Method Article

Deep learning program to predict protein functions based on sequence information

Chang Woo Ko [a,b], June Huh [c], Jong-Wan Park [a,b,*]

[a] *Department of Biomedical Sciences, Seoul National University College of Medicine, Seoul, Republic of Korea*
[b] *Department of Pharmacology, Seoul National University College of Medicine, Seoul, Republic of Korea*
[c] *Department of Chemical and Biological Engineering, Korea University, Seoul, Republic of Korea*

A B S T R A C T

Deep learning technologies have been adopted to predict the functions of newly identified proteins in silico. However, most current models are not suitable for poorly characterized proteins because they require diverse information on target proteins. We designed a binary classification deep learning program requiring only sequence information. This program was named 'FUTUSA' (function teller using sequence alone). It applied sequence segmentation during the sequence feature extraction process, by a convolution neural network, to train the regional sequence patterns and their relationship. This segmentation process improved the predictive performance by 49% than the full-length process. Compared with a baseline method, our approach achieved higher performance in predicting oxidoreductase activity. In addition, FUTUSA also showed dramatic performance in predicting acetyltransferase and demethylase activities. Next, we tested the possibility that FUTUSA can predict the functional consequence of point mutation. After trained for monooxygenase activity, FUTUSA successfully predicted the impact of point mutations on phenylalanine hydroxylase, which is responsible for an inherited metabolic disease PKU. This deep-learning program can be used as the first-step tool for characterizing newly identified or poorly studied proteins.

- We proposed new deep learning program to predict protein functions in silico that requires nothing more than the protein sequence information.
- Due to application of sequence segmentation, the efficiency of prediction is improved.
- This method makes prediction of the clinical impact of mutations or polymorphisms possible.

* Corresponding author at: Department of Biomedical Sciences, Seoul National University College of Medicine, Seoul, Republic of Korea.
E-mail address: parkjw@snu.ac.kr (J.-W. Park).

Specifications table

Subject Area:	Biochemistry, Genetics and Molecular Biology
More specific subject area:	Protein Function Prediction
Method name:	FUTUSA (function teller using sequence alone)
Name and reference of original method:	There is no original method
Resource availability:	https://github.com/snuhl-crain/FUTUSA

Overview

Advances in sequencing technologies help us discover a huge amount of protein variants that are generated from alternative mRNA splicing, incomplete translation, single amino acid polymorphism, or gene fusion. These mutations can lead to the expansion of protein functions, including protein-protein interactions, ligand binding, subcellular localization, or completely novel functions. Evolutionally, the sequence variations may be developed to overcome the limitation of gene numbers.

We designed a deep learning program for protein function prediction, using only amino acids and named it 'FUTUSA' (function teller using sequence alone). Compared with other baseline method, FUTUSA achieved a better performance in protein function prediction. In predicting rare GO terms, particularly, FUTUSA improved the efficiency of classification performance. We also successfully predict the contribution of each amino acid to protein function. Therefore, FUTUSA could detect the functional motifs in the case of newly identified proteins and may also help us predict the critical impact of some clinically identified mutations in disease progression.

Methods

Data source and preprocessing

To minimize human cognitive biases in machine training steps, we only used GO annotation data and protein sequence data. We downloaded UniProtKB/Swiss-Prot database (version 2020_02) (https://www.uniprot.org/downloads) [1]. The datasets included 17,965 human proteins and 6,195 yeast proteins. The hierarchical information of GO terms and GO annotation data was obtained from Gene Ontology Consortium (http://geneontology.org) [2,3]. We here used a GO OBO file (released on Sep 19th in 2018) containing 47,343 GO terms. The performance comparison was done with a previous GO file (released on Jan 1st 2016) containing 43,937 terms. The downloaded human and yeast GO annotation data files (released on Feb 2nd 2020) contained 495,361 annotations for human and 120,936 annotations for yeast. In this research, we used all evidence codes, including experimental, phylogenetically-inferred, and computational analyses. To compare ours with other prediction models, we used only EXP, IDA, IPI, IMP, IGI, IEP, TAS, and IC, which were considered as experimental evidence codes according to CAFA3 [4]. Using the datasets, we narrowed down the training target functions and constructed target specific dataset. For instance, if we picked one GO term such as "oxidoreductase activity", we labeled all proteins that were annotated the GO term and its 'is_a' relation descendants such as "sulfur reductase activity". We used protein segments composed of 16, 32, 64, or 128 amino acids for machine training. With this approach, we excluded zero padding step for fixing protein sequence information and reduced the file sizes of training data. We padded (size of segmentation-1) zeros at the N-terminus and (size of segmentation) zeros at the C-terminus of the sequence to give same weight to all amino acids except N-terminus methionine. Also, we didn't omit 'ambiguous' amino acids and considered all kinds of amino acids in sequence data. To reduce training time, however, we limited the maximum protein length catching up with each training purpose.

