 10-fold cross-validation**

   The specific process follows. The mutation dataset was divided into ten parts, and the test procedure was implemented in ten randomization cycles. For each cycle, nine parts of the dataset were used as the training set to train the model, while the remaining part was used to test the performance of the trained model. We then calculated the average values of ten cycles as the model performance.

2. **Leave-one-out cross-validation**

   Among seven datasets, the 546 mutations dataset is relatively small. Thus, the performance of the predictor may be biased on such dataset. Moreover, it is also somewhat unreasonable to use only 10% of dataset (only 55 mutations) and compare with the existing predictors. Therefore, we performed the “leave-one-out” test on this dataset.

### 2.4. Evaluation metrics

Based on the confusion matrix, several performance metrics can be derived, such as Recall (Sen), specificity (Spe), precision (Pre), accuracy (ACC), Matthew’s Correlation Coefficient (MCC), F1-score (F1), and negative predictive value (NPV), error rate (ER), false negative rate (FNR), and false positive rate (FPR). In this work, the above performance metrics were calculated to evaluate the developed predictor and other existing predictors [31,34,35].

\[
\begin{align*}
    \text{Pre} &= \frac{TP}{TP + FP} \\
    \text{Spe} &= \frac{TN}{TN + FN} \\
    \text{NPV} &= \frac{TN}{TN + FN} \\
    \text{Recall} & = \frac{TP}{TP + FN} \\
    F_1 &= \frac{2 \times TP}{2 \times TP + FP + FN}
\end{align*}
\]
Error rate = \frac{FP}{(TP + TN + FP + FN)} \tag{9}

ACC = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{10}

False positive rate = \frac{FP}{(TN + FP)} \tag{11}

False negative rate = \frac{FN}{(TP + FN)} \tag{12}

\[
MCC = \frac{(TP \times TN - FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{13}
\]

where \(TP\) (true positive)/\(TN\) (true negative) are the number of disease-associated mutations/neutral mutations that are correctly predicted as disease-associated/neutral mutations; \(FP\) (false positive) is the number of neutral mutations that are incorrectly predicted as disease-associated mutations; \(FN\) (false negative) is the number of disease-associated mutations that are incorrectly predicted as neutral mutations, respectively.

Among all performance metrics, MCC is considered to be the best indicator for evaluating the model performance, especially on unbalanced datasets [55]. The value of MCC ranges from \(-1\) to 1, while those of \(Pre\), \(Spe\), \(NPV\), \(Recall\), \(Sen\), \(F_1\), and \(ACC\) range from 0 to 1. Generally, the larger the metric’s value, the better the model’s predictive performance [56]. Herein, we also utilized another metric, i.e. the area under the curve (AUC), which is defined as the area under the receiver-operating characteristic (ROC) curve [57] and the coordinate axis. The closer the AUC value is to 1.0, the more accurate the prediction model.

Apart from traditional performance metrics, we also utilized three types of errors: error rate, false positive rate, and false negative rate, which reflect the predictive performance of the trained models and range also from 0 to 1. Generally, the lower the errors values, the better the predictor.

3. Results and discussions

3.1. Illustration of the developed MutTMPredictor

Fig. 1 illustrates the workflow of MutTMPredictor. Fig. 1(A) shows feature extraction steps: (1) Extract WAPSSM built on original PSSM matrix; (2) Collect the “Original” features from BordaTM [33]; (3) Generate “Individuals’ output” of SIFT [6], PROVEAN [5], PolyPhen-2 [11,12], and fathmm [53], (4) Feed “WAPSSM + two other types” features into three sub_XGBoost models and name the outputs as “Output1”, “Output2”, and “Output3”, respectively. As depicted in Fig. 1(B), we concatenated the extracted features obtained from Fig. 1(A) and applied mRMR [54] to perform feature selection. Thereafter, we obtained the feature vector subspace and then fed into the XGBoost algorithm for making the final prediction.

3.2. Comparison of PSSM with WAPSSM

In Section 2.2.2, we proposed the Gaussian WAPSSM algorithm, which requires two variables: the PSSM matrix and half of the microenvironment size. We empirically set half of the microenvironment size to 3 (i.e. \(2k + 1 = 7\)). Herein, we conducted some comparison experiments to examine whether the new encoding WAPSSM is effective and superior to the original PSSM. The XGBoost model was implemented using scikit-learn [58], which was trained on the training data (90%) and then tested on the test data (10%). The performance comparison results are documented in Table 2.

As shown in Table 2, WAPSSM features could achieve more \(TP\) and \(TN\), less \(FN\) and \(FP\) than PSSM. Besides, the \(MCC\) and \(ACC\) values of WAPSSM features were 0.2657 and 0.6364, which were 0.1688 and 0.0728 higher than those of PSSM. In terms of the \(Pre\), \(Recall\), and \(F_1\) values, WAPSSM were also higher than those of PSSM. Taking the results in Table 2 into consideration, we concluded that WAPSSM was more effective than PSSM for mutation prediction transmembrane proteins.

3.3. Effectiveness of different types of features and their combinations

In Section 2.2, three types of features (including “WAPSSM”, “Original”, and “Individuals’ output”) were obtained. Herein, we further concatenated them in a consecutive manner (i.e. WAPSSM + Original + Individuals’ output) and labeled it as “Combined”. In this section, we mainly examined the features effectiveness and demonstrate whether the “Combined” features can further improve the prediction performance. In these experiments, the XGBoost model was trained on the training dataset and tested on the test dataset again. The performance comparison results are shown in Table 3.

In terms of \(TP\), \(TN\), \(FP\), and \(FN\) values listed in Table 3, we can see that the “Combined” features performed best. In terms of the \(Pre\), \(Sen\), and \(F_1\) values, “Combined” was also superior to “WAPSSM”, “Original”, and “Individuals’ output” features. In addition, the \(ACC\) value of “Combined” features was 0.8364, which was 0.2000, 0.1273, and 0.0909, respectively, higher than that of the “WAPSSM”, “Original”, and “Individuals’ output”. The \(MCC\) values of “Original”, “WAPSSM”, and “Individuals’ output” were 0.4249, 0.2657, and 0.5068. By combining the three types of features together (i.e. WAPSSM + Original + Individuals’ output), the \(MCC\) value could be further increased to 0.6763, which was 0.4106, 0.2514, and 0.1695, respectively higher than that of “WAPSSM”, “Original”, and “Individuals’ output”. Taken together, we concluded that the “Combined” feature is an overall best choice for representing mutations in transmembrane proteins.

