Sequence Analysis

HECNet: a hierarchical approach to Enzyme Function Classification using a Siamese Triplet Network

Safyan Aman Memon [1,†], Kinaan Aamir Khan [1,†] and Hammad Naveed [1,*]

[1] Computational Biology Research Lab, Department of Computer Science, National University of Computer and Emerging Sciences, Islamabad, Pakistan

* To whom correspondence should be addressed, † Equal Contribution

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Understanding an enzyme's function is one of the most crucial problem domains in computational biology. Enzymes are a key component in all organisms and many industrial processes as they help in fighting diseases and speed up essential chemical reactions. They have wide applications and therefore, the discovery of new enzymatic proteins can accelerate biological research and commercial productivity. Biological experiments, to determine an enzyme's function, are time-consuming and resource expensive.

Results: In this study, we propose a novel computational approach to predict an enzyme's function up to the fourth level of the Enzyme Commission (EC) Number. Many studies have attempted to predict an enzyme's function. Yet, no approach has properly tackled the fourth and final level of the EC number. The fourth level holds great significance as it gives us the most specific information of how an enzyme performs its function. Our method uses innovative deep learning approaches along with an efficient hierarchical classification scheme to predict an enzyme's precise function. On a dataset of 11,353 enzymes and 402 classes, we achieved a hierarchical accuracy and Macro-F_1 score of 91.2% and 81.9%, respectively, on the 4th level. Moreover, our method can be used to predict the function of enzyme isoforms with considerable success. This methodology is broadly applicable for genome-wide prediction that can subsequently lead to automated annotation of enzyme databases and the identification of better/cheaper enzymes for commercial activities.

Availability: The web-server can be freely accessed at http://hecnet.cbrlab.org/.

Contact: hammad.naveed@nu.edu.pk

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Enzymes act as catalysts under mild conditions of pH, temperature and cell pressure (Blanco and Blanco, 2017). They are extremely selective which means that each enzyme only speeds up a specific reaction (Berg *et al.*, 2002). For example, amylase is an enzyme produced by the salivary glands and pancreas (Taniguchi and Honnda, 2009). Its primary purpose is to break down starch into simpler sugars to speed up the process of digestion. An enzyme is classified using the Enzyme Commission (EC) Number (Cornish-Bowden, 2014). The EC number is a numerical classification scheme for enzymes based on the chemical reactions they catalyze. Up until August 2018, there were a total of 6 major classes. Recently, Class 7 (Translocases) has been added to the hierarchy. To further explain the hierarchy, we use the example of amylase. It has an EC number of 3.2.1.1. The 3 (level 1) indicates that it is a hydrolase, the 2 (level 2) indicates that it is a glycosylase (hydrolyzes glycosyl compounds), the 1 (level 3) indicates that it is a glycosidase (hydrolyzes O- and S-glycosyl compounds) and finally the last 1 (level 4) indicates that it is an alpha-amylase (hydrolyses the alpha bonds). Figure 1 illustrates the hierarchical structure according to which each enzyme is classified.

Since enzymes are one of the most essential proteins, the area of enzyme function prediction holds great significance. Due to this importance and the cumbersome nature of the experiments used for elucidating their function, several studies have tried to predict their function computationally. **ECPred** is an enzyme function prediction tool that predicts the EC number of an enzyme up to the 4th digit (level 4) (Dalkiran *et al.*, 2018). ECPred uses 3 features: SPMap, BLAST-kNN and Pepstats-SVM. SPMap (Sarac *et al.*, 2008) is a subsequence-based method to predict the protein's function, BLAST-kNN uses the k-nearest neighbor algorithm to calculate the similarities between the query protein and the rest of the proteins using BLAST (Madden, 2013), and Pepstats-SVM is a 37-dimensional vector formed using information regarding the biophysical properties of proteins acquired from Pepstats (Rice *et al.*, 2000). The vector contains peptide statistics for each protein (molecular weight, isoelectric point, physiochemical properties, etc.) which were then fed into an SVM classifier. ECPred's F_1 score, recall and precision for all the levels, were greater than 0.96 for ECPred. The reason for such unusually high numbers is that ECPred uses the full database of Swiss-Prot (248,000 proteins) without redundancy reduction. Two enzymes, with a similar sequence, would have a higher probability of giving a correct prediction if one is present in the training set while the other is present in the test set.