Deep learning architecture

We built the deep learning neural network models using Tensorflow2.0 framework and implemented them with the Keras deep learning library [5]. We used cloud-based environment, Google Colaboratory (https://colab.research.google.com) and trained the neural network models on GPUs offered by Google Cloud [6]. The models are organized with four subdivided sections, embedding layers, feature extraction, dense layers, and scoring steps. The first embedding layers convert preprocessed sequence data to numerical vector space that can be fed to the neural network. One hot encoding, one of the most common encoding methods, represent each amino acid with binary vector. This approach is straightforward and able to preserve original amino acid information, but it can cause the excessive complexity of model and cannot concern about physiochemical properties of amino acids. Recently, several research groups try to use machine-learning based encoding methods instead of manually define methods to solve the problems [7]. In this study, we used 1×1 CNN to generate amino acid embedding vector. And next, we extracted and learned the spatial features using CNN. The dense layers calculate the weight of previously generated feature map without concerning the topology, improving the recognition of distant features and concerns of their combinations for the classification. Lastly, the final output of binary classification from the dense layer are assumed as a predictive score of input segment and the total score for the input protein sequence is computed. We used the one-dimensional convolutional neural network (Conv1D) to extract the protein sequence-derived features. In Conv1D, 1D array-like kernel slides along one dimension and identify the patterns from sequence information. Conv1D has been widely used in sequence-based protein function prediction techniques [8–11]. 'FUTUSA' was built to get more flexible at the feature extraction steps. We added 1×1 CNN after the one-hot encoding to make the model also learn how to understand each amino acid [12]. Using 1D convolution neural network with max pooling layer, it extracted the features without reducing its size. This model used variously sized convolution kernels, 2 and its geometric progressions with common ratio 2 smaller than its segmentation size and prime number just prior every geometric series. The sequence-derived feature vectors were batch-normalized and activated by Rectified-linear-unit (ReLU) function [13,14]. Through the concatenation layer and flatten layer, the feature map was fed to fully connected layers. All of the intermediate hidden layers were batch-normalized and performed dropout regularization ($p = 0.2$) [15]. We activated intermediate hidden layers with the ReLU function and classification layer with Sigmoid function. The characteristics of the models are shown in Table S1.

Datasets and model training

For the evaluation purpose, we randomly extracted 5% of the total protein data before the training. Next, we validated the model during the training by using five-fold cross validation. We randomly allocated 20% of the remaining 95% protein data as a validation dataset and the 80% as a training dataset. For a case study, we picked the specific target protein and set it as evaluation data. In our models, the deep learning system was trained with Adam optimizer at a learning rate of 0.001 [16]. We used weighted binary cross entropy as a loss function to address imbalanced dataset issue of rare protein function prediction problem [17]. The weighted binary cross entropy loss L_{wBCE} can be written as

$$L_{wBCE}(y, \hat{y}) = -((1 - w)y \log(\hat{y}) + w(1 - y) \log(1 - \hat{y})) \tag{1}$$

Where w is the ratio of number of labeled proteins to total trained proteins, y is the real label and $\hat{y}$ is the predicted value. This loss function is designed to give more weight to relatively rarely labeled functions. The mini batch size was set to 128. To avoid overfitting, we trained deep learning models for varied epochs from 10 to 100 but less than 12 h and adopted early stopping strategy. The characteristics of used datasets are shown in Table S2.

Comparison methods

To compare our models with others in predictive performance, we adopted CAFA baseline method, BLAST. BLAST model predicts protein function based on sequence identity. We made some minor

changes in this model to predict a specific protein function because CAFA challenge originally aims to explore general functions of proteins. We made database with training dataset and found similar sequences with queried target proteins using BLASTP software [18]. The BLAST results contained the target training proteins after they were filtered by E-value of E-5.

Performance evaluation

We evaluated the predictive performance through average precision (AP), F1-score (F1), area under receiver operating characteristic curve (AUROC), area under precision-recall curve (AUPR) and Matthews correlation coefficient (MCC) [19–21]. The MCC, widely used confusion matrix describing method, MCC_t at threshold t can be computed directly from the confusion matrix using the following formula:

$$MCC_t = \frac{(TP_t \cdot TN_t) - (FP_t \cdot FN_t)}{\sqrt{(TP_t + FP_t)(TP_t + FN_t)(TN_t + FP_t)(TN_t + FN_t)}} \tag{2}$$

Where TP_t is the number of true positives; TN_t is the number of true negatives; FP_t is the number of false positives; FN_t is the number of false negatives at threshold t.