In this part, some comparison experiments were performed to assess the effectiveness of each “Individuals’ output” feature and their combinations. Detailed information can be found in Supplementary Text S1 and Table S2. Besides, we also designed experiments to test whether the prediction performance changed after removing the WAPSSM feature. For more details, please refer to Supplementary Table S3 and Text S2.

3.4. Cascade XGBoost improved transmembrane protein mutation prediction

There exist some redundant features that might decrease the performance of prediction model. As such, feature selection methods, including Chi-Square (CHI2) [59], Information Gain (IG) [60], and Mutual Information (MI) [61], and mRMR [54], have been applied to select features. According to the results of CHI2, IG, MI, and mRMR, we finally selected mRMR [54] to perform feature selection and reserved some features with the property of minimum redundancy maximum relevance for the cascade XGBoost model.

In this section, we compared the performance of our new proposed cascade XGBoost with that of the XGBoost model. Again, two models were trained on the training data and tested on the test data. The comparison results are listed in Table 4. From Table 4, we can see that the cascade XGBoost model predicted seven more \(TP\) and one fewer \(FN\) than the XGBoost model. On the other hand, MCC and ACC values of cascade XGBoost were 0.7166 and 0.8727, which were 0.0403 and 0.0363 respectively higher than those of XGBoost. On concerning to the \(Pre\), \(Recall\), and \(F_1\) values, cascade XGBoost was also superior to XGBoost. Altogether, we concluded that the new proposed cascade XGBoost model is a better choice for mutation prediction.
Table 3
Performance comparison of “Original”, “WAPSSM”, “Individuals’ output”, and “Combined” features on the test data of the 546 mutations dataset.

| Features          | TP          | TN          | FP          | FN          | Pre         | Sen         | F1          | ACC         | MCC         |
|-------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| WAPSSM            | 26 (47.3%)  | 9 (16.4%)   | 16 (29.1%)  | 4 (7.27%)   | 0.6190      | 0.8667      | 0.7222      | 0.6364      | 0.2657      |
| Original          | 27 (49.1%)  | 12 (21.8%)  | 13 (23.6%)  | 3 (5.45%)   | 0.6750      | 0.9000      | 0.7714      | 0.7455      | 0.4249      |
| Individuals’ output| 28 (50.9%)  | 13 (23.6%)  | 12 (21.8%)  | 2 (3.64%)   | 0.7000      | 0.9333      | 0.8000      | 0.7455      | 0.5068      |
| Combined          | 28 (50.9%)  | 18 (32.7%)  | 7 (12.7%)   | 2 (3.64%)   | 0.8000      | 0.9333      | 0.8615      | 0.8364      | 0.6763      |

Fig. 1. An overall workflow of MutMPredictor.
3.5. Performance comparison between cascade XGBoost and seven machine learning methods

We compared cascade XGBoost with seven traditional machine learning methods to further illustrate its effectiveness. These seven methods were divided into two groups: (1) single methods, including Support vector machine (SVM) [63,64], K-nearest neighbors (KNN) [65], Decision tree (DT) [66], and (2) ensemble methods, including Random forest (RF) [67,68], Extremely randomized trees (ERT) [69], AdaBoost [70], and Gradient boosted decision trees (GBDT) [71].

As mentioned in Section 2.1, the 546 mutations dataset used in this section was relatively small, with only 392 disease-associated and 154 neutral mutations. Thus, the model performance may be biased if we only utilized the test data (only 55 mutations) to test each model. Accordingly, we designed the following experiments by performing 10-fold cross-validation on the entire dataset. Comparison results are discussed in Supplementary Text S4, Tables S7 and S8. By summarizing the results analyses, we conclude that the cascade XGBoost model performed best.

Building upon the new feature encoding algorithm and other extracted features, along with the cascade XGBoost model, we implemented a new transmembrane protein mutation predictor, named MutTMPredictor. In the following sections, the experiments are conducted to assess the efficiency of MutTMPredictor in mutation effect prediction.

3.6. Performance comparison of MutTMPredictor with six existing predictors on 442 mutations

In this section, we performed the leave-one-out cross-validation test to benchmark MutTMPredictor against several existing state-of-the-art predictors. Notably, as for MutTMPredictor, we utilized “individuals’ output” as part of feature vector. To prevent model over-fitting, before implementing comparison experiments, we first constructed a new dataset based on the 546 mutations and removed protein sequences used in individual experiments, we first constructed a new dataset based on the existing state-of-the-art predictors. Notably, as for MutTMPredictor, all utilized protein structural characteristics, we only retained proteins whose “released-date” was after the individual predictors’ published date. Accordingly, after protein sequence removing procedure, we could construct a new dataset based on 546 mutations. Specifically, we searched each protein from PDB [41] and extracted its "deposition_date" and "release_date".

The publication year of four individual predictors is described herein: SIFT(2002) [6], PROVEAN (2012) [5], PolyPhen-2(2010) [11,12], and fathmm (2013) [53]. According to these dates, we decided to take “Year: 2013” as the cut-off threshold. That is, we only kept those proteins whose “released-date” is after 2013. For detailed information about the “deposition_date” and “release_date” of proteins and whether we should “keep” or “delete” a specific protein, please refer to Supplementary Table S9.

After the above steps, we deleted 27 proteins containing 104 mutations and eventually kept 37 proteins with 442 mutations. After that, we constructed a test dataset, which comprised 350 disease-associated and 92 neutral mutations. Next, we conducted the following experiments on the remaining 442 mutations to compare MutTMPredictor with six existing predictors. Specifically, for fathmm [53], PROVEAN [5], SIFT [6], and PolyPhen-2 [11,12], we fed all 442 mutations into their webservers. Then we calculated the performance metrics based on the returned predictions and provided the results in Table 5 and Fig. 2(A)–(B). In Table 5, we collected the prediction results of BorodaTM [33] and Entprise [72] from BorodaTM [33]. Besides, we further calculated TP, TN, FP, and FN values for BorodaTM and Entprise based on the given ACC, Pre, Recall, F1, and MCC values.

Based on the comparison results in Table 5 and Fig. 2(A)–(B), we draw the following conclusions: (1) For PROVEAN, SIFT, fathmm, and PolyPhen-2, MCC values ranged from 0.1686 to 0.4936, and ACC ranged from 0.5290 to 0.8416. The average values of MCC and ACC were 0.3948 and 0.7806, which were much lower than those of Entprise (i.e. 0.6940 and 0.8553) and BorodaTM (0.8563 and 0.9358).