EzyPred uses a top-down approach for predicting an enzyme's functional class (Shen and Chou, 2007). It is able to predict an enzyme's function based on the EC number up to the 2nd digit. EzyPred uses Functional Domain Composition and Pse-PSSM features along with an optimized version of the kNN algorithm to help classify the enzymes up to level 2. They measured their results using success rate which was above 90% for levels 0, 1 and 2.

Deep learning approaches do not require manual construction of complex features or dimensionality uniformization as they have the ability to extract meaningful and significant features and perform dimensionality reduction on their own. Due to this capability, deep learning approaches have the capacity to perform dimensionality compression on highly non-uniform input spaces. **DEEPre** is an example of a deep learning based EC number prediction tool (Li *et al.*, 2017). DEEPre's prediction is up to the 3rd digit of the EC number. The features used by DEEPre include: one hot encoded protein sequence, PSSM, solvent accessibility information, secondary structure information, and functional domains. They employ a hierarchical strategy for the classification of an enzyme. The first classification they tackle is between an enzyme and a non-enzyme. If the protein is an enzyme, the main class of the enzyme is predicted next. This is followed by the prediction of the sub-class and the sub-sub class (level 3). The model comprises of a CNN followed by an LSTM for the first four features and a simple feedforward neural network for the functional domain feature. DEEPre achieved an accuracy and F_1 score of greater than 93 percent across the first 3 levels. However, DEEPre employed a model for each class that it aimed to classify further at the lower levels. Moreover, it was unable to predict the labels of the 4th level due to limited instances in each class.

In this work, we propose "Siamese Triplet Network (STNet)", which is inspired from Siamese and Triplet Networks. Thus far, Siamese Networks have been used in the problem domains of: Natural Language Processing (text similarity) (Neculoiu *et al.*, 2016), Image/Face Recognition and Verification (Chopra *et al.*, 2005; Koch *et al.*, 2015) and quite recently, alignment-free sequence comparison (Zheng *et al.*, 2018) and drug response similarity prediction (Jeon *et al.*, 2019). We proposed a hybrid approach consisting of a single network approach similar to DEEPre for the levels in the enzyme hierarchy that have an abundance of enzymes and STNet approach to deal with the scarce data at the lower levels. This framework is named "Hierarchical Enzyme Classification Network (HECNet)". We achieved an accuracy of 91.2% and an F_1 score of 81.9% at the 4th level even though few examples existed for many of the classes at the last level.

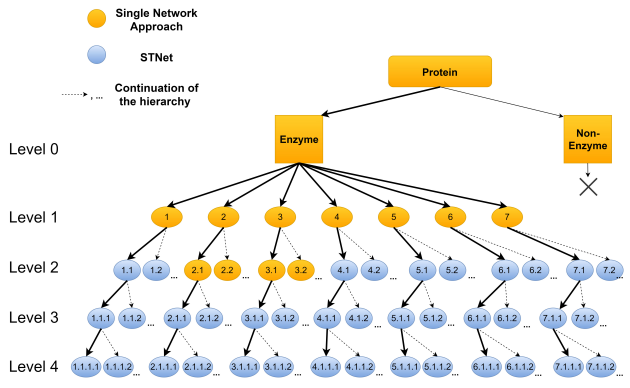

Fig. 1. The Hierarchical Structure of the EC Number and Classification Model Details. The yellow portion indicates the single network approach while the blue portion indicates the classification performed by the Siamese Triplet Network (STNet) approach.