F1-score is harmonic average of the precision and the recall. The precision Pr_t and the recall Rc_t at threshold t can be written as

$$\Pr_t = \frac{TP_t}{TP_t + FP_t} \tag{3}$$

$$Rc_t = \frac{TP_t}{TP_t + FN_t} \tag{4}$$

Hence, F1-score $F_{1,t}$ can be calculated by the following formulas:

$$F_{1,t} = \frac{2}{\Pr_t{}^{-1} + Rc_t{}^{-1}} = 2 \cdot \frac{\Pr_t \cdot Rc_t}{\Pr_t + Rc_t} = \frac{2 \cdot TP_t}{2 \cdot TP_t + FP_t + FN_t} \tag{5}$$

In this study, we presented the maximum evaluation value for all thresholds computed with a step size 0.01 unless otherwise noted. The F1-score and the MCC score are visualized with the MCC-F1 curve to address the full range of all possible thresholds [22]. AP, AUROC, and AUPR are independent of the threshold. The receiver operating characteristic (ROC) curve is a plot of true positive rate (recall) versus false positive rate and precision-recall curve is a plot of precision versus recall. The AP is also the method to summarize the precision-recall curve. The true positive rate TPR_t and false positive rate FPR_t at threshold t can be written as

$$TPR_t = \frac{TP_t}{TP_t + FN_t} \tag{6}$$

$$FPR_t = \frac{FP_t}{TN_t + FP_t} \tag{7}$$

Hence, AUROC, AUPR, AP can be calculated by the following formulas:

$$AUROC = \int_{\infty}^{-\infty} TPR_t \cdot FPR_t{}' dt \tag{8}$$

$$AUPR = \int_{\infty}^{-\infty} \Pr_t \cdot Rc_t{}' dt \tag{9}$$

$$AP = \sum_n (Rc_n - Rc_{n-1}) \cdot \Pr_n \tag{10}$$

Where Pr_n and Rc_n are the precision and recall at the nth threshold.

Table 1

The overall evaluation results for all tested sequence segmentation sizes.

Model	AP	MCC	F1	AUPR	AUROC
FUTUSA_FL	0.3089	0.3863	0.3421	0.3058	0.7525
FUTUSA_16	0.4604	0.4764	0.4533	0.4576	0.8754
FUTUSA_32	0.4661	0.4413	.0.4494	0.4631	0.8872
FUTUSA_64	0.5158	0.5186	0.5319	0.5129	0.8835
FUTUSA_128	0.4612	0.5406	0.5135	0.4592	0.7671

Table 2

The performance comparison of the competing models on the oxidoreductase activity (GO:0016491), acetyltransferase activity (GO:0016407) and demethylase activity (GO:0032451) datasets.

	Programs	AP	MCC	F1	AUPR	AUROC
Oxidoreductase	BLAST	0.1509	0.3386	0.3014	-	-
	FUTUSA	0.4319	0.4508	0.4528	0.4272	0.8136
Acetyltransferase	BLAST	0.0649	0.1818	0.2374	-	-
	FUTUSA	0.3212	0.4444	0.5331	0.3166	0.7587
Demethylase	BLAST	0.1521	0.3529	0.3826	-	-
	FUTUSA	0.3486	0.5000	0.5145	0.3297	0.6906

Calculation of area under ROC curve and PR curve does not assess the performance of binary predictor, BLAST model.

Validation and discussion

Sequence segments-based training

First, we tested whether the sequence segmentation method improves the predictive performance of protein functions. We compared the segmentation model with the full-length model that uses a zero-padding layer to fit the length of the protein sequences. The fully connected layers were unaltered in this model. We trained FUTUSA to predict oxidoreductase activity (GO:0016491) and its descendants, which were 2371 terms. We used two different scoring methods to determine the score of each protein. One used the average for the predictive scores of all segments from the protein, and the other did the average for only the highest-scored segments in the top 10% to reduce the error from the protein lengths and non-functional residues. Because the performances of the methods depend on the context, we selected the better scoring method whenever the model training was completed. In all five metrics (AP, F1, AUROC, AUPR, and MCC), the sequence segmentation improved the classification performance regardless of the size of segmentation and CNN architecture. Fig. 1. displays the MCC-F1 curve, ROC curve, and P-R curve for various segmentation sizes. All the sequence segmentations were found to be better than the full-length process. Of these segmentations, the 64 amino acid segmentation was shown as the best setting. The overall results are summarized in Table 1. The results revealed that the machine learning with sequence segments significantly improves the ability of protein feature recognition.