(2) Entprise is superior to the above four predictors. For instance, the MCC value of Entprise was 0.6940, which was
0.2236, 0.2004, 0.2476, and 0.5254 higher than those of PolyPhen-
2, SIFT, PROVEAN, and fathmm.

(3) BorodaTM is the first predictor that could predict mutations
in transmembrane protein. As can be seen from Table 5 and Fig. 2,
five performance metrics of BorodaTM were much higher than
those of Entprise, PROVEAN, SIFT, fathmm, and PolyPhen-2. Specif-
ically, Pre, Recall, F1, ACC, and MCC values of BorodaTM were
0.9917, 0.9184, 0.9536, 0.9358, and 0.8563, which were 0.1250,
0.0178, 0.0703, 0.0805, and 0.1623 respectively higher than those
of Entprise.

(4) From Table 5 and Fig. 2, it can be easily found that
MutTMPredictor performed best among seven predictors. Specifi-
cally, MutTMPredictor predicted more TP and TN, and fewer FP
and FN than other predictors. Besides, it also achieved an MCC
value of 0.8950, which was 0.0387 and 0.2010 higher than that
of BorodaTM and Entprise.

3.7. Interpretation of incorrectly predicted mutations

In this section, we comprehensively evaluated the results of fathmm [53], PROVEAN [5], SIFT [6], and PolyPhen-2 [11.12] and
found that 18 out of 442 mutations were predicted incorrectly by
all four predictors concurrently. These 18 mutations included
P11166 (R223P), P28472 (G32R), O15118 (I1220T), O15118
(V757A), P02730 (R832H), P02730 (E508K), P28472 (Q173L),
P29033 (A148P), P29033 (R32L), P29033 (G45E), P29033 (I203T),
P29033 (F191L), P30542 (R105H), Q13255 (E741D), Q9H221
(G575R), Q9Y6J6 (A66V), Q9Y6J6 (T8A), and Q9Y6J6 (T8I). It is of
particular interest to note that, among these 18 mutations, there
were two disease-associated mutations, i.e. P11166 (R223P) and
P28472 (G32R), and 16 neutral mutations.

Herein, we elaborated on the above two disease-associated
mutations that four existing predictors incorrectly predicted. From
the perspective of physicochemical properties, in two mutations,
i.e. P28472 (G32R), the wild-type residue G is hydrophilic, whereas
mutant residue R is alkaline. In P11166 (R223P), the wild-type resi-
due R is alkaline, whereas mutant residue P is hydrophobicity. As
known, mutations with different physicochemical property
changes could result in the abnormal expression of protein biological
function. As reported, for P11166 (R223P), the R223 residue is
involved in the hydrogen bond interactions that enable the trans-
porter inward open configuration [73]. As such, the mutation
occurring at this site may lead to the transporter property changes
[74]. Besides, Ref. [75] reported that G32R in P28472 could result
in hyperglycosylated and reduce GABA currents in GABRB3.

Among the above 18 mutations, MutTMPredictor incorrectly
predicted only two mutations, i.e. P29033 (I203T), PDB ID: 5ER7
and P30542 (R105H), PDB ID: 5UEN, both of which belonged to
neutral mutations. We utilized PYMOL software [76] to show the
3D structures of two proteins with “sticks + spheres” representa-
tion, as depicted in Fig. 3.

From Fig. 3, we can see that both mutations in 5ER7 and 5UEN
occur within the inner region of proteins. P29033 (5ER7) is
assigned as a known gap junction beta-2 protein, which is located
in the plasma membrane [77]. Meanwhile P30542 (5UEN) is a
member of the heterotrimeric guanine nucleotide-binding
protein–coupled receptor family A, which is reported to be associ-
ated with several neurological diseases, such as Parkinson and Alz-
heimer [78]. Thus, P30542 is being pursued as a therapeutic target
to treat the above human diseases [60].

As reported, even one missense mutation occurs on the critical
site of α-helix, it may be deleterious to protein folding and/or its
biological function [25]. As such, the α-helices in membrane pro-
teins are often “hot spots” of disease-associated missense muta-
tions [79]. From Fig. 3, we can see that mutations I203T in PDBid
5ER7, and R105H in PDBid 5UEN are both located within the α-
helices region. Furthermore, for I203T, both I and T are hydrophilic
amino acid residues. But the mutant residue T is less hydrophobic
than the wild-type residue I, resulting in the loss of hydrophobic
interactions [80]. For R105H, both R and H are alkaline amino acid
residues. Similarly, the residue H is less alkaline than the wild-type
R. However, the above two missense mutations are labeled as neu-
tral, which is often ignored in functional analysis of a specific gene.
For example, P29033 (I203T) has not been included in GJB2 deaf-
ness gene analysis [81]. Notably, the above two mutations (I203T
and R105H) in the corresponding proteins were incorrectly pre-
Predicted by five predictors at the same time. Among the 18 incorrectly predicted mutations, the number of neutral mutations predicted incorrectly as positive (i.e. FP) was larger than FN. This implies that all these predictors set a more stringent threshold for predicting disease-associated mutations.

3.8. Performance comparison of MutTMPredictor with the consensus predictor PredictSNP and its component predictors on 546 mutations

In this section, we compared MutTMPredictor with the consensus predictor PredictSNP [13] and its component predictors, including MAPP [82], PhD-SNP [83], PolyPhen1 [10], and SNAP [9]. Specifically, we fed 546 mutations into the PredictSNP webserver, downloaded the outputs, and calculated the prediction performance. The results are documented in Tables 6–7.

In Tables 6–7, MAPP [82], PhD-SNP [83], PolyPhen1 [10], and SNAP [9] are single predictors, which applied the score threshold, support vector machine, prediction rules, and neural network to predict mutation effect. In contrast, PredictSNP is a consensus predictor which integrates the outputs of several single predictors. Detailed results are discussed below.

(1) Four single predictors and consensus predictor. As for four single predictors, the ACC values ranged from 0.7161 to 0.7967 with the average of 0.7445. MCC values were in the range from 0.3743 to 0.4932 with the average of 0.4230. AUC values were in the range from 0.7067 to 0.7670 with average of 0.7388. The consensus predictor PredictSNP increased the ACC, MCC, and AUC values to 0.7985, 0.5190, and 0.7670, which were 0.0018, 0.0258, and 0.0229, respectively higher than the above best predictor PhDSNP. Apparently, PredictSNP outperformed the four single predictors.

