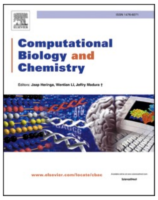
Computational Biology and Chemistry 95 (2021) 107593

Contents lists available at ScienceDirect

Computational Biology and Chemistry

journal homepage: www.elsevier.com/locate/cbac



Research article

Protein function prediction using functional inter-relationship

Richa Dhanuka [*], Jyoti Prakash Singh

Department of Computer Science and Engineering, National Institute of Technology Patna, India

ARTICLE INFO

Keywords:
Protein function
Multi layer Neural Network
Correlation
Functional redundancy

ABSTRACT

With the growth of high throughput sequencing techniques, the generation of protein sequences has become fast and cheap, leading to a huge increase in the number of known proteins. However, it is challenging to identify the functions being performed by these newly discovered proteins. Machine learning techniques have improved traditional methods' efficiency by suggesting relevant functions but fails to perform well when the number of functions to be predicted becomes large. In this work, we propose a machine learning-based approach to predict huge set of protein functions that use the inter-relationships between functions to improve the model's predictability. These inter-relationships of functions is used to reduce the redundancy caused by highly correlated functions. The proposed model is trained on the reduced set of non-redundant functions hindering the ambiguity caused due to inter-related functions. Here, we use two statistical approaches 1) Pearson's correlation coefficient 2) Jaccard similarity coefficient, as a measure of correlation to remove redundant functions. To have a fair evaluation of the proposed model, we recreate our original function set by inverse transforming the reduced set using the two proposed approaches: Direct mapping and Ensemble approach. The model is tested using different feature sets and function sets of biological processes and molecular functions to get promising results on DeepGO and CAFA3 dataset. The proposed model is able to predict specific functions for the test data which were unpredictable by other compared methods. The experimental models, code and other relevant data are available at https://github.com/richadhanuka/PFP-using-Functional-interrelationship.

1. Introduction

Proteins serve as building blocks and functional components of a living cell. They are responsible for performing various functions inside a living organism, such as reproduction, cell growth, and metabolism. Proteins are an essential functional ingredient of life, and knowledge about their functions becomes a crucial link for developing new drugs, biofuels, and other biochemicals.

In terms of protein, a function can be defined as “anything that happens to or through a protein" Rost et al. (2003). Gene Ontology (GO) is a comprehensive, structured vocabulary of protein functions created to ease the computational reasoning about the functions Hill et al. (2008). It represents the functions of a protein and associates it with a GO term. The Gene Ontology (GO) Consortium[1] Ashburner et al. (2000), Consortium (2018) categorizes protein functions into three major groups: (i) Molecular Function (MF), (ii) Biological Process (BP), and (iii) Cellular Component (CC). Molecular Functions are defined as the activities that occur at the molecular level, e.g., metal ion binding, catalytic activity. Biological processes involve a series of molecular functions performed in an ordered manner, e.g., cell killing, reproduction. Cellular Component is the location inside the cell where protein activates, e.g., nucleus, cytoplasm. In GO, the protein functions are represented as GO: XXXXXXX, where X is a number. An example GO term in BP is “GO:0000003," which represents the “reproduction" function. GO contains a huge number of GO terms, which constitutes approximately 4045 terms related to CC, 10543 terms related to MF, and 29385 terms related to BP Carbon et al. (2008).

Protein function prediction (PFP) is defined as the task of predicting functions for a given protein. The traditional approaches for protein function prediction are experimental and are mainly conducted in laboratories, making these approaches slow, expensive, and resource-intensive. Due to these constraints, the experiments are mainly concentrated on a few important proteins and functions Pandey et al. (2006). After the success of new generation sequencing techniques, protein sequences are being generated cheaply and abundantly. As the traditional annotation processes are slow, the number of known

[*] Corresponding author.
E-mail addresses: richa.dhanuka@gmail.com (R. Dhanuka), jps@nitp.ac.in (J.P. Singh).

[1] http://www.geneontology.org/

https://doi.org/10.1016/j.compbiolchem.2021.107593
Received 21 April 2021; Received in revised form 25 August 2021; Accepted 3 October 2021
Available online 16 October 2021

unannotated protein sequences is continuously increasing. UniProt Boutet et al. (2016) data shows that only 0.5% of the discovered proteins are annotated.[2] This huge gap between proteins with known functions and proteins with unknown functions has attracted researchers from the computational field to develop computational methods to help biologists by suggesting probable functions for a protein. Biologists can later validate these findings through manual experiments. The protein function prediction by computational techniques reduces the search space for biologists to experiment with increasing traditional methods' efficiency.

The research community has been continuously showing interest in developing computational approaches for protein function prediction (Chen et al., 2016; Godzik et al., 2007; Zhou et al., 2019; Wang et al., 2018; Cozzetto et al., 2016; You et al., 2019). The initial approaches for PFP were homology-based transfer, which annotates the proteins with their corresponding functions based on the similar proteins in the database Sleator and Walsh (2010). Basic Local Alignment Search Tool (BLAST) Altschul et al. (1990) is one of the earliest homology-based transfer approaches that find matches between sequences and calculates the statistical significance of those matches to annotate a sequence based on that. BLAST gained recognition in PFP, and many variations like PSI-BLAST, Gapped BLAST Altschul et al. (1997) and further optimizations Li et al. (2014) were built. These techniques were based on complex string matching algorithms, which makes them computationally expensive. Yu and Huang Yu and Huang (2013) proposed an alignment-free sequence comparison method and achieved highly coinciding results with the traditional alignment-based approaches. They showed that the vectors are numerical representations of sequences that can preserve the sequence information in numerical form Yu and Huang (2013) which opens up a vast arena of feature selection methods Wang et al. (2016), Jing et al. (2021) on protein sequences. Later, Asgari and Mofrad Asgari and Mofrad (2015) created ProtVec representation for protein sequences inspired by the Word2Vec Goldberg and Levy (2021) models for creating word embeddings in the protein sequence. ProtVec gave outstanding results in protein family classification.

Deep learning and machine learning techniques are currently outperforming traditional methods in various fields like text mining, speech recognition, etc. It has also gained recognition and popularity in PFP for its capability of learning from a wide range of data Zhang et al. (2018); You et al. (2018); Sønderby et al. (2015); Hou et al. (2019a), Hou et al. (2019b). One such work is proposed by Cao et al. (2017), where a neural machine translation model is built using Recurrent Neural Network (RNN). They viewed protein sequences as one language and the corresponding functions as another language. Kulmanov et al. (2017) proposed a model using the Convolution Neural Network (CNN) to predict the protein functions using protein sequences and protein interaction data. They have collected data from the SwissProt and STRING database. They received an F1-score of 0.36 for BP, 0.46 for MF, and 0.63 for CC. In Miranda and Hu (2018), Miranda and Hu have proposed a machine learning model using stacked autoencoders and Support Vector Machine (SVM). They have extracted features using deep-stacked denoising autoencoders, and used a multi-labeled SVM for classification. They achieved an F_{max} of 0.593 and 0.993 for yeast and genbase datasets, respectively. In Chicco et al. (2014), researchers have used deep autoencoders for gene function annotation. They focused on the Bos taurus (cattle) and Gallus gallus (red junglefowl) gene datasets.

The other aspect which the research community of PFP has kept an eye on is the inter-relationship of functions Pandey et al. (2009); Yu et al. (2017); Barutcuoglu et al. (2006); Eisner et al. (2005); Masseroli et al. (2012); Sun et al. (2018). The hierarchical structure of gene ontology, being the most comprehensive form of representing those relationships has been used in protein function prediction. Kulmanov et al. (2017) have used this hierarchical structure of GO in the form of a hierarchical classification layer to predict functions. The main disadvantage of using a hierarchical classification layer is the complexity being added to the model in terms of the number of trainable parameters, run-time, and the model's training time. A few recent research has proposed transforming the GO terms into simple computational forms and used it further to predict protein functions. Makrodimitris et al. (2019) have transformed the GO terms (with redundancy) into a compact latent representation using the Label-Space Dimensionality Reduction technique and outperforms the state-of-the-art methods. Another effort was made by Ahmed et al. (2011) to determine the correlation between the functions using an overlapping number of proteins and interactions over protein clusters, which improved the degree of certainty and accuracy of protein function prediction in yeast proteome. Meng et al. (2016) have also used functional correlation to enhance the performance of protein function prediction. They defined a functional relationship in the form of inter-functional similarity and protein interaction similarity. The usage of multiple datasets gave them better coverage and quality data to predict the functions. Incorporating inter-relationships of functions into any model reduces the model's complexity by reducing the redundancy among functions, reducing the ambiguity in the model training.