Predictive performance comparison

To verify our approach, we compared our model with the baseline method BLAST. We selected oxidoreductase activity (GO:0016491, 721 annotated proteins) as a representative case for the big-sized dataset (>100 proteins), and the acetyltransferase activity (GO:0016407, 81 annotated proteins) and demethylase activity (GO:0032451, 28 annotated proteins) as the cases for the small-sized datasets. We trained FUTUSA with 64 amino acid segmentation. Table 2 shows the predictive performances of various models on the three different datasets. In case of oxidoreductase, FUTUSA obtained AP = 0.4319, MCC = 0.4508, and F1 = 0.4528, which are higher than those in BLAST. We here adopted acetyltransferase and demethylase as the cases for small dataset size. FUTUSA showed higher values of AUPR (0.3166 for acetyltransferase; 0.3297 for demethylase) than BLAST model.
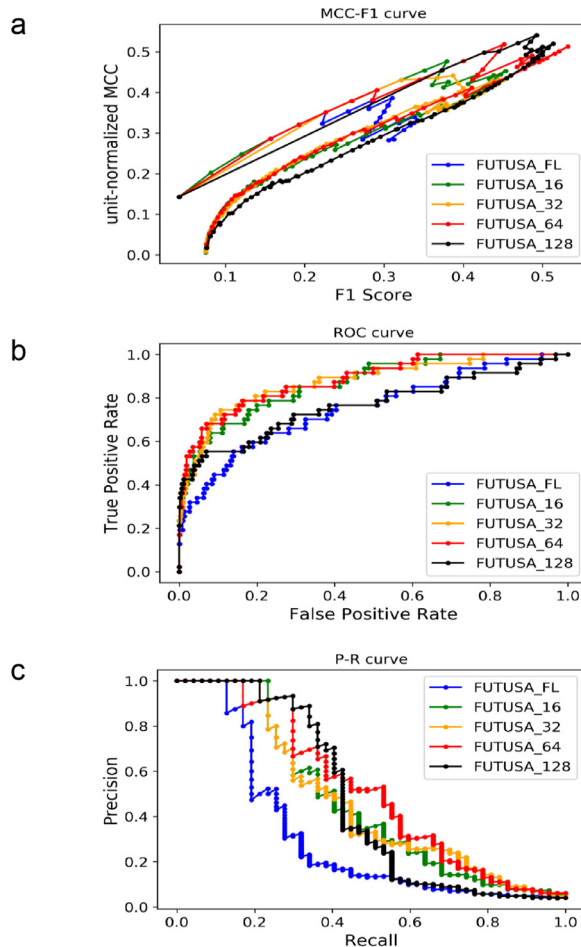
Fig. 1. The MCC-F1, ROC, P-R curves of all tested sequence segmentation sizes. The MCC-F1 (a), ROC (b), P-R (c) curves (FL: full-length model (blue); 16: segmentation size 16 model (green); 32: segmentation size 32 model (orange); 64: segmentation size 64 model (red); 128: segmentation size 128 model (black).

Consequently, we propose FUTUSA as a powerful protein function-predicting program applicable to either big-sized or small-sized dataset.

Prediction of the functional impacts of mutations in phenylalanine hydroxylase

To explore a new application of FUTUSA, we investigated whether it could predict the functional consequence of single amino acid variation. To evaluate the contribution of each amino acid to functional property, we computed the average values of all segments in phenylalanine hydroxylase (PAH), which is responsible for an inherited metabolic disease PKU. PAH, which belongs to the monooxygenase family, is a homo-tetrameric enzyme composed of an N-terminal regulatory domain, a central catalytic domain, and a C-terminal tetramerization domain [23]. Clinically, it is very important to predict the functional consequences of the PAH mutations in patients' specimens. Such an information may be one of guiding factors to decide how to care PKU patients. To predict PAH activity, we trained FUTUSA with new dataset comprising 114 monooxygenases. The trained model successfully predicted the catalytic domain as a crucial region and assigned low scores to regulatory

and tetramerization domains (Fig. 2a). The N-terminus is classified as the ACT domain, which is widely expressed in enzymes of amino acid metabolism. Notably, this region obtained a low score even though there were several identical residues with other aromatic amino acid hydroxylases. It may be interpreted that this domain is not essential for monooxygenase activity. To visualize the prediction results, we highlighted the high score region in the 3D structure of human PAH protein using CAVER Analyst 2.0 [24–26]. Previous studies reported that phenylalanine forms hydrophobic interactions with R270, T278, P279, F331, G346, G349, and S350. The thiophene ring of the substrate was stacked against the imidazole ring of H285 [25,27]. The high score region of the model was close to the active site of human PAH. Fig. 2b shows that they covered most of the substrate-binding residues and iron-binding residues. The low score regions do not contact the substrate and are located far from the active site. In particular, the prediction result of FUTUSA covered the active site lid, Y138 [28].

Next, we computed the change in the predictive score after single amino acid deletion and substitution. There are 100 or more PAH mutations (missense and frameshift) that can lead to phenylketonuria (PKU), as reported in the ClinVar database [29]. These mutations present a broad range of phenotypic variations depending on the residual enzymatic activities. Hence, we verified the predictive performance of the full-length models and segmentation models with the mutated sequences. As a result, FUTUSA_FL (full-length) failed to predict the outcomes of single amino acid changes (Fig. 3a). However, the segmentation models composed of 16 (Fig. 3b) and 64 (Fig. 3c) amino acids were able to compute remarkable score changes after mutation occurs. FUTUSA_16 showed score drops only in several crucial regions, but FUTUSA_64 marked crucial regions more widely. The regulatory and oligomerization domains showed trivial score changes in both segmentation models. FUTUSA_64 showed that many substitutions of amino acids in the catalytic domain decreased the predictive score, but those in exons 4 and 11 did not substantially (Fig. 3c). Numbers of missense variants and exon deletions, related to loss of function and phenylketonuria, were detected in the catalytic domain [30,31]. These results reveal that the functional sites predicted by FUTUSA are well matched with the known functional domains.