(2) Consensus predictor and MutTMPredictor. The ACC, MCC, and AUC values of MutTMPredictor were 0.9634, 0.9090, and 0.9508, which were 0.1649, 0.3900, and 0.1838 higher than PredictSNP. In terms of the TP, TN, FP, and FN values, MutTMPredictor could predict 55 more TP, 35 more TN, 35 fewer FP, and 55 fewer FN than PredictSNP (Table 7). Such advantage is also reflected by the ER, FPR, and FNR values.

| Predictor      | ACC  | precision | recall | F1-score | MCC  | SN   | AUC  | SP   | NPV |
|---------------|-----|-----------|--------|----------|------|------|------|------|-----|
| PredictSNP    | 0.7985 | 0.8750    | 0.8393 | 0.8568   | 0.5190 | 0.8393 | 0.7670 | 0.6948 | 0.6294 |
| MAPP          | 0.7161 | 0.8496    | 0.7347 | 0.7880   | 0.3743 | 0.7347 | 0.7670 | 0.6688 | 0.4976 |
| PhD-SNP       | 0.7967 | 0.8539    | 0.8648 | 0.8593   | 0.4932 | 0.8648 | 0.7441 | 0.6234 | 0.6443 |
| PolyPhen1     | 0.7289 | 0.8486    | 0.7577 | 0.8005   | 0.3879 | 0.7577 | 0.7067 | 0.6558 | 0.5153 |
| SNAP          | 0.7363 | 0.8780    | 0.7347 | 0.8000   | 0.4364 | 0.7347 | 0.7375 | 0.7403 | 0.5229 |
| MutTMPredictor| 0.9634 | 0.9697    | 0.9796 | 0.9746   | 0.9090 | 0.9796 | 0.9508 | 0.9221 | 0.9467 |

Note: the PredictSNP webserver can provide prediction results for nine predictors, including MAPP [82], nsSNPAnalyzer [84], PANTHER [85], PhD-SNP [83], PolyPhen1 [10], PolyPhen-2 [11], SIFT [6], SNAP [9], and PredictSNP [13]. In Tables 6–7, the results of nsSNPAnalyzer, PANTHER, SIFT, and PolyPhen-2 were not listed, because: (1) there were too many “unknown” in nsSNPAnalyzer and PANTHER outputs; (2) performance comparison with SIFT, and PolyPhen-2 is discussed in the previous section.

| Predictor      | TP   | TN   | FP   | FN   | ER  | FPR  | FNR  |
|---------------|------|------|------|------|-----|------|------|
| PredictSNP    | 329(60.26%) | 107(19.60%) | 47(8.61%) | 63(11.54%) | 0.0861 | 0.3052 | 0.1607 |
| MAPP          | 288(52.75%) | 103(18.86%) | 51(9.34%) | 104(19.05%) | 0.0934 | 0.3312 | 0.2653 |
| PhD-SNP       | 339(62.09%) | 96(17.58%) | 58(10.62%) | 53(9.71%) | 0.1062 | 0.3766 | 0.1352 |
| PolyPhen1     | 297(54.40%) | 101(18.50%) | 53(9.71%) | 95(17.40%) | 0.0971 | 0.3442 | 0.2423 |
| SNAP          | 288(52.75%) | 114(20.88%) | 40(7.33%) | 104(19.05%) | 0.0733 | 0.2597 | 0.2653 |
| MutTMPredictor| 384(70.33%) | 142(26.01%) | 12(2.20%) | 8(1.47%)  | 0.0220 | 0.0779 | 0.0204 |
For example, MutTMPredictor (with ER of 0.0220, FPR of 0.0779, and FNR of 0.0204) had lower errors than PredictSNP (with ER of 0.0861, FPR of 0.3052, and FNR of 0.1607).

There are three main possible reasons for aforementioned phenomena: First, single and consensus predictors may exhibit excellent prediction performance on their own datasets. However, the performance may be lower when switched to 546 mutations datasets; Second, PredictSNP is a consensus predictor by taking six best outputs from eight single predictors and accordingly it generally outperforms its component predictors [13], and Third, MutTMPredictor takes the outputs of fathmm [53], PROVEAN [5], SIFT [6], and PolyPhen-2 [11,12] and can be seen as a consensus predictor in some sense. Besides, we utilize the cascade XGBoost algorithm to reuse the useful features. As such, MutTMPredictor outperforms single and consensus predictors and is more robust for large-scale mutation prediction.

3.9. Performance comparison of MutTMPredictor with Pred-MutHTP and mCSM-membrane on 546 mutations

Pred-MutHTTP [31] and mCSM-membrane [34] are two predictors specifically developed for the pathogenicity prediction of mutations in transmembrane proteins. In this section, we conducted comparison experiments to further examine the effectiveness of MutTMPredictor on 546 mutations dataset. In particular, we fed one of 546 mutations once into the webserver of Pred-MutHTTP [31] and mCSM-membrane [34] and then calculated their evaluation metrics based on the returned prediction results. It is noteworthy that, when feeding 546 mutations into the webserver of mCSM-membrane, 101 out of 546 mutations were returned with the "error" mark, such as "Error: Provided PDB file has multiple predictions". Accordingly, we evaluated the results of mCSM-membrane in two ways: (i) assessing total 546 mutations. Herein, we treated the aforementioned 101 mutations with "error" mark as "prediction errors",

(ii) deleting the aforementioned 101 mutations. That is, we calculated the performance metrics values only based on the prediction results of 445 mutations. After that, we list the performance comparison results of (i) and (ii) in "mCSM-membrane (546 mutations)" and "mCSM-membrane (445 mutations)" of Tables 8–9 and Fig. 4.