To the best of our knowledge, none of these works has utilized statistical approach to define functional inter-relationship and included it into a machine learning model to make a better performing system for protein function prediction. DeepPred Rifaioglu et al. (2019) and DeepGO Kulmanov et al. (2017) are the only works we found which have used machine learning with hierarchical GO information to predict protein functions. DeepPred Rifaioglu et al. (2019) has used a hierarchical stack of a multi-task feed-forward network to predict GO-based protein function prediction, which performed better than other compared state-of-the-art methods. DeepGO has used deep CNN and the hierarchical classification module, which considers the functional dependencies to optimize the predictive performance on the complete GO hierarchy Kulmanov et al. (2017). The inclusion of a deep hierarchical classification layer increases the model's complexity, increasing the model's run time and training time. The major disadvantage of the technique is the requirement of substantial computational resources. Later, the same research group developed DeepGOPlus Kulmanov and Hoehndorf (2020) in which they removed the hierarchical classification layer due to time complexity and memory limitations.

In the proposed work, we have leveraged the machine learning capabilities of learning from diverse data, adding functional inter-relationship benefits to make a better performing model for protein function prediction. We propose a multi-layer perceptron (MLP) based model which incorporates protein function inter-relationships to find functions performed by protein sequences. In this approach, the co-occurrence of protein functions is used to define the protein functions' inter-relationship. If two functions are being performed together by many different proteins, these two functions are considered interrelated. It is assumed that if a protein performs one of the inter-related functions, it will perform the other function too. To quantify the inter-relationship between these functions, we calculate Pearson's correlation coefficient and Jaccard similarity index separately for each protein function pair and determine the distance between them. We empirically decide a threshold for the distance between functions to be considered redundant to form various pairs of most redundant functions. We train a multi-layer perceptron network (MLP), with input as protein sequences and output as the corresponding set of less redundant functions. The predicted output from the MLP is the probability of each protein function being performed by a sequence. Once we get the probabilities for each function (from the set of less redundant functions) being performed by a protein sequence, we generate the resultant probabilities for other redundant functions too which helps in determining a fair evaluation of the proposed model. Two approaches are used to generate the probabilities for redundant functions 1) Direct Mapping 2) Ensemble

[2] https://www.uniprot.org/ accessed on July, 2019.

approach, which is explained in detail in Section 2.3 and Section 2.4 respectively. Our proposed MLP model outperforms other states of the art model in terms of F_{max}, average precision, and average recall. It is found that training the model by considering functional inter-relationships can predict the functions that were not being predicted by earlier proposed models like MLDA Wang et al. (2010) and DeepGO Kulmanov et al. (2017).

The main contributions of this work are summarized as: .

- **Creation of non-redundant reduced functions set using similarity measures**: It has led to improved predictability of the functions as measured through F_{max}, average precision, average recall, and area under the precision-recall curve (AUPR) by reducing the redundancy of the functions and ambiguity of the proposed model.
- **A MLP based model to predict the functions which utilizes functional inter-relationships**: The functional inter-relationship is included to build MLP based model with enhanced capabilities to predict the protein functions.
- **Recreating the prediction of the reduced functions set into the original set of functions**: It is found that if the number of functions to be predicted is small, the models tend to perform better. Hence, we evaluate the proposed model on the original functions set by recreating the predictions of the reduced functions set.
- **Increase in the predictability of specific functions**: The proposed model that utilizes functional inter-relationships can predict specific functions like positive regulation of cytoplasmic transport, NADP binding, etc (see Table 8 for complete list) which were unpredictable by other compared methods.

2. Materials and methods

In this work, protein sequences are used as input, and corresponding function annotations are used as an output to train a multi-layer perceptron (MLP) for predicting the protein function. Fig. 1 represents the block diagram for this. The model is trained to predict a reduced set of non-redundant functions created using inter-relationship of functions. Later, the predictions for the non-redundant reduced function set is used to create the predictions for the complete set of functions. For the recreation of the removed functions, we use two approaches. First, Direct mapping in which the value for the removed set of functions is populated as same as the similar redundant function's predicted values. Second, the ensemble approach in which we automate this by combining the results of models trained with a non-redundant set of functions, to train an MLP with output as a complete set of functions. Lastly, the model is tested and evaluated on the unseen test data.

The rest of this section elaborates about the dataset used, the formation of feature vectors by pre-processing the protein sequences, the creation of reduced function sets to be used as an output to the MLP, proposed model, and the two processes to recreate the prediction for complete function set.

2.1. Data description and pre-processing

We have used the data [3] of Kulmanov et al. (2017) and CAFA3 Zhou et al. (2019) data [4] to train and test our model. The DeepGO dataset has 932 functions for BP and 589 functions for MF, while the CAFA3 dataset has 3352 functions for BP and 617 functions for MF. Table 1 shows the statistics of the data used in our work. The DeepGO dataset is SwissProt's manually reviewed and annotated protein sequences with GO function annotations. This was originally downloaded from UniProt and contained annotations with experimental evidence codes (EXP, IDA, IPI, IMP, IGI, IEP, TAS, and IC). The dataset does not contain any ambiguous amino acids (B, O, J, U, X, Z).

To make the data suitable for MLP to learn, we need to pre-process the input and output data. The subsequent sections give details about pre-processing, implementation, and methodology involved.

2.1.1. Input data preparation

In this work we create a normalized frequency feature vector (NFFV) for protein sequences which is inspired by bag of words model from text processing. Similar to this model, a sequence is represented as a collection of predefined terms called as k-mers. k-mers are words formed using k consecutive amino acids in a sequence. The proposed method has been investigated with 1-mer (k = 1), 2-mer (k = 2) and 3-mer (k = 3) for initial experiments. The F_{max}, average precision, average recall on DeepGO dataset for k = 1 are 0.29, 0.51, 0.20, for k = 2 are 0.36, 0.48, 0.29, for k = 3 are 0.33, 0.39, 0.28 respectively. As the performance evaluated is maximum at 2-mer. Hence, we proceeded the rest of the experiment with 2-mer keeping the check on number of features along with the complexity of the model. There can be at most 400 combinations of these 2-mers, as there exists 20 known amino acids. Every i^{th} protein sample P_i, is represented in the form of a normalized frequency feature vector V_i of size 400. NFFV for each protein P_i is calculated as discussed below: $V_i = \{v_{1i}(AA), v_{2i}(AC), v_{3i}(AD), \ldots\ldots, v_{400i}(TT)\}$.

where each component $v_{ki}\ (k = 1, 2, 3, \ldots, 400)$ can be calculated as.

$v_{1i} = count(AA)/length(P_i), v_{2i} = count(AC)/length(P_i), . v_{400i} = count(2-mer)/length(P_i)$.

where $count(AA)$, $count(AC)$ represents number of times "AA", "AC" has occurred in a protein sequence P_i, respectively. The length of the protein sequence varies from 30 to 30,000. The length of the protein sequence can highly impact the count values of the amino acids, impacting the count of k-mers. Long protein sequences tend to have more amino acid count. Hence, we normalize the frequencies by protein sequence length ($length(P_i)$) to remove this dependency. At this point, we have represented each protein sequence into a feature vector of size 400. The whole dataset can be represented by a matrix (M) of size n * 400, where n is the total number of protein samples, and 400 is the feature size.

Now, we reduce these features by using MLDA Wang et al. (2010) to get a more compact representation of the input data. MLDA gives at max $C-1$ features where C is the number of classes. MLDA is an extension of Linear Discriminant Analysis (LDA) to incorporate the multi-label classification. PFP is a multi-label classification problem as a protein can perform many functions and hence is a good candidate for using MLDA. After this step, we get a matrix of size n * f, where n is the number of protein samples, and f is the number of reduced features. In Fig. 1, the input block describes the process of preparing the input data from raw sequences to a reduced feature vector, which can be used to train our model.