Discussion

We here propose a deep learning-based protein function predictor, which is named FUTUSA. Since it requires only the sequence information, FUTUSA is suitable for preliminary characterization of newly found proteins or uncharacterized variants. We also established a preprocessing method for sequence segmentation, which effectively extracts functional features from protein sequence data and augments the performance of function prediction. FUTUSA showed a better performance than the baseline method, BLAST. Therefore, we propose FUTUSA as a new deep learning program predicting the functions of uncharacterized proteins. Additionally, FUTUSA distinguished functionally essential amino acids with nonessential ones, suggesting that FUTUSA could be used for predicting the clinical impact of point mutations or single amino acid polymorphisms.

Generally, protein function prediction has a fundamental problem resulting from training with imbalanced dataset. For instance, a number of oxidoreductase proteins is 719, which is only 3.4% of total proteins trained in FUTUSA. It means that the machine was trained more with irrelevant proteins (as negative controls) than with target proteins. Especially in the small-sized dataset such as acetyltransferase and demethylase, this class imbalance in training raises the false-positive rate to a greater extent. Reviewing the equations for AUPR and AUROC, AUROC is substantially affected by the false-positivity, whereas AUPR does not [32]. In case of the small-sized dataset, therefore, AUPR may evaluate the efficiency of function prediction more accurately than AUROC. For this reason, we emphasize the AUPR values (Table 2).

It should be noted that FUTUSA has some limitations in the data process. The sequence segmentation process increases the input data size, thereby increasing the training time. Thus, the segmentation should be optimized to achieve a good balance between the predictive performance and training time. Also, since FUTUSA is not a ready-to-use predictor, users should modify training dataset, optimize preprocessing parameters, and train it according to their purposes.