According to the performance results in Tables 8–9 and Fig. 4, we have the following observations:

Table 8

| Predictor                        | TP    | TN    | FP    | FN    | ACC  | Pre   | Recall | F1    | Spe  | NPV  |
|---------------------------------|-------|-------|-------|-------|------|-------|--------|-------|------|------|
| Pred-MutHTTP (546 mutations)    | 362(66.30%) | 103(18.86%) | 50(9.16%) | 31(5.68%) | 0.8516 | 0.8786 | 0.9211 | 0.8994 | 0.6732 | 0.7687 |
| mCSM-membrane (546 mutations)   | 322(59.08%) | 110(20.18%) | 42(7.71%) | 71(13.03%) | 0.7927 | 0.8846 | 0.8193 | 0.8507 | 0.7237 | 0.6077 |
| mCSM-membrane (445 mutations)   | 321(72.13%) | 111(24.94%) | 42(7.71%) | 71(13.03%) | 0.7927 | 0.8846 | 0.8193 | 0.8507 | 0.7237 | 0.6077 |
| MutTMPredictor (546 mutations)  | 384(70.33%) | 142(26.01%) | 12(2.20%) | 8 (1.47%) | 0.9634 | 0.9697 | 0.9796 | 0.9746 | 0.9221 | 0.9467 |

Pred-MutHTTP: https://www.iitr.ac.in/bioinfo/PredMutHTTP/; mCSM-membrane: http://biosig.unimelb.edu.au/mcsm_membrane/.

Table 9

| Predictor                        | Error rate | False positive rate | False negative rate |
|---------------------------------|------------|---------------------|---------------------|
| Pred-MutHTTP (546 mutations)    | 0.0916     | 0.3268              | 0.0789              |
| mCSM-membrane (546 mutations)   | 0.0771     | 0.2763              | 0.1807              |
| mCSM-membrane (445 mutations)   | 0.0270     | 0.0976              | 0.0031              |
| MutTMPredictor (546 mutations)  | 0.0220     | 0.0779              | 0.0204              |

The underlying reasons for the above results are discussed as follows: (1) features utilized by three methods are quite different. Specifically, Pred-MutHTTP mainly used protein sequence-based features, such as substitution matrices values, residue distributions in certain regions, as well as physicochemical properties and evolutionary information [31]. mCSM-membrane mainly utilized graph-based signatures, protein geometry, and physical and chemical properties [34]. In contrast, MutTMPredictor applied various features extracted from characteristics of protein sequence, structure and outputs of four existing predictors. Therefore, features utilized in MutTMPredictor are more comprehensive. (2) For the total
546 mutations, mCSM-membrane [34] only predicted 445 mutations. That is, mCSM-membrane could only predict about 81.50% mutations, whereas Pred-MutHTP [31] and MutTMPredictor predicted all 546 mutations.

3.10. Performance evaluation of MutTMPredictor on 67,584 mutations dataset

We did not use the structure- and energy-based features for the prediction task of 67,584 mutations due to certain reasons. In order to check the prediction performance of models trained using different numbers of input features ranked by mRMR [54], we displayed the MCC and ACC value changes of MutTMPredictor in Supplementary Fig. S2. Detailed analyses are documented in Supplementary Text S5.

On this large test dataset, we compared the performance of MutTMPredictor with fathmm [53], PROVEAN [5], SIFT [6], and PolyPhen-2 [11,12]. For MutTMPredictor, we input top 20 features selected by mRMR [54] and applied 10-fold cross-validation to evaluate it. For each cycle in 10-fold cross-validation, we documented the TP, TN, FP, and FN values in Supplementary Table S10 and then we further calculated the sums of TP, TN, FP, and FN and recorded them in Table 11. As four predictors, again, we submitted 67,584 mutations to their respective webserver and calculated the performance metrics based on the prediction results. Comparison results are provided in Tables 10–11 and depicted in Fig. 5(A)–(B).

Table 10

| Predictor     | ACC   | Pre   | Recall | F1    | Spe   | NPV   |
|---------------|-------|-------|--------|-------|-------|-------|
| SIFT*         | 0.7298| 0.6422| 0.8370 | 0.7268| 0.6491| 0.8411|
| PolyPhen-2*   | 0.7362| 0.6357| 0.8939 | 0.7430| 0.6189| 0.8869|
| PROVEAN*      | 0.7680| 0.6963| 0.8155 | 0.7512| 0.7323| 0.8406|
| fathmm*       | 0.7993| 0.7694| 0.7605 | 0.7649| 0.8284| 0.8213|
| MutTMPredictor| 0.8776| 0.8641| 0.8526 | 0.8567| 0.8965| 0.8914|

Note: PROVEAN*; PolyPhen-2*; fathmm*.

Table 11

| Predictor     | TP    | TN    | FP    | FN    | ER    | FPR   | FNR   |
|---------------|-------|-------|-------|-------|-------|-------|-------|
| SIFT*         | 24292| 25031| 13533| 20.02%| 0.2002| 0.3509| 0.1630|
| PolyPhen-2*   | 25638| 23864| 6616  | 9.79%  | 0.1527| 0.2677| 0.1845|
| PROVEAN*      | 23665| 28240| 10324| 15.28% | 0.1527| 0.2677| 0.1845|
| fathmm*       | 22069| 31948| 6915  | 10.28% | 0.0979| 0.1716| 0.2386|
| MutTMPredictor| 24743| 34572| 3992  | 5.91%  | 0.0591| 0.1035| 0.1474|

Fig. 4. Performance comparison of Pred-MutHTP, mCSM-membrane, and MutTMPredictor in terms of MCC and AUC on 546 mutations dataset.
AUC values of SIFT, PolyPhen-2, PROVEAN, and fathmm were 0.7439, 0.7616, 0.7739, and 0.7945. In contrast, MutTMPredictor achieved an AUC of 0.8746, which was 0.1307, 0.1130, 0.1007, and 0.0801 respectively, higher than that of SIFT, PolyPhen-2, PROVEAN, and fathmm.

3.11. Performance evaluation of MutTMPredictor on mutations located in three different topological regions of membrane proteins

3.11.1. Performance comparison between MutTMPredictor and Pred-MutHTP

Four datasets were collected from the Pred-MutHTP [31] website, including the whole dataset and three datasets containing mutations in different topological regions of membrane proteins, i.e. “Cytoplasmic or Inside”, “Membrane”, and “Extracellular or Outside”. We conducted several experiments to compare MutTMPredictor with Pred-MutHTP on the four datasets. Detailed comparison results are documented in Tables 12–13.