2 Materials and Methods

2.1 Dataset

We have used all the proteins from the Swiss-Prot (Bairoch and Apweiler, 2000) database (January 2019) for this study. This dataset was divided into two main sets: enzymes and non-enzymes. For the enzymes set, all the Swiss-Prot entries for the 7 major enzyme classes were gathered (up to the 4th level). For the non-enzymes set, we subtracted the enzyme entries from the complete Swiss-Prot database thereby retrieving only the non-enzyme proteins for level 0. Both sets were then cleaned using the following steps:

- To make sure the data was non-redundant, we used CD-HIT (Fu *et al.*, 2012) to cull the sequences at a similarity threshold of 40% following DEEPre and EzyPred.
- Any sequences containing amino acids other than the 20 naturally occurring ones, were removed.
- All the enzymes belonging to more than one class were removed.
- All the protein sequences having a length of less than 50 and greater than or equal to 1000 amino acids were removed.

After this process, we were left with 25,272 enzymes and 55,646 non-enzymes belonging to level 0. From the non-enzymes, we selected 25,272 proteins at random to ensure a balanced dataset. The remaining 30,374 non-enzymes were used later on to combine with another validation dataset.

2.2 Data Preparation

As explained earlier, we had 25,272 enzymes at levels 0 and 1 (the dataset is given in Supplementary file 2). However, when we go further down the hierarchy towards levels 2, 3 and 4, the enzyme count decreases. This is due to the fact that the number of enzymes per class decreases at each ensuing level as the number of enzymes at the above level get divided into the subsequent sub-classes at the lower level as can be seen in Figure 1. Also, Swiss-Prot has a number of enzymes that are characterized up to certain levels of the EC number only, meaning that the entire EC number is not known for several enzymes.

Many classes were present which had a small number of examples to enable proper learning in a classification model and therefore we did not include them in the final dataset. So, after setting a threshold of a minimum

of 10 enzymes per class the enzyme count decreased to 11,353. These enzymes belonged to a total of 402 classes at the 4th level. Supplementary Section S1 further explain the class distribution and the number of enzymes per class. We split our dataset into the training and testing sets using an 80% and 20% split. This resulted in 2,420 enzymes in our test set and the remaining in our training set, 8,933 enzymes. This dataset can be found in Supplementary file 3.

2.3 Features

The raw input space comprised of two different categories of features: sequence-length dependent features and sequence-length independent features. Sequence-length dependent features consist of: the sequence itself, PSSM, disordered regions, secondary structure and solvent accessibility. Whereas, sequence-length independent features are made up of amino acid composition and functional domains.

In order to extract the following features, only the sequences of the enzymes and non-enzymes were used:

- **Sequence encoding**: Each sequence was represented using one-hot encoding where 1s and 0s were used to represent each amino acid's position. As a result, each amino acid in a sequence was represented by one 1 and nineteen 0s. Thus, for each protein sequence, an Lx20 matrix was produced where **L** represents the sequence length.
- **Position-Specific Scoring Matrix (PSSM)**: Each PSSM (Schäffer *et al.*, 2001) was obtained using PSI-Blast (Madeira *et al.*, 2019) with Swiss-Prot as the reference database. We performed 3 iterations of PSI-Blast with an E-Value of 0.002. Hence, for each sequence, an Lx20 matrix was generated.
- **Disordered Regions, Secondary Structure and Solvent Accessibility**: When given a sequence, RaptorX-Property Wang *et al.* (2016a,b) returns the probabilities of each amino acid being an ordered residue or a disordered residue. It also returns the probabilities of each amino acid folding into one of the three secondary structures: an alpha helix, beta sheet or random coil. Finally, it also returns the probabilities of each amino acid being in one of the 3 states of solvent accessibility: buried, medium or exposed. Hence, for each sequence, we retrieved an Lx8 matrix that gave us the predicted disordered regions, secondary structure and solvent accessibility information.
- **Amino Acid Composition and Functional Domains**: For each sequence, we construct a vector of 20 amino acid counts (Lee *et al.*, 2006) which gives us the amino acid composition for each sequence. Also, for each sequence, HMMER (hmmscan) (Finn *et al.*, 2011) was used to search against the Pfam database (El-Gebali *et al.*, 2018) (17929 entries as of Pfam 32.0, September 2018) to identify functional domains. For each region where a hit was generated, 1 was marked in a 17929-dimension vector.