Using multi-label linear discriminant analysis for dimensionality reduction

MLDA is considered as a generalization of classical LDA Wang et al. (2017). In classical LDA, the number of total positive classes is equal to the number of samples as each sample can belong to a single class in a traditional classification problem. However, in case of multi-label classification where a sample can belong to multiple classes at a time (Refer Fig. 2), the analogy that the number of positive classes is equal to the number of samples does not hold Zhang and Zhou (2013), ElKafrawy et al. (2015), Tsoumakas and Katakis (2007). Hence, the way between class matrices and within-class matrices gets populated, changes for multi-label classification problem. Eq. 1 formulates the multi-label between class matrix as

$$S_b = \sum_{k=1}^{K}((\sum_{i=1}^{n} Y_{ik})(\mu_k - \mu)(\mu_k - \mu)^T) \tag{1}$$

[3] Downloaded from https://github.com/bio-ontology-research-group/deepgo .

[4] Downloaded from https://www.biofunctionprediction.org/cafa/.

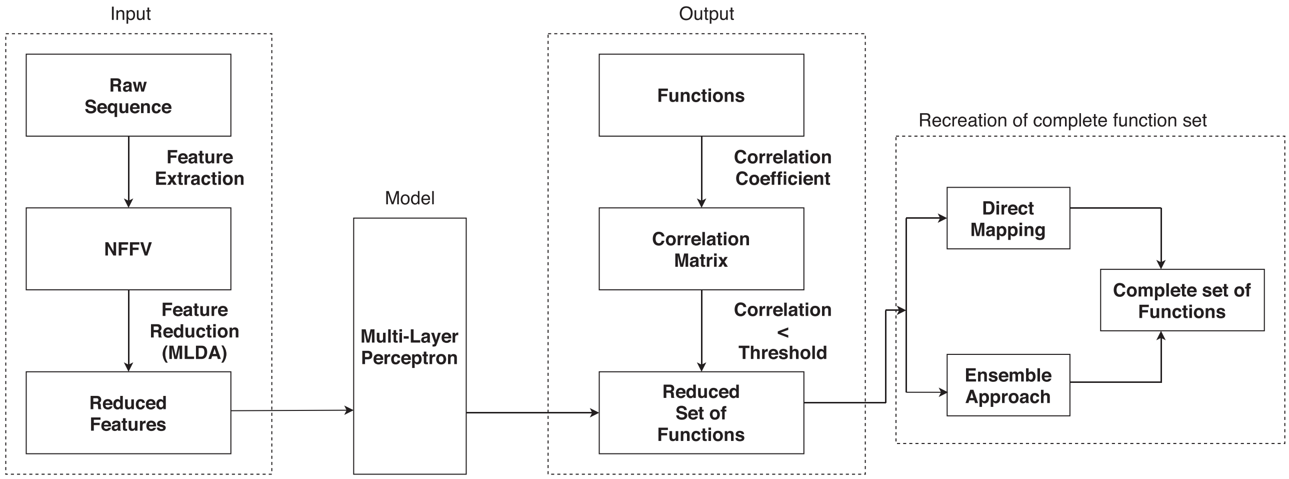


Fig. 1. Block diagram of the proposed system.

Table 1
Data statistics.

Dataset	Protein Functions	# Training Sample	# Testing Sample	# of GO terms
DeepGO	Biological Processes	36380	9096	932
DeepGO	Molecular Functions	25224	6306	589
CAFA3	Biological Processes	53498	2392	3352
CAFA3	Molecular Functions	36064	1132	617

and Eq. 2 defines the within class scatter matrix

$$S_w = \sum_{k=1}^{K}\sum_{i=1}^{n} Y_{ik}(x_i - \mu_k)(x_i - \mu_k)^T \tag{2}$$

where n is the number of data points in each class k, K is the total number of classes, μ_k is class mean for k^{th} class and μ is the global multi-label mean and are defined as:

$$\mu_k = \frac{\sum_{i=1}^{n} Y_{ik}x_i}{\sum_{i=1}^{n} Y_{ik}} \tag{3}$$

$$\mu = \frac{\sum_{k=1}^{K}\sum_{i=1}^{n} Y_{ik}x_i}{\sum_{k=1}^{K}\sum_{i=1}^{n} Y_{ik}} \tag{4}$$

Here, Y_{ik} represents the annotation of i^{th} protein for k^{th} function. Y_{ik} is 1, if the protein performs that function and 0 otherwise. x_i is the feature representation for each protein.

2.1.2. Output data preparation

As the discussed problem is a multi-label classification problem, we need to formulate protein functions into a multi-hot vector, similar to a one-hot vector. However, it can have more than one 1′s in a row vector. For each protein sample P_i, corresponding function annotations are converted into a multi-hot vector where ones represent performing functions, and zeros represent non-performing or missing functions.

Our data contain 932 terms in BP and 589 terms in MF. So, the vector size for BP and MF is 932 and 589, respectively. We use two statistical approaches to remove the redundant functions based on their correlation with each other. The choice of these two coefficients is based on the way they calculate the degree of similarity. Pearson's correlation coefficient compares the functions based on both the positivity and negativity of the functions, i.e., it considers the functions which always perform together as well as which never performs together as similar. While in the case of the Jaccard similarity coefficient, it considers only the functions which perform together as similar and not otherwise.

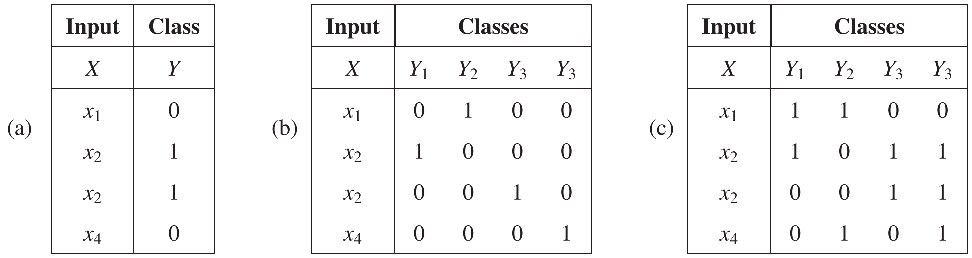
Pearson's correlation coefficient: We calculate Pearson's correlation coefficient (r) as given in Eq. 5 for each pair of functions and creates a correlation matrix.

$$r_{ij} = \frac{\left(\sum_{k=1}^{n} x_k \sum_{k=1}^{n} y_k\right) - n\sum_{k=1}^{n} x_k y_k}{\sqrt{\left[n\sum_{k=1}^{n} x_k{}^2 - \left(\sum_{k=1}^{n} x_k\right)^2\right]\left[n\sum_{k=1}^{n} y_k{}^2 - \left(\sum_{k=1}^{n} y_k\right)^2\right]}} \tag{5}$$

where:

a. r_{ij} is the Pearson's correlation coefficient between function f_i, f_j
b. n is the number of samples
c. x_k is value of sample for function f_i

$$x_k = \left\{\begin{array}{ll} 1, & \text{if the sample performs} f_i \\ 0, & \text{otherwise} \end{array}\right\}$$



(a)

Input	Class
X	Y
x_1	0
x_2	1
x_2	1
x_4	0

(b)

Input	Classes			
X	Y_1	Y_2	Y_3	Y_3
x_1	0	1	0	0
x_2	1	0	0	0
x_2	0	0	1	0
x_4	0	0	0	1

(c)

Input	Classes			
X	Y_1	Y_2	Y_3	Y_3
x_1	1	1	0	0
x_2	1	0	1	1
x_2	0	0	1	1
x_4	0	1	0	1

Fig. 2. Examples of different classification problems. (a) A single class classification problem. A data point X either belongs to a class Y or does not. (b) A multi-class classification problem where a data point X belongs to a single class out of many classes. (c) A multi-label classification problem where a data point X belongs to more than a class in the set of target classes.