One of the reasons for the delay in the development of deep learning-based protein function predictors is that the performance of deep learning models depends on the quality and quantity
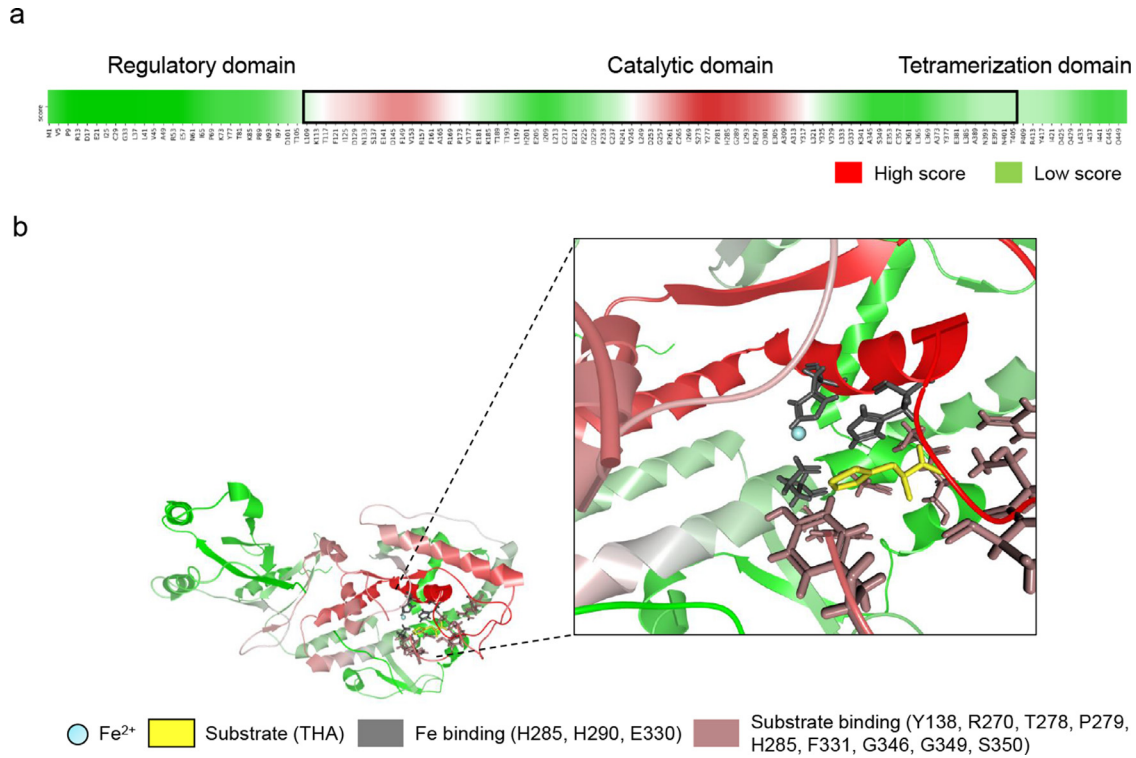
Fig. 2. The heatmap visualization of predicted functional contribution score of individual amino acids. The predicted scores of FUTUSA are also overlaid onto crystal structure of the full-length human PAH (residues 21–446; PDB:6N1K) and catalytic domain of human PAH (residues 117-428; PDB:1MMK). (a) The heatmap is mapping with green as low predictive score and red as high predictive score. (b) The iron ion (cyan) is highlighted in balls. The substrate analogue, beta-2-Thienylalanine (THA; yellow) and amino acid residues of binding pockets (gray and brown) are presented as sticks. The prediction was performed with Phenylalanine-4-hydroxylase for monooxygenase activity.
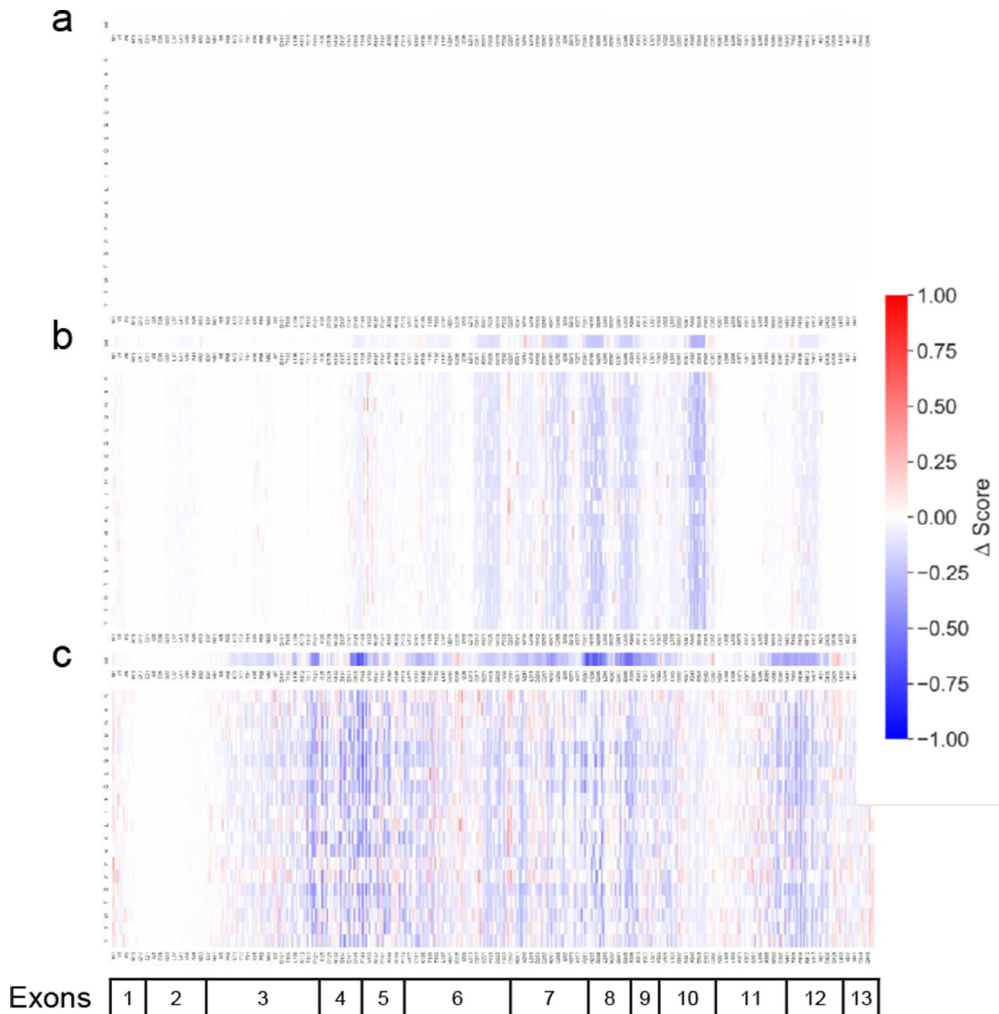
Fig. 3. The heatmap of the score changes by single amino acid mutations. Protein function changes by point mutations were predicted using FUTUSA_FL (a), FUTUSA_16 (b), FUTUSA_64 (c) The color indicates the score changes after mutation, blue as decreased score and red ad increased score. Each column represents the position of the amino acid and each row represents the changed amino acid after mutation. The first row, del, indicates the deletion of the amino acid. The prediction was performed with Phenylalanine-4-hydroxylase for monooxygenase activity.

of training data. Therefore, some researchers used diverse protein features, such as protein-protein interactions or protein motifs [8,9,33]. However, the features are available for extensively studied proteins, but not for uncharacterized proteins. To solve such a problem, many researchers have tried to develop new protein function predictors requiring only amino acid sequences [10,34,35]. Despite many efforts, the sequence-alone approaches have not been satisfactory to users. In this context, FUTUSA is proposed as a new method to meet users' needs.