2.4 Data Normalization

In this key step, we normalized the PSSMs by stacking them vertically on top of each other and then computing the mean and variance in depth (see Figure 2). As a result, we were able to get position specific means, standard deviations of the PSSM matrix. The dimensions of both the mean and standard deviations matrices were equal to Lx20 where L is the maximum length of the sequence in our dataset. Each row in the mean and standard deviation matrix was only affected by those enzymes whose length was at least equal to that row's index. We also computed the means and standard deviation vectors for the residue counts vectors for the residue counts of shape 1x20. Normalization for disordered regions, solvent accessibility and secondary structure was not required since their values ranged between 0 and 1.

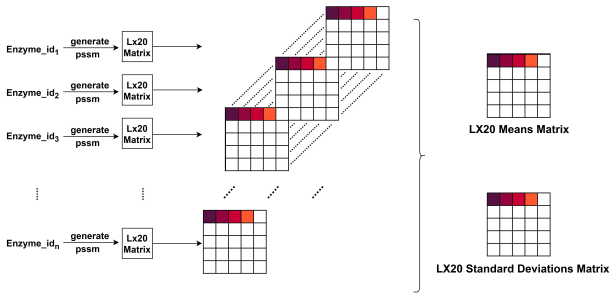

Fig. 2. Each enzyme sequence was used to generate an Lx20 PSSM. All the PSSMs generated were stacked on top of each other and the means and standard deviations at each position were calculated. For example, the first value of the matrices (purple) were used to calculate the means and standard deviations for that specific position.

2.5 Model

2.5.1 Basic Structure of the classification models: Single Network Approach and STNet

The first component of the base model, the feature extractor, was made up of two modules: CNN (Convolutional Neural Network) and RNN (Recurrent Neural Network) module. The CNN module was made up of convolutional and max-pooling layers that work as a feature extractor for the length dependent features. CNNs are not dependent upon the length and width of the input features and are only dependent on their depth, which was 1 for all our input features because they were either 1D or 2D matrices. Therefore, CNNs are able to handle inputs having different dimensions and solve the problem of uniformizing the dimensions. After passing all the length dependent features (PSSMs, solvent accessibility, etc.) through the CNN module, the resultant feature maps were flattened and then vertically stacked together forming 1D vectors which then became the input of the RNN module (Figure 3). The RNN module, made up of long short term memory (LSTM) cells (Hochreiter and Schmidhuber, 1997), is used to extract sequential features from the output of the previous layers. As for the length independent features, they were passed directly to the fully connected layers of a feedforward neural network, after which their output was concatenated with the output of the RNN module.

The first component of the single network approach and STNet (all branches) consisted of 4 convolutional layers, 3 max-pooling layers, 3 LSTM layers and 4 dense layers (2 layers for the residue counts and 2 for the functional domains). The second component is different for both approaches and is discussed in the next two sections.

2.5.2 Single Network Approach

For this approach, the second component acted as a classifier. We employed weighted categorical cross entropy loss function to train the CNN and RNN modules. The penalties of the loss function were weighted according to the class distribution which meant that smaller classes were given more importance than the larger classes. During training, the training error generated was back propagated to each module. This error would weigh more on the features that improve the overall performance and less on the less significant features thus providing an end-to-end feature selection. As a result, the weights of both modules would be adjusted to adopt the change.