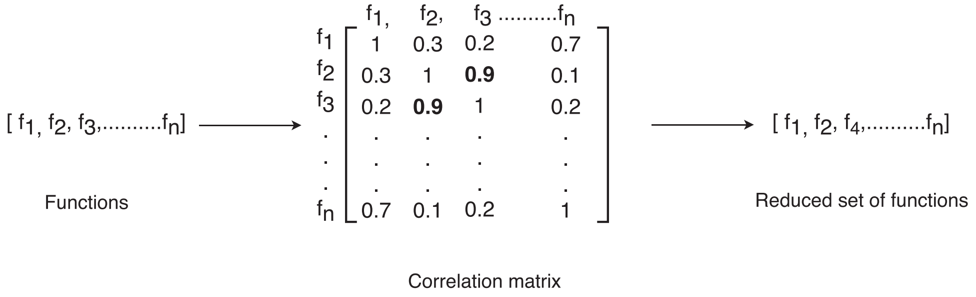


Fig. 3. Example of output data preparation.

d. y_k is value of sample for function f_j

$$y_k = \left\{ \begin{array}{ll} 1, & \text{if the sample performs} f\_j \\ 0, & \text{otherwise} \end{array} \right\}$$

Jaccard similarity coefficient: The Jaccard similarity coefficient is a measure to find the extent of overlap between two sets. Jaccard similarity coefficient can be calculated as per the below formula for each pair of protein functions.

$$J = \frac{C_{11}}{C_{00} + C_{01} + C_{10}} \tag{6}$$

where:

A. J is the jaccard similarity coefficient for function f_i and f_j
B. C_{11} represents the total number of samples where f_i and f_j both have a value of 1
C. C_{00} represents the total number of samples where f_i and f_j both have a value of 0
D. C_{01} represents the total number of samples where f_i is 0 and f_j has value 1
E. C_{10} represents the total number of samples where f_i is 1 and f_j has a value of 0

Correlation coefficient values towards 1 indicate that the two functions are highly correlated, while the value towards 0 indicates that the two functions are completely unrelated. We set a threshold value empirically to decide on the optimum value for two functions being correlated. Fig. 1, output block describes this process of preparing the output data, i.e., creating the non-redundant set of functions. Fig. 3 shows an example where we have an initial set of functions $f_1, f_2, f_3, \ldots, f_n$. We generate a correlation matrix by calculating the correlation coefficients using Pearson's correlation coefficient and Jaccard similarity coefficient separately for every pair of functions. We see from the correlation matrix that functions f_2 and f_3 are highly correlated with a correlation coefficient of 0.9, so we remove one of these functions and creates a new set of less redundant functions which is later used as a target output to train our model.

Table 2
Overview of the proposed work.

Input:
- Reduced NFFV features constructed from protein sequences (section 2.1.1)
- Reduced set of non redundant functions (section 2.1.2)

Output:
- Predicted functions for unannotated protein sequences

Predicting protein function with proposed methodology
- Compute NFFV features for each protein sequence
- Create reduced NFFV features using MLDA
- Create set of non redundant functions using two different statistical approaches
  (1) Pearson's correlation coefficient
  (2) Jaccard similarity coefficient
- Train a MLP network to predict the non redundant set of functions
- Compute the prediction probability of redundant functions using two approaches:
  (1) Direct Mapping (section 2.3)
  (2) Ensemble approach (section 2.4)

2.2. Proposed model

We propose a multi-layer perceptron network trained with protein sequences as input and a reduced set of non-redundant functions as output. Protein sequences are represented as a reduced set of NFFV features (refer section 2.1.1), and a non-redundant set of function was created out of a redundant set of functions (refer section 2.1.2). Though the model is trained to predict only the non-redundant function, we evaluate our model with a complete set of functions. For which we need to recreate the redundant set of functions for test data. We have used two approaches to do the same. First, the redundant function gets direct mapping from its counter similar function. Second is an ensemble approach in which another multi-layer perceptron network is used to train the leftover set of redundant functions. The results of these two MLP are concatenated together and fed to another MLP to get the final output. Table 2 provides the overview of the proposed system.

The MLP has a single hidden layer and is trained for 500 epochs with a batch size of 32, loss function as binary cross-entropy, and rmsprop optimizer Igel and Hüsken (2000)). The number of neurons in the hidden layer is between the input feature and the number of output functions. The activation function used at the hidden layer is ReLU, and at the output layer is the sigmoid function.

2.3. Direct mapping

Direct mapping is the simplest approach to populate the prediction probability of redundant functions. In this case, the prediction value for redundant functions is populated directly by the values of similar functions on which the model was trained. All the functions which are similar to each other will have the same prediction probability for a protein. For example, if there are n functions in the complete dataset, and after removing the set's similar functions, only k where $k < n$ functions are left. The model will then be trained on k functions and will give a prediction for k functions only. To evaluate the model's performance completely, we need to populate the rest of the $n - k$ function prediction values. To do so, we use the prediction values of k functions. If function f_1 is similar to f_2 and the model was trained on f_2 and gave the prediction probability of f_2 for a protein as 0.24, then for function f_1, the same value 0.24 will be taken.

2.4. Ensemble approach

In the second approach, we automate the process of mapping the prediction probabilities. For this, we use an ensemble approach in which two MLPs are trained, one with the non-redundant set of functions (k) and another with all the leftover functions ($n - k$). The resultant probabilities for both the MLPs are concatenated together to train the third MLP, which takes the concatenated resultant probabilities of the previously trained MLPs as input to compute the complete set of functions. The third MLP is used to give a distinction between the prediction probabilities of similar functions. Fig. 4 explains this approach in detail.

2.5. Incorporating protein Interaction data

We also tested the proposed model with protein interaction data available in DeepGO dataset along with the protein sequences. The interaction data constitutes knowledge embeddings of size 256 for each protein. These interaction features are part of the original dataset Kulmanov et al. (2017). We concatenate these features with the feature vector generated from sequences and create a new multi-modal vector representation of proteins. We use the proposed model on this new set of features to check the model's generalization on other feature sets. We see that the inclusion of functional inter-relationship improves the performance with this feature set.

2.6. Evaluation metrics

We evaluate our model's performance with protein-centric measures F_{max}, average precision, and average recall. We calculate these metrics for a threshold $t \in [0,1]$. We have used the adaptive threshold for various models used in this work (Refer Section 2.7 for details). In addition to these, we also report area under the precision-recall curve (AUPR), which is a measure for highly class imbalanced dataset Davis and Goadrich (2006).

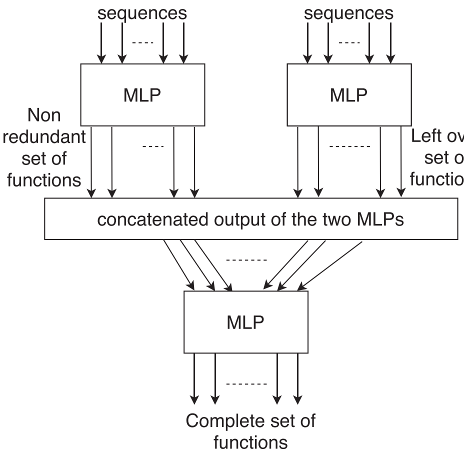
Average Precision: Average precision is a measure of correct predictions among all the actual predictions, averaged over all the samples. In protein function prediction, precision is calculated only in terms of performing functions and does not consider non-performing functions as the count of non-performing functions is huge. We are mostly interested in knowing the functions performed by a protein correctly, rather than knowing about the functions that protein does not perform. Equation 7 computes average precision in our work.

$$avgPrecision = \frac{1}{n} * \sum_{i=1}^{n} \frac{Pred_i \cap True_i}{Pred_i} \quad (7)$$

Average Recall: Average recall is a measure of correct predictions among all the actual labels, averaged over all the samples. Equation 8 computes average recall in our work.

$$avgRecall = \frac{1}{n} * \sum_{i=1}^{n} \frac{Pred_i \cap True_i}{True_i} \quad (8)$$