From Table 12, we can see that MutTMPredictor predicted more TP/TN and less FN/FP than Pred-MutHTP on the four datasets. For example, for the “Cytoplasmic or Inside” mutations, MutTMPredictor predicted 1,190 more TP, 306 less FP, and 460 less FN than Pred-MutHTP over “10-fold”. On the other hand, MutTMPredictor achieved ER of 0.0909, FPR of 0.2264, and FNR of 0.1270, which were 0.0414, 0.0382, and 0.1498 respectively lower than Pred-MutHTP. Such advantages can also be seen in terms of SN, SP, ACC, MCC, and AUC values listed in Table 13. For example, for the “Extracellular or Outside” mutations using the “test” evaluation, Pred-MutHTP achieved the average of 0.7118. In contrast, the specific predictors increased the AUC to the range (0.7900, 0.8277) with the average of 0.8137.

3.11.2. Performance comparison of MutTMPredictor, four non-specific, and two specific predictors

In this section, we compared MutTMPredictor with four predictors non-specific for membrane proteins (including fathmm [53], PROVEAN [5], SIFT [6], and PolyPhen-2 [11,12]) and two predictors specific for membrane proteins (i.e. Pred-MutHTP [31] and TMSNP [35]). The performance results are documented in Tables 14–15. From Tables 14–15, we can see that specific predictors were generally superior to non-specific predictors. For example, for “Cytoplasmic or Inside region” mutations, the AUC values of four non-specific predictors were in range of (0.6844, 0.7524) with the average of 0.7118. In contrast, the specific predictors increased the AUC to the range (0.7900, 0.8277) with the average of 0.8137.

MutTMPredictor performed best among all the three specific predictors. For instance, on the “Membrane” mutations, the ACC, Recall, F1, Spe, NPV, MCC, and AUC values of MutTMPredictor were 0.9321, 0.9544, 0.9485, 0.8898, 0.9113, 0.8490, and 0.9141, which were 0.0402, 0.0418, 0.0125, 0.1317, 0.3381, 0.2507, and 0.0788, respectively higher than those of TMSNP, and 0.1388, 0.1406, 0.1043, 0.1417, 0.2656, 0.3090, and 0.0741, respectively higher than those of Pred-MutHTP.

The underlying reasons for the above phenomena are discussed below. First, in terms of different types of features, Pred-MutHTP mainly used evolutionary information, physiochemical properties, neighboring residue information, and specific membrane protein attributes [31]. In contrast, MutTMPredictor used individual’s outputs except for the above features, which might make MutTMPredictor more robust. Second, in terms of feature selection and classification methods, Pred-MutHTP utilized two feature selection methods, including CfsSubsetEval and Consistency evaluator in WEKA [86]. Then Pred-MutHTP adopted all available methods in WEKA and selected the voting algorithm to classify mutations [31]. In contrast, MutTMPredictor applied the mRMR [54] feature selection method to score each feature and then fed the top features into the cascade XGBoost model for making the final prediction.

Fig. 5. Performance assessment of five predictors in terms of the MCC and AUC values on 67,584 mutations dataset.
other predictors on all the three datasets. Summary, MutTMPredictor achieved a better performance than with a richer set of features, including evolutionary information, MutTMPredictor utilized the cascade XGBoost algorithm combined in membrane proteins.

Table 12

| Dataset          | Predictor        | Num of fea | Validation | TP       | TN       | FP       | FN       | ER       | FPR     | FNR     | MCC     | AUC     |
|------------------|------------------|------------|------------|----------|----------|----------|----------|----------|---------|---------|---------|---------|
| Whole data       | MutTMPredictor   | 61         | 10-fold    | 10501(49.72%) | 7803(36.98%) | 1523(7.20%) | 1284(6.11%) | 0.0720  | 0.1629  | 0.1094  |         |         |
|                  | Pred-MutHTP      | 20         | test       | 2175(51.50%) | 1569(37.15%) | 288(6.82%) | 191(4.52%) | 0.0682  | 0.1551  | 0.0807  |         |         |
|                  | Pred-MutHTP#v    | 20         | test       | 1380(21.28%) | 1972(46.13%) | 537(12.65%) | 386(9.03%) | 0.1256  | 0.2144  | 0.1816  |         |         |
| Cytoplasmic or Inside | MutTMPredictor | 20         | 10-fold    | 3856(52.24%) | 2289(31.07%) | 669(9.09%) | 560(7.60%) | 0.0909  | 0.2264  | 0.1270  |         |         |
|                  | Pred-MutHTP#v    | 15         | 10-fold    | 2666(36.16%) | 2711(36.77%) | 975(13.23%) | 1020(13.84%) | 0.1323  | 0.2646  | 0.2768  |         |         |
| Membrane         | MutTMPredictor   | 60         | 10-fold    | 2304(62.50%) | 1137(30.71%) | 148(3.80%) | 117(2.99%) | 0.0380  | 0.1102  | 0.0456  |         |         |
|                  | Pred-MutHTP#v    | 15         | 10-fold    | 457(61.59%)  | 237(31.94%)  | 36(4.85%)  | 12(1.62%)  | 0.0485  | 0.1319  | 0.0256  |         |         |
| Extracellular or Outside | MutTMPredictor | 25         | 10-fold    | 4323(43.23%) | 443(44.12%)  | 652(6.47%) | 616(6.18%) | 0.0647  | 0.1280  | 0.1250  |         |         |
|                  | Pred-MutHTP#v    | 19         | 10-fold    | 802(44.94%)  | 822(43.95%)  | 114(5.68%) | 109(5.43%) | 0.0968  | 0.1145  | 0.1078  |         |         |

Note: TP, TN, FP, and FN values of Pred-MutHTP# were calculated based on the given SN, SP, ACC, MCC, AUC values of Pred-MutHTP. “Num of fea” is the number of features used in the model prediction. Validation*: in Pred-MutHTP [31], the authors used CD-HIT [44] to aggregate sequences into ten clusters and performed 10-fold-group-wise cross-validation on the datasets. However, the authors did not provide the specific sequences in ten clusters. Herein, we applied 10-fold cross-validation to the corresponding datasets. “test” means 20% independent test.