The above proposed model, while extremely efficient and flexible, brings about the heavy risk of overfitting. This problem was handled by employing a method called weight decay (Krogh and Hertz, 1992) which decreases the risk of overfitting (Li *et al.*, 2017). Weight decay is the mechanism by which the learning rate reduces after each epoch by a fixed

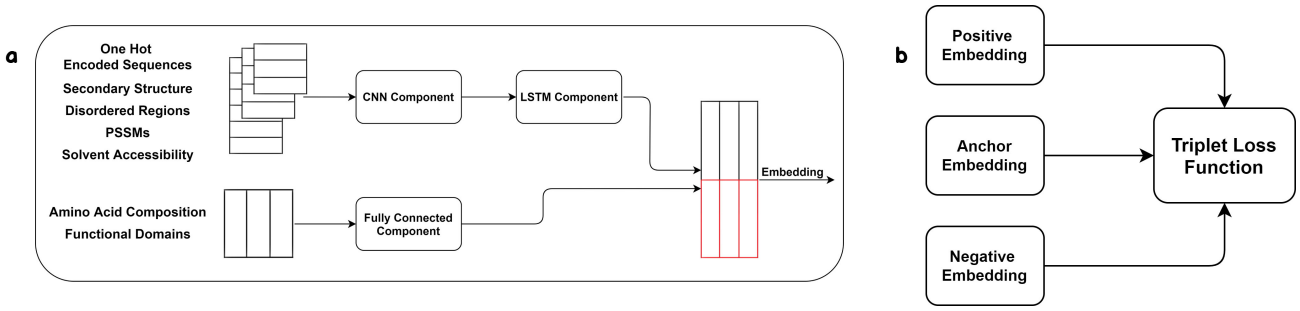

Fig. 3. The Siamese Triplet Network (STNet). The network consists of three branches. Each branch's architecture can be seen in (a). Each branch takes as input the positive, anchor and negative enzyme, respectively. Each enzyme and its subsequent features pass through the CNN, LSTM and the fully connected components and ultimately, generate the embeddings. The three embeddings generated by the three branches of STNet are used to compute the triplet loss as seen in (b). In case of the single network approach, a network similar to a single branch of STNet was used. The only difference was that instead of a triplet loss function, the output was fed into a softmax loss function.

percentage. To minimize the weighted cross entropy loss, we used Adam (Adaptive Moment Estimation) (Kingma and Ba, 2014) as the optimizer.

2.5.3 STNet Approach

STNet (see Figure 3) consisted of 3 identical networks that shared the same weights. This network draws inspiration from the siamese network (Koch *et al.*, 2015) and triplet network approach (Hoffer and Ailon, 2015). Each example consisted of 3 enzymes, 2 of which belonged to the same class while the third belonged to a different class. Each example is called a triplet. The enzymes were split into the training and test sets. After this, triplets of both sets were made separately thus ensuring that an enzyme in a triplet belonging to the test set would never be found in a triplet of the training set. Each triplet contains an anchor enzyme, a positive enzyme and a negative enzyme. The anchor and positive enzymes belonged to the same class while the negative enzyme belonged to a different class. Most of the classes at the fourth level had a small number of enzymes. Therefore, we used all the possible triplets (25,477,869) from the training set with the restriction that the parent class (1, 4, 5, 6, 7, sub-classes of 2 and 3) of the positive and negative enzyme is the same.