In a future work, we plan to modify the preprocessing process. In the present study, we assigned the same weight to all segmented sequences, even if the sequences insignificantly contributed to protein function. It is expected that assigning different weights to each segment can minimize artificial biases. For that, we should add a step to re-evaluate how much each segment contributes

to the whole function. In addition, the model should be changed to a multi-class classification model. This version of the model was built as a single-class classification model to focus its predictive ability in classifying the protein function. However, it is evident that the hierarchical structure of GO terms and the corresponding annotated patterns also contain important information. Therefore, we will intend to find the GO terms grouping method to optimally predict a single GO term.

All the source code and datasets are available at https://github.com/snuhl-crain/FUTUSA.

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

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.mex.2022.101622.

References

[1] A. Bateman, UniProt: a worldwide hub of protein knowledge, Nucleic Acids Res. 47 (D1) (2019) D506–D515.
[2] M. Ashburner, C.A. Ball, J.A. Blake, D. Botstein, H. Butler, J.M. Cherry, et al., Gene ontology: tool for the unification of biology, Nat. Genet. 25 (1) (2000) 25–29.
[3] S. Carbon, E. Douglass, N. Dunn, B. Good, N.L. Harris, S.E. Lewis, et al., The gene ontology resource: 20 years and still GOing strong, Nucleic Acids Res. 47 (D1) (2019) D330–D338.
[4] N. Zhou, Y. Jiang, T.R. Bergquist, A.J. Lee, B.Z. Kacsoh, A.W. Crocker, et al., The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens, Genome Biol. 20 (1) (2019) 1–23.
[5] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, et al., TensorFlow: a system for large-scale machine learning, in: Proceedings of the 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16) [Internet], {USENIX} Association, Savannah, GA, 2016, pp. 265–283. https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi. Available from.
[6] T. Carneiro, R.V.M. Da Nobrega, T. Nepomuceno, B.G. Bin, V.H.C. De Albuquerque, P.P.R. Filho, Performance analysis of google colaboratory as a tool for accelerating deep learning applications, IEEE Access 6 (2018) 61677–61685.
[7] X. Jing, Q. Dong, D. Hong, R. Lu, Amino acid encoding methods for protein sequences: a comprehensive review and assessment, IEEE/ACM Trans. Comput. Biol. Bioinform. 17 (6) (2020) 1918–1931.
[8] Y. Cai, J. Wang, L. Deng, SDN2GO : an integrated deep learning model for protein function prediction, Front. Bioeng. Biotechnol. 8 (April) (2020) 1–11.
[9] M. Kulmanov, M.A. Khan, R. Hoehndorf, DeepGO : predicting protein functions from sequence and interactions using a deep ontology-aware classifier, Bioinformatics 34 (2018) 660–668 October 2017.
[10] R. Hoehndorf, DeepGOPlus : improved protein function prediction from sequence, Bioinformatics 36 (2020) 422–429 July 2019.
[11] G. Zhou, J. Wang, X. Zhang, G. Yu, DeepGOA: predicting gene ontology annotations of proteins via graph convolutional network, in: Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2020, pp. 1836–1841.
[12] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S.E. Reed, D. Anguelov, et al., Going deeper with convolutions, CoRR (2014) [Internet]abs/1409.4. Available from http://arxiv.org/abs/1409.4842.
[13] Sergey Ioffe, Szegedy Christian, Batch normalization: Accelerating deep network training by reducing internal covariate shift. International conference on machine learning, PMLR (2015).
[14] X. Glorot, A. Bordes, Deep sparse rectifier neural networks, in: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, 15, 2011, pp. 315–323.
[15] G. Hinton, Dropout : a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (2014) 1929–1958.
[16] D.P. Kingma, J.L. Ba, Adam: a method for stochastic optimization, in: Proceedings of the International Conference on Learning Representations (ICLR), 2014, pp. 1–15.
[17] Y.S. Aurelio, G.M. de Almeida, C.L. de Castro, A.P. Braga, Learning from imbalanced data sets with weighted cross-entropy function, Neural Process. Lett. 50 (2) (2019) 1937–1949 [Internet]Available from, doi:10.1007/s11063-018-09977-1.
[18] C. Camacho, G. Coulouris, V. Avagyan, N. Ma, J. Papadopoulos, K. Bealer, et al., BLAST+: architecture and applications, BMC Bioinform. 10 (2009 Dec) 421.
[19] A. Hanley, J. Mcneil, The meaning and use of the area under a receiver operating characteristic (ROC) curve, Radiology 143 (1) (1982) 29–36.