Table 13

| Dataset          | Predictor        | Num of fea | Validation | TP       | TN       | FP       | FN       | ER       | FPR     | FNR     | MCC     | AUC     |
|------------------|------------------|------------|------------|----------|----------|----------|----------|----------|---------|---------|---------|---------|
| Whole data       | MutTMPredictor   | 61         | 10-fold    | 0.8906   | 0.8371   | 0.867    | 0.7297   | 0.8048   |         |         |         |         |
|                  | Pred-MutHTP#v    | 20         | test       | 0.9193   | 0.8449   | 0.8866  | 0.7693   | 0.8221   |         |         |         |         |
|                  | Pred-MutHTP#v    | 15         | test       | 0.8732   | 0.7738   | 0.8333  | 0.6527   | 0.8235   |         |         |         |         |
|                  | Pred-MutHTP#v    | 15         | test       | 0.7232   | 0.7354   | 0.7293  | 0.4500   | 0.7900   |         |         |         |         |
|                  | Pred-MutHTP#v    | 60         | test       | 0.9544   | 0.8898   | 0.9321  | 0.8480   | 0.9141   |         |         |         |         |
|                  | Pred-MutHTP#v    | 15         | test       | 0.9138   | 0.7481   | 0.7933  | 0.5400   | 0.8400   |         |         |         |         |
| Extracellular or Outside | MutTMPredictor | 25         | 10-fold    | 0.8664   | 0.8380   | 0.8542  | 0.7000   | 0.9100   |         |         |         |         |
|                  | Pred-MutHTP#v    | 19         | test       | 0.7335   | 0.7484   | 0.7450  | 0.4400   | 0.8100   |         |         |         |         |
|                  | Pred-MutHTP#v    | 19         | test       | 0.7871   | 0.7490   | 0.7724  | 0.5300   | 0.8400   |         |         |         |         |

Note: the SN, SP, ACC, MCC, and AUC values of Pred-MutHTP were collected from Pred-MutHTP [31].

being compared with specific predictors for predicting mutations in membrane proteins. Second, when searching mutations in the above three datasets, we found many mutations were not stored in the TMSNP database [35]. Hence, we were only able to calculate the evaluation metrics based on fewer mutations. Third, MutTMPredictor utilized the cascade XGBoost algorithm combined with a richer set of features, including evolutionary information, wild-type and mutant amino acids physicochemical properties, neighboring residue information, and four individual’s outputs. In summary, MutTMPredictor achieved a better performance than other predictors on all the three datasets.

3.12. Other performance comparison experiments

Except for the above comparison experiments, on the 67,584 mutations, Pred-MutHTP, and TMSNP datasets, we also performed other experiments to evaluate the effectiveness of MutTMPredictor further, described as follows:

(1) Reconstructing an objective test dataset based on 67,584 mutations dataset

We constructed an objective test dataset based on the 67,584 mutations dataset. In such test dataset, the training data of the four individual predictors did not overlap with each other. Performance comparison results and analyses can be found in Supplementary Tables S11-S14 and Text S6.

(2) Blind test on a third-party test dataset using MutTMPredictor

Herein, we performed additional blind test on a new third-party test dataset from the TMSNP database [35], which comprised 196,705 non-pathogenic, 2,624 pathogenic, and 437 likely pathogenic mutations in membrane proteins. More specifically, we performed three levels of blind test, including (I) test on the entire database, (II) test on three balanced sub-datasets, and (III) test only on pathogenic like pathogenic mutations. Supplementary Text S7 and Table S15 provide detailed descriptions of the dataset processing, bind test results, and the corresponding analyses.

(3) Performance comparison on three balanced sub-datasets from TMSNP
In order to compare MutTMPredictor with TMSNP, we performed comparison experiments on three balanced sub-datasets as external validation on the independent test data. Detailed comparison and analyses can be found in Supplementary Tables S16-S17 and Text S8.

(4) Removing training sequences from the blast database during the performance test

We also conducted comparison experiments to examine whether we need to discard training sequences from the blast database. The details can be found in Supplementary Tables S18-S19 and Text S9. According to the obtained results, we argue that it is unnecessary to discard the training sequences from the blast database during the testing and discard the test sequences from the blast database during training.

In summary, we conclude that MutTMPredictor is a robust mutation predictor with excellent prediction performance.

4. Conclusions

In this work, we have developed a new feature encoding algorithm based on evolutionary information, referred to WAPSSM. Moreover, we proposed a cascade XGBoost algorithm. Benchmarking experiments illustrate the effectiveness of the proposed WAPSSM and cascade XGBoost algorithms. Based on four types of features and cascade XGBoost, we developed a new mutation predictor named MutTMPredictor. Performance benchmarking
experiments on seven datasets demonstrate that MutTM Predictor is an effective predictor for transmembrane protein mutation prediction.

Three key factors can be attributed to the performance improvement of MutTM Predictor, including the weight attenuation for WAPSSM extraction, integration of the outputs of individual predictors, and cascade XGBoost. Despite its promising performance, MutTM Predictor also has some room for further improvement. For example, more effective mutation coding algorithms are anticipated to be developed and applied in the future work. In addition, it is also possible to develop ensemble deep learning models to further improve the predictive performance when more datasets in transmembrane proteins become available.

CRediT authorship contribution statement

Fang Ge: Conceptualization, Data curation, Methodology, Software, Writing-original draft. Yi-Heng Zhu: Methodology, Software.

Jian Xu Data: Visualization, Investigation. Arif Muhammad: Validation, Formal analysis. Jiangning Song: Writing – review & editing, Supervision. Dong-Jun Yu: Writing – review & editing, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (62072243, 61772273, and 61872186), the Natural Science Foundation of Jiangsu (BK20201304), the Foundation of National Defense Key Laboratory of Science and Technology (JZXY202001SY000901), the National Health and Medical Research Council of Australia (NHMRC) (APP1127948, APP1144652), the Australian Research Council (ARC) (LP110203333 and DP120104460), the National Institute of Allergy and Infectious Diseases of the National Institutes of Health (R01 AI111965), a Major Inter-Disciplinary Research (IDR) project awarded by Monash University, and the Natural Science Foundation of Anhui Province of China (K2018AO572).

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.csbj.2021.11.024.