The goal of STNet is to find feature embeddings in such a way that the embeddings of the anchor and positive enzyme are close together, while the embeddings of the anchor and negative enzyme are far apart, in terms of euclidean distance (Danielsson, 1980). Therefore, for this approach, the second component acted as a feature encoder rather than a classifier. A modified version of triplet loss (Schroff *et al.*, 2015) was used to train STNet:

$$\sum_{i=1}^{t} max\Big[\Big\{\Big(d(a_i, p_i) - d(a_i, n_i)\Big) \times hw_i \times cw_i\Big\} + m_i, 0\Big] \quad (1)$$

where '$\boldsymbol{t}$' is the total number of triplets, '$\boldsymbol{a}_{\mathrm{i}}$' denotes the anchor enzyme embedding, '$\boldsymbol{p}_{\mathrm{i}}$' denotes the positive enzyme embedding and '$\boldsymbol{n}_{\mathrm{i}}$' denotes the negative enzyme embedding. '$\boldsymbol{hw}_{\mathrm{i}}$', a hyperparameter, denotes the hierarchical weight of the triplet and '$\boldsymbol{cw}_{\mathrm{i}}$' is the class weight of a class that is dependent on its distribution in the dataset. If a class is abundant in enzymes then the class weight assigned to that class will be lower as compared to a class with fewer enzymes. Finally, '$\boldsymbol{m}_{\mathrm{i}}$' is another hyperparameter that enforces a margin between the positive and negative pairs (Schroff *et al.*, 2015). The difference between the embeddings was calculated using the euclidean distance, '$\boldsymbol{d}$'. The resultant value was then multiplied with the class weights and hierarchical weights which was finally added to the margin. A max function is enforced so that we are always left with a result that is not negative. This value is the individual loss of one triplet. Hence, to calculate the total triplet loss, we take the sum of the loss generated for each triplet.

The loss function depends on how deep we go into the EC number's hierarchy. The values of the hierarchical weights and margins are dependent on the class of the negative example. '$\boldsymbol{m}_{\mathrm{i}}$' and '$\boldsymbol{hw}_{\mathrm{i}}$' are dependent on this depth. Suppose we are going to level 4 and we encounter two different classes. They have the same digits up to level 3. So, the negative class, in this case, has a difference of only one digit. Hence, the penalty will be smaller and '$\boldsymbol{hw}_{\mathrm{i}}$' will be 0.5 for level 4. For level 3, the penalty will be greater at 0.7. And finally, at level 2, the penalty will be at a maximum of 1. The value of margin for levels 4, 3 and 2 was assigned as 0.1, 0.15 and 0.2, respectively following the above mentioned strategy. A separate feedforward neural network took the feature embeddings produced by STNet as input to classify the enzymes up to level 4 as seen in Figure 4. We employed weighted categorical cross entropy loss to train this feedforward neural network and used Adam as the optimizer.

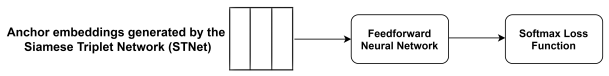

Fig. 4. After training STNet for a class, the anchor embeddings of that class are fed into the feedforward neural network whose last layer corresponds to the number of neurons which are equal to the number of classes at level 4 of that particular class. Finally, a softmax loss function is used to train the network.

3 Results and Discussion

3.1 Model Overview

The EC number has a hierarchical structure which makes the enzyme function prediction problem a typical, yet complex, hierarchical classification problem. We employed two approaches for the classification. The first method is a single network approach similar to the classification approach performed by DEEPre (Li *et al.*, 2017). The second approach named STNet is inspired from the Siamese Network (Bromley *et al.*, 1994) and the Triplet Network (Hoffer and Ailon, 2015). The architecture of STNet is explained in Figure 3.

Following the single network approach, a level-by-level strategy was employed for the classification of: a protein into an enzyme or non-enzyme (level 0) and an enzyme into one of the 7 classes (level 1). We further used this approach for the classification of the enzymes belonging to Classes 2 and 3 (of level 1) into their respective sub-classes (see Figure 1). The reason behind following this approach is that, at the higher levels, the enzyme count is extremely high whereas, at the lower levels, the amount of data is significantly less. To deal with the data scarcity at the lower levels,

we used STNet to expand the data by creating triplets. STNet classified the enzyme (up to the 4th digit) for the classes: 1, 4, 5, 6 and 7 and the sub-classes of 2 and 3.