F_{max}: F_{max} is the maximum protein-centric F1-score calculated over all thresholds. F1 score is the harmonic mean between Precision and Recall. Equation 9 shows the mathematical representation of F_{max}.


sequences
----
sequences

MLP
MLP
Non redundant set of functions

Left over set of functions
concatenated output of the two MLPs

MLP

Complete set of functions

Fig. 4. Flow diagram for ensemble approach.

$$F_{max} = \max_{t} \left(\frac{2 * avgPrecision * avgRecall}{(avgPrecision + avgRecall)} \right) \quad (9)$$

where:

$Pred_i$ is the set of positive prediction for protein sample P_i
$True_i$is the set of true annotations for protein sample P_i
n is the total number of samples
t is the thresholds

2.7. Estimation of threshold

We have proposed an adaptive threshold estimation while cross-validating our models with a 5-fold cross-validation technique. As discussed earlier, we have separate training and testing sets in the dataset. We cross-validate our model on the training set by splitting the training data into 5 parts. It is assumed that 4 parts are annotated and rest of 1 part is unannotated. We run our model to predict the functions of unannotated data. To predict the functions as performing and non-performing, we evaluate the model in terms of evaluation metrics for various threshold values $t \in [0,1]$. If the prediction probability of a function is greater than the threshold, the corresponding function will be considered as performing a function and vice-versa. The prediction results are evaluated in terms of the evaluation metrics defined in section 2.6. The threshold for which we get maximum F1 score and high precision is noted. This step is repeated 5 times by keeping each part unannotated in turn as we have used 5 fold cross-validation. We calculate mean, median, and mode for the thresholds found in the 5 folds to get the final thresholds. The mean, median, mode prediction probability threshold value is used further to test the performance of our model on test data. Table 3 shows the final threshold values achieved and we see that the threshold estimation in our case is robust across the folds and hence the variation between mean, median, and mode is negligible, showing the generalization of the models trained. We later use mean threshold as the actual threshold value to test our models.

3. Results and discussion

A machine learning model is built to predict huge set of functions that utilizes the inter-relationships between functions to improve the model's predictability. The model is trained on reduced set of non-redundant functions to minimize the ambiguity created due to correlated functions. The reduced set of functions is created using different similarity measures to quantify the correlation between functions. The model is evaluated on unseen data for the complete set of functions created using Direct mapping and ensemble approach.

We compare the proposed approach with two well-performing existing approaches 1) DeepGO Kulmanov et al. (2017) 2) MLDA Wang et al. (2017). Compared with MLDA, we have used the features generated using the MLDA approach as an input to the multi-layer perceptron network. We have used the same features ahead in the proposed approach. We have evaluated the proposed model on two correlation parameters 1) Pearson's correlation coefficient and 2) the Jaccard similarity coefficient. Table 4 and Table 5 summarizes the results on the DeepGO dataset, while Table 6 and Table 7 summarizes the results on the CAFA3 dataset for both the parameters, respectively.

The results are analyzed through three different perspectives. Firstly, the overall performance of the proposed model when compared to other approaches. In the performance tests, it is found that the proposed approach performed better than the baseline approach MLDA for both biological process and molecular functions on both the datasets. It produced similar results with the dataset used in DeepGO for molecular functions compared to the DeepGO model, which used a hierarchical

Table 3
Mean, median, mode threshold values corresponding to biological processes and molecular functions across different methods on DeepGO dataset.

Degree of correlation	Methods	Biological Processes			Molecular Functions		
		Mean	Median	Mode	Mean	Median	Mode
Pearson's correlation coefficient	MLDA (seq+MLP)	0.184	0.18	0.18	0.18	0.18	0.18
	seq+MLP+Ensemble	0.216	0.22	0.22	0.244	0.24	0.24
	MLDA (seq+interaction+MLP)	0.244	0.24	0.24	0.228	0.22	0.22
	seq+interaction+MLP+Ensemble	0.244	0.24	0.24	0.244	0.24	0.24
Jaccard similarity coefficient	MLDA (seq+MLP)	0.184	0.18	0.18	0.18	0.18	0.18
	seq+MLP+Ensemble	0.216	0.22	0.22	0.232	0.22	0.22
	MLDA (seq+interaction+MLP)	0.244	0.24	0.24	0.228	0.22	0.22
	seq+interaction+MLP+Ensemble	0.244	0.24	0.24	0.236	0.24	0.24

Table 4
F_{max}, average precision, average recall of various compared approaches (DeepGOseq, MLDA (seq+MLP), DeepGO, MLDA (seq+interaction+MLP)) and proposed approaches (seq+MLP+DirectMap, seq+MLP+Ensemble, seq+interaction+MLP+DirectMap, seq+interaction+MLP+ensemble) for biological processes and molecular functions using pearson's correlation coefficient as degree of similarity on DeepGO dataset.

Methods	Pearson's correlation coefficient							
	Biological Process				Molecular Function			
	F_{max}	Avg Precision	Avg Recall	AUPR	F_{max}	Avg Precision	Avg Recall	AUPR
DeepGOseq	0.293	0.304	0.282	–	0.364	0.453	0.304	–
MLDA (seq+MLP)	0.339	0.3332	0.3455	0.26	0.3591	0.365	0.3541	0.25
seq+MLP+DirectMap	0.3463	0.3573	0.336	0.26	0.3657	0.3688	0.3626	0.25
seq+MLP+Ensemble	0.3519	0.3656	0.3392	0.27	0.3843	0.444	0.3387	0.3
DeepGO	0.395	0.412	0.379	–	**0.47**	**0.577**	0.397	–
MLDA (seq+interaction+MLP)	0.4158	0.4074	**0.4251**	0.38	0.4312	0.4693	0.399	0.35
seq+interaction+MLP+DirectMap	0.3953	0.3853	0.40577	0.39	0.4376	0.4694	0.4099	0.35
seq+interaction+MLP+Ensemble	**0.4245**	**0.4288**	0.4203	0.4	0.462	0.5203	**0.4154**	0.41

Table 5
F_{max}, average precision, average recall of various compared approaches (DeepGOseq, MLDA (seq+MLP), DeepGO, MLDA (seq+interaction+MLP)) and proposed approaches (seq+MLP+DirectMap, seq+MLP+Ensemble, seq+interaction+MLP+DirectMap, seq+interaction+MLP+ensemble) for biological processes and molecular functions using jaccard similarity coefficient as degree of similarity on DeepGO dataset.

Methods	Jaccard Similarity coefficient							
	Biological Process				Molecular Function			
	F_{max}	Avg Precision	Avg Recall	AUPR	F_{max}	Avg Precision	Avg Recall	AUPR
DeepGOseq	0.293	0.304	0.282	–	0.364	0.453	0.304	–
MLDA (seq+MLP)	0.339	0.3332	0.3455	0.26	0.3591	0.365	0.3541	0.25
seq+MLP+DirectMap	0.3462	0.3244	0.3711	0.27	0.3816	0.4015	0.3636	0.29
seq+MLP+Ensemble	0.35	0.3623	0.3386	0.27	0.3848	0.436	0.3444	0.3
DeepGO	0.395	0.412	0.379	–	**0.47**	**0.577**	0.397	–
MLDA (seq+interaction+MLP)	0.4158	0.4074	**0.4251**	0.38	0.4312	0.4693	0.399	0.35
seq+interaction+MLP+DirectMap	0.415	0.4084	0.4219	0.39	0.4484	0.4556	**0.4413**	0.40
seq+interaction+MLP+Ensemble	**0.4254**	**0.4312**	0.4198	0.40	0.4608	0.5111	0.4196	0.41

Table 6
F_{max}, average precision, average recall, AUPR of various compared approaches (MLDA (seq+MLP), DeepGO)) and proposed approaches (seq+MLP+DirectMap, seq+MLP+Ensemble) for biological processes and molecular functions using pearson's correlation coefficient as degree of similarity on CAFA3 dataset.