References

[1] Quan L, Wu H, Lyu Q, Zhang Y. DAMpred: recognizing disease-associated nsSNPs through Bayes-guided neural-network model built on low-resolution structure prediction of proteins and protein-protein interactions. J Mol Biol Jun 14 2019;439(13):2449–59.
[2] Baranowski JF, Kalani MYS, Przybylowski CJ, Rajendran J. Corrigendum: a review: genetic conservation of proteins in protein–protein interactions resulting in disease pathogenesis. Front Surgery 2017;4:31.
[3] Capriotti E, Nehrt NL, Kann MG, Bromberg Y. Bioinformatics for personal genome interpretation. Brief Bioinform 2012;13(4):495–512.
[4] Hassan MS, Shaalan AA, Dessouky MT, Abdelaal EA, Elhefnawi MA. A review study: Computational techniques for expecting the impact of non-synonymous single nucleotide variants in human diseases. Gene Jan 5 2019;680:20–31.
[5] Choi Y, Chan AP. PROVEAN web server: a tool to predict the functional effect of amino acid substitutions and indels. Bioinformatics 2015;31(16):2745–7.
[6] Ng PC, Henikoff S. SIFT: predicting amino acid changes that affect protein function. Nucleic Acids Res Jul 1 2003;31(13):3824–2.
[37] Consortium GP. A global reference for human genetic variation. Nature 2015;526(7571):68–74.
[38] Lek M, Karczewski KJ, Minikel EV, Samocha KE, Banks E, Fennell T, et al. Analysis of protein-coding genetic variation in 60,706 humans. Nature 2016;536(7616):285–91.
[39] Forbes SA, Bindal N, Bamford S, Cole C, Kok CY, Beare D, Jia M, Shepherd R, Leung K, Menzies A, Teague JW, Campbell PJ, Stratton MR, Futreal PA. COSMIC: mining complete cancer genomes in the Catalogue of Somatic Mutations in Cancer. Nucleic Acids Res 2013;41(Database issue):D945–50.
[40] Landrum MJ, Lee JM, Riley CR, Jiang W, Ruhinstein WS, Church DM, et al. ClinVar: public archive of sequence and phenotypic variation. Nucleic Acids Res 2014;42(Database issue):D980–5.
[41] Burley SK, Berman HM, Bhikadiya C, Bi C, Chen L, Di Costanzo L, Christie C, Dalenberg C, Duarte JM, Dittla S, RCSB Protein Data Bank: biological macromolecular structures enabling research and education in fundamental biology, biomedicine, biotechnology and energy. Nucleic Acids Res 2010;40(Database issue):D464–74.
[42] Zhou H, Feng J. Deep forest[J]. National Science Review, 2019, 6(1): 74–86.
[43] Huang G, Liu Y, Van Der Maaten L, et al. Densely connected convolutional networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 4700–4708.
[44] Fu X, Niu B, Zhu Z, Wu S, Li W. CD-MIT: accelerated for clustering the next-generation sequencing data. Bioinformatics 2012;28(23):3150–2.
[45] Bair C, Apweiler R. The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. Nucleic Acids Res 2000;28(1):45–8.
[46] Karczewski KJ, Francioli LC, Tiao G, Cummings BB, Alföldi J, Wang Q, et al. The mutational constraint spectrum informs disease biology and guides variant aggregation. Science 2020;369(6500):1460–4.
[47] Popov P, Peng Y, Shen L, Stevens RC, Cherevov V, Liu JZ, et al. Computational design of thermostabilizing point mutations for G protein-coupled receptors. Proteins 2019;77(3):732–49.
[48] Kashwawish P, Pokarowski P, Pokarowska M, Kolinski A, Katayama T, Kanehsa M, Aanideg: amino acid index database, progress report 2008. Nucleic Acids Res 2008;36(Database issue):D202–5.
[49] Schiffer AA, Avadanei M, Ben-David M, Shavirin S, Spouge JL, Wolf YI, et al. Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements. Nucleic Acids Res 2007;35(1):57–63.
[50] Hu J, Li Y, Zhang M, Yang X, Shen HB, Yu DJ, Predicting Protein-DNA Binding Residues by Weightedly Combining Sequence-Based Features and Boosting Multiple SVMs. IEEE/ACM Trans Comput Biol Bioinform Nov-Dec'17;14(1):1058–68.
[51] Yu DJ, Shen HB, Yang JY. SOMPNN: an efficient non-parametric model for predicting transcriptional helices. Amino Acids 2012;42(2):2195–205.
[52] Henikoff S, Henikoff JC. Amino acid substitution matrices from protein blocks. Proc Natl Acad Sci 1992;89(22):10915–9.
[53] Shahab HA, Gough J, Cooper DN, Stenson PD, Barker GLA, Edwards KJ, et al. Predicting the functional, molecular, and phenotypic consequences of amino acid substitutions using hidden Markov models. Hum Mutat 2013;34(10):1671–87.
[54] Peng H, Long F, Ding C. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans Pattern Anal Mach Intell 2005;27(8):1226–38.
[55] Ramírez-Largo TT, Andrés J, Jover L, Marinacci C, Gómez-Cabrero D, Becskei Á, et al.拢aller: a comprehensive database of essential barley genes and their orthologs in 15 species. Nucleic Acids Res 2016;44(Database issue):D729–36.
[56] Liu HJ, Setiono R. Chi2: Feature selection and discretization of numeric data mining software: an update. ACM SIGKDD Explor Newslett 2009;11(1):279–324.
[57] Zou Z H, Feng J. Deep forest[J]. National Science Review, 2019, 6(1): 74–86.
[58] Danilev VA, Pyshmaev AS, Erokhov AV, et al. Predicting the functional, molecular, and phenotypic consequences of amino acid substitutions using hidden Markov models. Hum Mutat 2013;34(10):1671–87.
[59] 2014;510(7503):121–5.
[60] Tanaka M, Olsen RW, Medina MT, Schwartz E, Alonso ME, Duron RM, et al. Predicting the functional, molecular, and phenotypic consequences of amino acid substitutions using hidden Markov models. Hum Mutat 2013;34(10):1671–87.
[61] Lee EE, Ma J, Sacharidou A, Wu MT, Salisto VK, Nguyen N, et al. A protein kinase C phosphorylation site predictor using support vector machine and is mutated in GLUT1 deficiency syndrome. Mol Cell 2015;58(5):833–45.
[62] Johar Town, Lahore, Pakistan. His research interests include bioinformatics, pattern recognition, and machine learning.
Jiangning Song is an Associate Professor and group leader in the Monash Biomedical Discovery Institute and the Department of Biochemistry and Molecular Biology, Monash University, Melbourne, Australia. He is also affiliated with the Monash Centre for Data Science, Faculty of Information Technology, Monash University. His research interests include bioinformatics, computational biomedicine, machine learning, and pattern recognition.

Dong-Jun Yu received his Ph.D. degree from Nanjing University of Science and Technology in 2003. He is currently a full professor in the School of Computer Science and Engineering, Nanjing University of Science and Technology. His research interests include pattern recognition, machine learning, and bioinformatics. He is a senior member of the China Computer Federation (CCF) and a senior member of the China Association of Artificial Intelligence (CAAI).