The base model was made up of 2 components. The first component, common to both approaches, acted as a feature extractor which also performed dimensionality uniformization on the length dependent features. The second component acted as a classifier for the single network approach and as a feature encoder for STNet.

3.2 Comparison with previous methods

To evaluate the performance of our models, we used several metrics. For the initial classification task (a binary classification problem) between an enzyme and non-enzyme, we used: accuracy and F_1 score. For the classifications performed at the lower levels (multi-class classification problem), we used accuracy and Macro-F_1 score (Goutte and Gaussier, 2005). The formula for Macro-F_1 score is given in Supplementary Section S2. All the results, mentioned below, are produced using an ensemble of the output produced by the 3 models in our 3-fold cross validation.

Table 1 shows the results of HECNet, DEEPre, ECPred and EzyPred on HECNet's test set of 2,420 enzymes and 4,559 non-enzymes (after the removal of class 7 from HECNet's set, we were left with 2,396 enzymes to test on the other networks). Since EzyPred and DEEPre's source codes were not available, their web servers were used. The detailed information about the train and test split in the dataset of EzyPred and DEEPre was not available.

Therefore, we tested the entire HECNet test set on the web servers of DEEPre and EzyPred without any knowledge of their training and testing set split (this means that the performance of these tools is being overestimated). For ECPred the training and test set was available, therefore we subtracted ECPred's training data from HECNet's testing data. We were left with only 87 enzymes and 3,848 non-enzymes to test on ECPred. Among the existing methods, EzyPred classifies an enzyme up to level 2 only, while DEEPre and ECPred can classify a given enzyme up to levels 3 and 4, respectively. DEEPre's performance was superior than the other state of the art tools. DEEPre's Macro-F_1 score for levels 0, 1, 2 and 3 was 92.4%, 89.2%, 74.1% and 61.1%, respectively. HECNet performs better than all other networks on almost every level in the hierarchy. On level 4, HECNet reported a Macro-F_1 score of 81.9% as compared to 11.3% of ECPred. Similarly, on level 3, HECNet achieved a Macro-F_1 score of 79.4% as compared to 11.0% and 61.1% achieved by ECPred and DEEPre, respectively.

Table 1. Results of HECNet, DEEPre, ECPred and EzyPred using HECNet's test set of 2,420 enzymes and 4,559 non-enzymes. The best values are shown in bold. Acc and F_1 indicate the accuracy and F_1 score, respectively. '-' shows the unavailability of the score due to the tool's limitation in going down the EC number's hierarchy.

	HECNet		ECPred		DEEPre		EzyPred	
	Acc	F_1	Acc	F_1	Acc	F_1	Acc	F_1
Level 0	**94.0**	**94.1**	75.0	83.8	92.3	92.4	72.3	73.0
Level 1	**93.4**	87.5	74.3	15.9	91.8	**89.2**	70.3	72.9
Level 2	**93.0**	**77.2**	74.0	11.9	91.3	74.1	68.4	58.9
Level 3	**92.8**	**79.4**	74.0	11.0	90.8	61.1	-	-
Level 4	**91.2**	**81.9**	73.8	11.3	-	-	-	-

3.3 July2019 Dataset

In order to further test our methodology, we downloaded the enzymes from Swiss-Prot for July, 2019 and subtracted it from our 11,353 enzymes (January, 2019). Following this, the redundancy reduction cutoffs were applied. This dataset was combined with the 30,374 non-enzymes that were not included in our initial non-enzymes dataset. Together, this dataset shall henceforth be known as the July2019 dataset and can be found in Supplementary file 4. The performance of HECNet, DEEPre, EzyPred and ECPred on the July2019 dataset is shown in Table 2. As in the previous section, we checked the overlap between this dataset and the training set of ECPred. After removing the overlap, we were left with 1,096 enzymes and 25,459 non-enzymes which were used to test the performance of ECPred. For EzyPred, DEEPre and HECNet the July2019 dataset was slightly altered so that it contained only the classes that existed in the respective tools' datasets to ensure a fair comparison. Similar to the last experiment, HECNet performs better on nearly all the levels as compared to the previous methodologies. HECNet achieved a Macro-F_1 score of 76.0% as compared to 3.4% of ECPred on level 4. Similarly, on level 3, HECNet achieved a Macro-F_1 score of 74.1% as compared to 10.7% and 54.7% achieved by ECPred and DEEPre, respectively.