Methods	Pearson's correlation coefficient							
	Biological Process				Molecular Function			
	F_{max}	Avg Precision	Avg Recall	AUPR	F_{max}	Avg Precision	Avg Recall	AUPR
MLDA (seq+MLP)	0.355	0.3652	0.3465	0.25	0.2275	0.2178	0.2383	0.18
seq+MLP+DirectMap	0.4011	0.3971	0.4051	0.27	0.2672	0.2847	0.2757	0.22
seq+MLP+Ensemble	0.4043	0.4124	0.3965	0.28	0.2831	0.2753	0.2913	0.23
DeepGO	0.435	–	–	0.385	0.393	–	–	0.303

classification layer. We investigated the applicability of incorporating a purely statistical approach of defining inter-functional relationships in an MLP instead of focusing on optimizing the model for some specific function families. The model failed to perform well on the CAFA3 dataset compared to DeepGO. The reason being the low performance of the baseline model, and hence the inclusion of functional inter-relationship could not lift the performance well above DeepGO. According to our observations, it is feasible to use a simple machine learning model like MLP to perform better when strengthened by the inclusion of functional similarity measures.

Secondly, the performance of the proposed approach is analyzed with variations in data and protein features. From Table 4 and Table 5, it can be seen that a multi-modal approach in which multiple types of protein features (like protein interaction data along with protein

Table 7
F_{max}, average precision, average recall, AUPR of various compared approaches (MLDA (seq+MLP), DeepGO)) and proposed approaches (seq+MLP+DirectMap, seq+MLP+Ensemble) for biological processes and molecular functions using jaccard similarity coefficient as degree of similarity on CAFA3 dataset.

Methods	Jaccard Similarity coefficient							
	Biological Process				Molecular Function			
	F_{max}	Avg Precision	Avg Recall	AUPR	F_{max}	Avg Precision	Avg Recall	AUPR
MLDA (seq+MLP)	0.355	0.3652	0.3465	0.25	0.2275	0.2178	0.2383	0.18
seq+MLP+DirectMap	0.4032	0.4329	0.3774	0.26	0.2803	0.2705	0.2753	0.23
seq+MLP+Ensemble	0.41	0.407	0.428	0.28	0.2874	0.2887	0.2861	0.24
DeepGO	0.435	–	–	0.385	0.393	–	–	0.303

sequences) are used at once performs better compared to the methods that use only protein sequences. However, the amount of such data characterizations is still an issue as compared to the number of protein sequences. The machine learning approaches suffer from this challenge as it requires a high amount of data for training. From Table 6 and Table 7, it can be seen that with the different datasets and different feature representations, the inclusion of functional inter-relationship increases the performance of the model promisingly.

Thirdly, the performance of the proposed approach in terms of the functional similarity measures being used. In the problem of protein function prediction, we are mainly concerned about the functions performed by a protein rather than the functions that are not being performed. Hence, the Jaccard similarity coefficient seemed to be a better alternative for the problem at hand. For the same reason, it can be inferred that the Jaccard similarity coefficient performed better in the case of Direct Mapping when the information of leftover redundant function set did not play any role.

3.1. Prediction of more specific protein functions

The proposed work is analyzed in term-centric measures and found that it can predict most specific functions too, along with the general functions. Table 8 shows the term centric precision, recall, and F_{max} for some of the specific protein functions which were unpredictable by MLDA and DeepGO. The results are the outcome of the process in which protein sequences and interaction data are used to predict the protein functions using an ensemble approach with Pearson's correlation coefficient. These results show the better learning capability of the proposed model for specific functions.

3.2. Correlation threshold value's relationship with function predictability

The threshold value for correlation depicts the degree of correlation which shows the extent of redundancy in GO terms. The proposed model is evaluated at various threshold values for correlation to check the performance of the model. It is found that for Pearson's correlation coefficient if the threshold value is taken as 0.7 for two functions being correlated, our model achieves the highest performance. Hence, we proceed further with this threshold value and formulates the results for Direct mapping and ensemble MLP. Fig. 5 shows the change in F_{max} with the change in the threshold of Pearson's correlation coefficient. F_{max} (reduced) is the F_{max} achieved by removing the redundant GO terms from the dataset to train and test the model. F_{max} (expanded) is the F_{max} achieved by recreating the original functions set on test data using the Direct Mapping approach, which as previously explained in section 2.3. From Fig. 5 it can be seen that F_{max} (reduced) has a direct relationship with decreasing threshold, while F_{max} (expanded) shows a slight increase till a threshold value after which it decreases drastically. The reason behind the increase in F_{max} (reduced) with decreasing threshold is the number of protein functions that get predicted. When the correlation threshold value is decreased, more functions become correlated to each other, increasing the number of redundant functions. When the redundant functions are removed from the dataset, the data becomes thinner in terms of output, making the model training better and hence the better F_{max}. However, in the case of F_{max} (expanded), it shows a slight increase till a threshold point and decreases heavily later. As we decrease the correlation threshold value, we decrease the degree of similarity between the functions. For example, two functions f_1 and f_2 with correlation coefficient as 0.7 is still similar to each other to a good extent, while functions f_3 and f_4 with correlation coefficient as 0.4 are not very similar functions. So, if we take f_1 and f_2 as redundant functions, the model performs better, but if we take f_3 and f_4 as redundant functions, the model's performance is hampered.

Fig. 6 explains the behavior of the F_{max} with the change in threshold value in the case of the Jaccard similarity coefficient. The threshold value for correlation is 0.8 for biological processes and 0.7 for molecular functions. At these threshold values, the proposed model gave the highest performance.

Table 8
Term centric metrics for few specific functions using proposed technique which were not predictable using MLDA and DeepGO.

Biological processes				Molecular Functions			
Functions	precision	recall	F_{max}	Functions	precision	recall	F_{max}
GO:0051259	0.33	0.01	0.02	GO:0043021	0.17	0.03	0.05
GO:0022411	0.25	0.06	0.09	GO:0008017	0.27	0.07	0.11
GO:0051346	0.5	0.01	0.03	GO:0051015	0.17	0.03	0.05
GO:0042391	0.17	0.09	0.12	GO:0000979	0.2	0.08	0.12
GO:0045861	1	0.03	0.05	GO:0035326	0.22	0.07	0.1
GO:0006275	0.5	0.02	0.04	GO:0001046	0.2	0.08	0.11
GO:0070372	0.14	0.03	0.05	GO:0001047	0.17	0.08	0.11
GO:0032102	1	0.01	0.02	GO:0003730	0.19	0.16	0.17
GO:0090316	0.25	0.03	0.05	GO:0003729	0.21	0.15	0.18
GO:0051222	0.12	0.04	0.06	GO:0003727	0.05	0.06	0.05
GO:1903829	0.18	0.03	0.05	GO:0003725	0.14	0.06	0.08
GO:0010976	0.14	0.02	0.04	GO:0050661	1	0.09	0.17
GO:0045666	0.17	0.03	0.05	GO:0003697	0.5	0.06	0.11
GO:1903651	0.36	0.05	0.09	GO:0005524	0.28	0.11	0.15
GO:0032388	0.17	0.04	0.07	GO:0051287	1	0.06	0.12