[20] J. Davis, M. Goadrich, The relationship between precision-recall and ROC curves, in: Proceedings of the 23rd International Conference on Machine Learning, 2006, pp. 233–240.

[21] B.W. Matthews, Comparison of the predicted and observed secondary structure of T4 phage lysozyme, Biochim. Biophys. Acta Protein Struct. 405 (1975) 442–451.

[22] C. Cao, D. Chicco, M.M. Hoffman The MCC-F1 curve : a performance evaluation technique for binary classification. arXiv Prepr arXiv200611278. 2020;1–17.

[23] M.I. Flydal, A. Martinez, Phenylalanine hydroxylase: Function, structure, and regulation, IUBMB Life 65 (4) (2013) 341–349 [Internet]Apr 1Available from, doi:10.1002/iub.1150.

[24] E.C. Arturo, K. Gupta, M.R. Hansen, E. Borne, E.K. Jaffe, Biophysical characterization of full-length human phenylalanine hydroxylase provides a deeper understanding of its quaternary structure equilibrium, J. Biol. Chem. 294 (26) (2019) 10131–10145 [Internet]Jun 28Available from http://www.jbc.org/content/294/26/10131.abstract.

[25] O.A. Andersen, A.J. Stokka, T. Flatmark, E. Hough, 2.0Å resolution crystal structures of the ternary complexes of human phenylalanine hydroxylase catalytic domain with tetrahydrobiopterin and 3-(2-Thienyl)-l-alanine or l-norleucine: substrate specificity and molecular motions related to substrate binding, J. Mol. Biol. 333 (4) (2003) 747–757 [Internet]Available from http://www.sciencedirect.com/science/article/pii/S002228360301115X.

[26] A. Jurcik, D. Bednar, J. Byska, S.M. Marques, K. Furmanova, L. Daniel, et al., CAVER analyst 2.0: analysis and visualization of channels and tunnels in protein structures and molecular dynamics trajectories, Bioinformatics 34 (20) (2018) 3586–3588 [Internet]Oct 15Available from, doi:10.1093/bioinformatics/bty386.

[27] O. Andreas Andersen, T. Flatmark, E. Hough, Crystal structure of the ternary complex of the catalytic domain of human phenylalanine hydroxylase with tetrahydrobiopterin and 3-(2-Thienyl)-l-alanine, and its Implications for the mechanism of catalysis and substrate activation, J. Mol. Biol. 320 (5) (2002) 1095–1108 [Internet]Available from http://www.sciencedirect.com/science/article/pii/S0022283602005600.

[28] M.I. Flydal, M. Alcorlo-Pagés, F.G. Johannessen, S. Martínez-Caballero, L. Skjærven, R. Fernandez-Leiro, et al., Structure of full-length human phenylalanine hydroxylase in complex with tetrahydrobiopterin, Proc. Natl. Acad. Sci. 116 (23) (2019) 11229–11234 [Internet]Available from https://www.pnas.org/content/116/23/11229.

[29] M.J. Landrum, J.M. Lee, M. Benson, G.R. Brown, C. Chao, S. Chitipiralla, et al., ClinVar: Improving access to variant interpretations and supporting evidence, Nucleic Acids Res. 46 (D1) (2018) D1062–D1067.

[30] C. Chen, Z. Zhao, Y. Ren, X. Kong, Characteristics of PAH gene variants among 113 phenylketonuria patients from Henan Province, Zhonghua Yi Xue Yi Chuan Xue Za Zhi 35 (6) (2018) 791–795 Dec.

[31] Y. Lee, D.H. Lee, N. Kim, S. Lee, J.Y. Ahn, T. Choi, et al., Mutation analysis of PAH gene and characterization of a recurrent deletion mutation in Korean patients with phenylketonuria, Exp. Mol. Med. 40 (5) (2008) 533–540.

[32] T. Saito, M. Rehmsmeier, The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets, PLoS One 10 (3) (2015) e0118432.

[33] F. Zhang, H. Song, M. Zeng, Y. Li, L. Kurgan, M. Li, DeepFunc: A deep learning framework for accurate prediction of protein functions from protein sequences and interactions, Proteomics 19 (12) (2019) 1–7.

[34] N. Strodthoff, P. Wagner, M. Wenzel, W. Samek, UDSMProt: universal deep sequence models for protein classification, Bioinformatics 36 (8) (2020) 2401–2409 [Internet]Available from, doi:10.1093/bioinformatics/btaa003.

[35] A. Ranjan, M.S. Fahad, D. Fernández-Baca, A. Deepak, S. Tripathi, Deep robust framework for protein function prediction using variable-length protein sequences, IEEE/ACM Trans. Comput. Biol. Bioinform. 17 (5) (2020) 1648–1659.