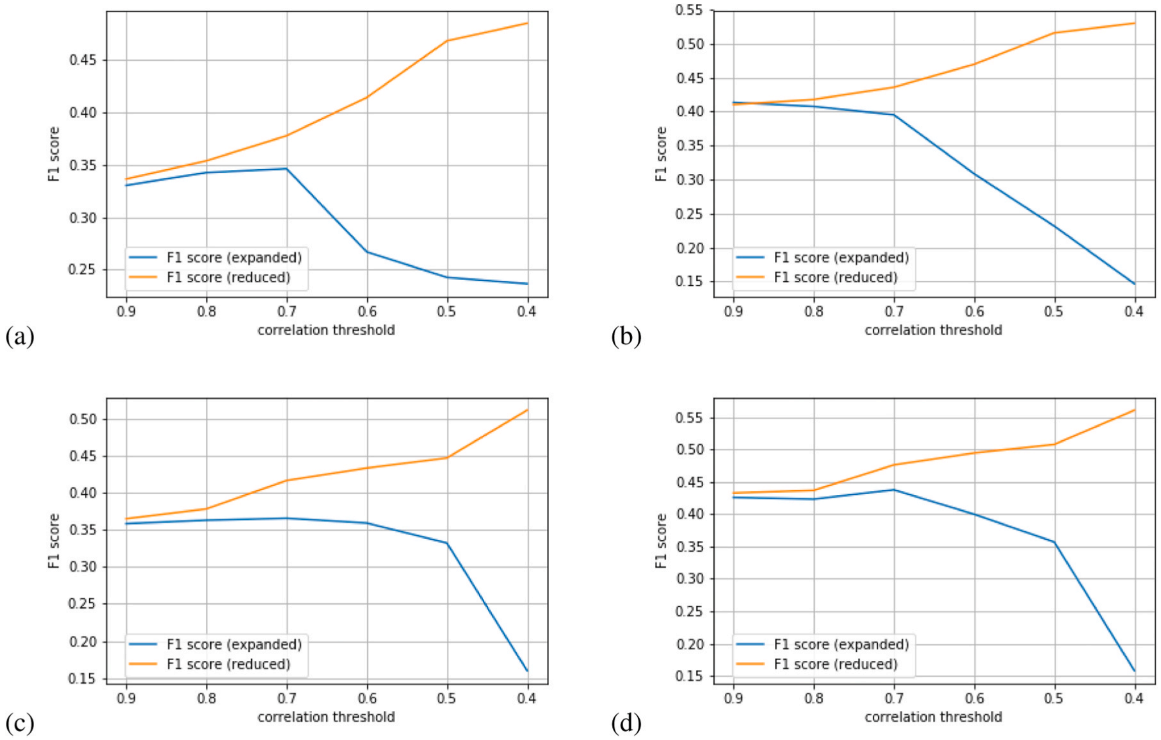


Fig. 5. Plots showing the behaviour of F_{max} with the change in threshold for Pearson's correlation coefficient (a) behaviour of F_{max} in case of biological processes with sequence data (b) behaviour of F_{max} in case of biological processes with sequence and interaction data (c) behaviour of F_{max} in case of molecular function with sequence data (d) behaviour of F_{max} in case of molecular functions with sequence and interaction data.

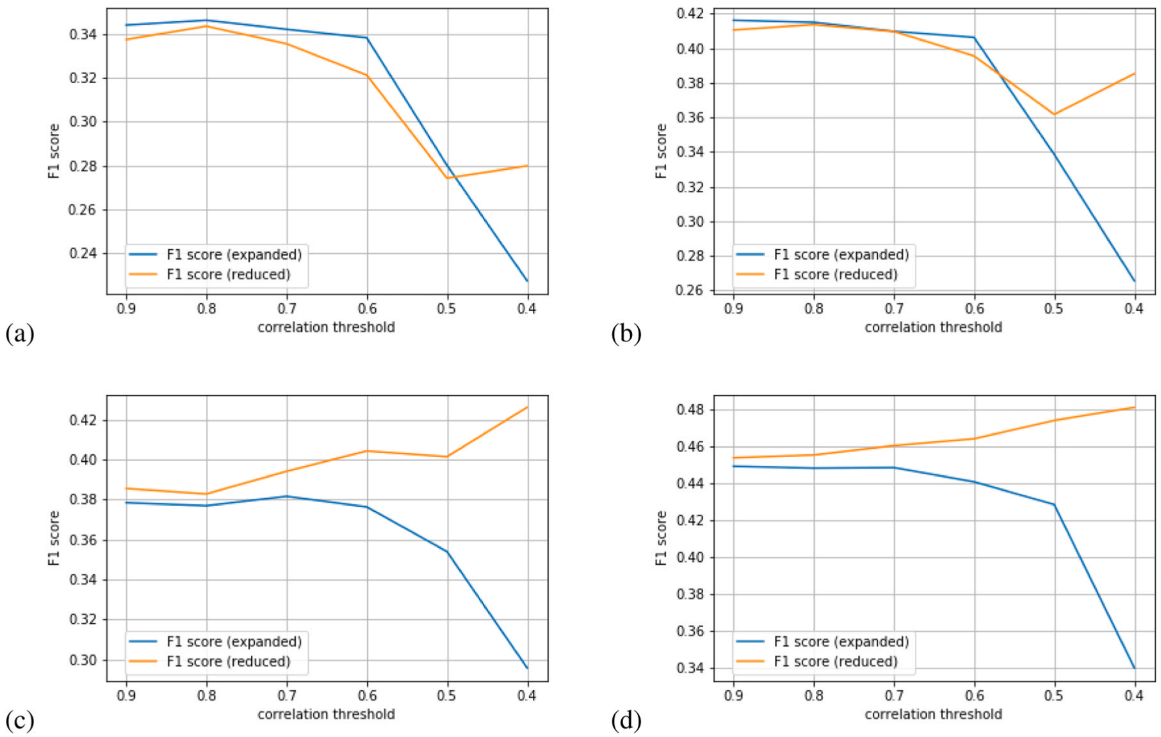


Fig. 6. Plots showing the behaviour of F_{max} with the change in threshold for Jaccard similarity coefficient (a) behaviour of F_{max} in case of biological processes with sequence data (b) behaviour of F_{max} in case of biological processes with sequence and interaction data (c) behaviour of F_{max} in case of molecular function with sequence data (d) behaviour of F_{max} in case of molecular functions with sequence and interaction data.

4. Conclusion and future works

In this paper, we proposed an approach that incorporates functional inter-relationship as a tool to improve the machine learning model's performance. In protein function prediction, the number of functions to be predicted is relatively high, which hinders machine learning models' efficiency. The functions to be predicted are similar to each other, which adds ambiguity while learning to the already complex model. We proposed approaches to remove the redundant or similar functions, and also proposed the way by which we can get the predictions for removed redundant functions. For removing the redundant functions, we used Pearson's correlation coefficient and Jaccard similarity coefficient. For recreating the functions, we used two approaches 1) Direct Mapping 2) Ensemble approach to help us make a fair comparison with other known approaches as a model generally performs well with fewer functions. To prove the proposed approaches' applicability, we used two datasets with different sets of protein functions (biological processes and molecular functions) along with two different representations of proteins (sequences and protein interaction graph). The proposed approach gave promising results for all the separate sets of data proving the models' reliability and applicability.

We observed that the model performs better when protein interaction data is combined with the sequences, which increases the overall performance of protein function prediction. There are various other types of data like protein structure, microarray expression data, etc., which can further improve protein function prediction performance. We have used Pearson's correlation coefficient and Jaccard similarity coefficient to measure the degree of similarity between functions. We saw that the Jaccard similarity coefficient worked better than Pearson's correlation coefficient for direct mapping. There are many other similarity measures like cosine similarity, edit distance, etc., which can be used to measure the degree of similarity between functions by analyzing its usability in the case of protein functions.

Data statement

All data and code is available at https://github.com/richadhanuka/PFP-using-Functional-interrelationship.

Declaration of Competing Interest

None.

References

K.S. Ahmed, N.H. Soloma, Y.M. Kadah, Exploring protein functions correlation based on overlapping proteins and cluster interactions, in: 2011 1st Middle East Conference on Biomedical Engineering, IEEE, 2011, 247–251.

Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J., 1990. Basic local alignment search tool. J. Mol. Biol. 215 (3), 403–410.

Altschul, S.F., Madden, T.L., Schäffer, A.A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.J., 1997. Gapped blast and psi-blast: a new generation of protein database search programs. Nucleic Acids Res. 25 (17), 3389–3402.

Asgari, E., Mofrad, M.R., 2015. Continuous distributed representation of biological sequences for deep proteomics and genomics. PLoS One 10 (11), e0141287.

Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., et al., 2000. Gene ontology: tool for the unification of biology. Nat. Genet. 25 (1), 25.

Barutcuoglu, Z., Schapire, R.E., Troyanskaya, O.G., 2006. Hierarchical multi-label prediction of gene function. Bioinformatics 22 (7), 830–836.

Boutet, E., Lieberherr, D., Tognolli, M., Schneider, M., Bansal, P., Bridge, A.J., Poux, S., Bougueleret, L., Xenarios, I., 2016. Uniprotkb/swiss-prot, the manually annotated section of the uniprot knowledgebase: how to use the entry view. In: Plant Bioinformatics. Springer, pp. 23–54.

Cao, R., Freitas, C., Chan, L., Sun, M., Jiang, H., Chen, Z., 2017. Prolango: protein function prediction using neural machine translation based on a recurrent neural network. Molecules 22 (10), 1732.

Carbon, S., Ireland, A., Mungall, C.J., Shu, S., Marshall, B., Lewis, S., Hub, A., W.P.W. Group, 2008. Amigo: online access to ontology and annotation data. Bioinformatics 25 (2), 288–289.

Chen, J., Guo, M., Wang, X., Liu, B., 2016. A comprehensive review and comparison of different computational methods for protein remote homology detection. Brief. Bioinformatics 19 (2), 231–244.

D. Chicco, P. Sadowski, P. Baldi, Deep autoencoder neural networks for gene ontology annotation predictions, in: Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics, ACM, 2014, 533–540.

Consortium, G.O., 2018. The gene ontology resource: 20 years and still going strong. Nucleic Acids Res. 47 (D1), D330–D338.

Cozzetto, D., Minneci, F., Currant, H., Jones, D.T., 2016. Ffpred 3: feature-based function prediction for all gene ontology domains. Sci. Rep. 6 (1), 1–11.

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

R. Eisner, B. Poulin, D. Szafron, P. Lu, R. Greiner, Improving protein function prediction using the hierarchical structure of the gene ontology, in: 2005 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational biology, IEEE, 2005, 1–10.

ElKafrawy, P., Mausad, A., Esmail, H., 2015. Experimental comparison of methods for multi-label classification in different application domains. Int. J. Comput. Appl. 114 (19), 1–9.

Godzik, A., Jambon, M., Friedberg, I., 2007. Computational protein function prediction: are we making progress? Cell. Mol. Life Sci. 64 (19–20), 2505.

Y. Goldberg, O. Levy, word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method, arXiv preprint arXiv:1402.3722 2021.

Hill, D.P., Smith, B., McAndrews-Hill, M.S., Blake, J.A., 2008. Gene ontology annotations: what they mean and where they come from (BioMed Central). BMC Bioinformatics 9, 1–9.

Hou, J., Cao, R., Cheng, J., 2019a. Deep convolutional neural networks for predicting the quality of single protein structural models. bioRxiv, 590620.

Hou, J., Guo, Z., Cheng, J., 2019b. Dnss2: improved ab initio protein secondary structure prediction using advanced deep learning architectures. bioRxiv, 639021.

C. Igel, M. Hüsken, Improving the rprop learning algorithm, in: Proceedings of the Second International ICSC Symposium on Neural Computation (NC 2000), 2000, Citeseer, 2000, 115–121.

Jing, X., Dong, Q., Hong, D., Lu, R., 2021. Amino acid encoding methods for protein sequences: a comprehensive review and assessment. IEEE/ACM Trans. Comput. Biol. Bioinformatics.

Kulmanov, M., Hoehndorf, R., 2020. Deepgoplus: improved protein function prediction from sequence. Bioinformatics 36 (2), 422–429.

Kulmanov, M., Khan, M.A., Hoehndorf, R., 2017. Deepgo: predicting protein functions from sequence and interactions using a deep ontology-aware classifier. Bioinformatics 34 (4), 660–668.

Li, W., Ma, B., Zhang, K., 2014. Optimizing spaced *k*-mer neighbors for efficient filtration in protein similarity search. IEEE/ACM Trans. Comput. Biol. Bioinformatics 11 (2), 398–406.

Makrodimitris, S., van Ham, R.C., Reinders, M.J., 2019. Improving protein function prediction using protein sequence and go-term similarities. Bioinformatics 35 (7), 1116–1124.

M. Masseroli, D. Chicco, P. Pinoli, Probabilistic latent semantic analysis for prediction of gene ontology annotations, in: The 2012 international joint conference on neural networks (IJCNN), IEEE, 2012, 1–8.

Meng, J., Wekesa, J.-S., Shi, G.-L., Luan, Y.-S., 2016. Protein function prediction based on data fusion and functional interrelationship. Math. Biosci. 274, 25–32.

L.J. Miranda, J. Hu, A deep learning approach based on stacked denoising autoencoders for protein function prediction, in: 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), 1, IEEE, 2018, 480–485.

G. Pandey, V. Kumar, M. Steinbach, Computational approaches for protein function prediction: A survey, Tech. Rep. 06–028, Twin Cities: Department of Computer Science and Engineering, University of Minnesota (2006).

Pandey, G., Myers, C.L., Kumar, V., 2009. Incorporating functional inter-relationships into protein function prediction algorithms. BMC Bioinformatics 10 (1), 142.

Rifaioglu, A.S., Doğan, T., Martin, M.J., Cetin-Atalay, R., Atalay, V., 2019. Deepred: automated protein function prediction with multi-task feed-forward deep neural networks. Sci. Rep. 9 (1), 1–16.

Rost, B., Liu, J., Nair, R., Wrzeszczynski, K.O., Ofran, Y., 2003. Automatic prediction of protein function. Cell. Mol. Life Sci. 60 (12), 2637–2650.

Sleator, R.D., Walsh, P., 2010. An overview of in silico protein function prediction. Arch. Microbiol. 192 (3), 151–155.

S.K. Sønderby, C.K. Sønderby, H. Nielsen, O. Winther Convolutional lstm networks for subcellular localization of proteins, in: International Conference on Algorithms for Computational Biology, Springer, 2015, 68–80.

Sun, P., Tan, X., Guo, S., Zhang, J., Sun, B., Du, N., Wang, H., Sun, H., 2018. Protein function prediction using function associations in protein-protein interaction network. IEEE Access 6, 30892–30902.

Tsoumakas, G., Katakis, I., 2007. Multi-label classification: an overview. Int. J. Data Warehouse Min. (IJDWM) 3 (3), 1–13.

H. Wang, C. Ding, H. Huang, Multi-label linear discriminant analysis, in: European Conference on Computer Vision, Springer, 2010, 126–139.

Wang, H., Yan, L., Huang, H., Ding, C., 2017. From protein sequence to protein function via multi-label linear discriminant analysis. IEEE/ACM Trans. Comput. Biol. Bioinformatics (TCBB) 14 (3), 503–513.

Wang, L., Wang, Y., Chang, Q., 2016. Feature selection methods for big data bioinformatics: a survey from the search perspective. Methods 111, 21–31.

Wang, Z., Zhao, C., Wang, Y., Sun, Z., Wang, N., 2018. Panda: protein function prediction using domain architecture and affinity propagation. Sci. Rep. 8 (1), 1–10.

You, R., Huang, X., Zhu, S., 2018. Deeptext2go: Improving large-scale protein function prediction with deep semantic text representation. Methods 145, 82–90.

You, R., Yao, S., Xiong, Y., Huang, X., Sun, F., Mamitsuka, H., Zhu, S., 2019. Netgo: improving large-scale protein function prediction with massive network information. Nucleic Acids Res. 47 (W1), W379–W387.

Yu, G., Zhao, Y., Lu, C., Wang, J., 2017. Hashgo: hashing gene ontology for protein function prediction. Comput. Biol. Chem. 71, 264–273.

Yu, H.-J., Huang, D.-S., 2013. Normalized feature vectors: a novel alignment-free sequence comparison method based on the numbers of adjacent amino acids. IEEE/ ACM Trans. Comput. Biol. Bioinformatics (TCBB) 10 (2), 457–467.

Zhang, M.-L., Zhou, Z.-H., 2013. A review on multi-label learning algorithms. IEEE Trans. Knowl. Data Eng. 26 (8), 1819–1837.

Zhang, Z., Zhao, Y., Liao, X., Shi, W., Li, K., Zou, Q., Peng, S., 2018. Deep learning in omics: a survey and guideline. Brief. Funct. Genom. 18 (1), 41–57.

Zhou, N., Jiang, Y., Bergquist, T.R., Lee, A.J., Kacsoh, B.Z., Crocker, A.W., Lewis, K.A., Georghiou, G., Nguyen, H.N., Hamid, M.N., et al., 2019. The cafa challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. Genome Biol. 20 (1), 1–23.