Table 2. Results of HECNet, DEEPre, ECPred and EzyPred using the July2019 dataset of 12,889 enzymes and 30,374 non-enzymes. The best values are shown in bold. Acc and F_1 indicate the accuracy and F_1 score, respectively. '-' shows the unavailability of the score due to the tool's limitation in going down the EC number's hierarchy.

	HECNet		ECPred		DEEPre		EzyPred	
	Acc	F_1	Acc	F_1	Acc	F_1	Acc	F_1
Level 0	94.0	94.1	74.7	83.3	**95.9**	**95.9**	71.7	72.8
Level 1	**93.6**	82.8	74.1	20.8	91.8	**87.3**	69.4	68.5
Level 2	**93.5**	**70.6**	73.6	16.9	88.8	63.4	66.4	66.1
Level 3	**93.3**	**74.1**	73.2	10.7	86.9	53.3	-	-
Level 4	**92.5**	**76.0**	72.7	3.4	-	-	-	-

In order to check whether the performance of our model was due to an increased number of annotated enzymes, we constructed another dataset which included enzymes with annotations before December 2015. This dataset had approximately 10% fewer enzymes and classes at level 4. We retrained our model using this dataset in the same manner as described above and found no significant change in the performance (SI section S3).

3.4 Isoform Analysis

Isozymes (Markert and Møller, 1959) are the protein isoforms of enzymes. A single protein can have multiple isoforms that are produced by the same gene. These isoforms usually perform the same function but they can have large differences in sequence length. An extremely hard test for the enzyme function prediction methods would be to predict correctly the function of the isoforms having different lengths. Therefore, we gathered all the isozymes from Swiss-Prot. One of the isoforms from the enzymes was chosen as the base or canonical sequence. We removed the isoforms that had a sequence length of less than 50 or greater than or equal to 1000 and any isoforms that belonged to multiple classes. Moreover, only those isoforms for which experimental confirmation was available were retained. We found 2,048 isoforms belonging to 139 classes after removing all the

enzymes that belonged to our training set. The isoforms dataset is available in Supplementary file 5. As in the previous two sections, we also checked the overlap of this dataset with the training set of ECPred. After removing the overlap, we were left with 1478 isoforms which were used to test the performance of ECPred. For EzyPred, DEEPre and HECNet all the isoforms were used to test their performance.

Figure 5 shows the results of testing the isoforms on HECNet and other enzyme function prediction tools. HECNet outperformed the other methodologies in terms of F_1 score on almost all levels, particularly on levels 3 and 4.

Three case studies about Adenosine deaminase 2, ATP-dependent (S)-NAD(P)H-hydrate dehydratase and Mitochondrial peptide methionine sulfoxide reductase are discussed below. The enzyme Adenosine deaminase 2 contributes to the degradation of extracellular adenosine (Zavialov *et al.*, 2010). It belongs to class 3.5.4.4 that contains 12 examples on level 4 in our dataset. The enzyme possessed two isoforms of length 511 and 270 and HECNet managed to predict both isoforms as belonging to the same class up to the fourth level, 3.5.4.4. ATP-dependent (S)-NAD(P)H-hydrate dehydratase is an enzyme belonging to class 4.2.1.93 (Van Bergen *et al.*, 2018). Its role is to catalyze the dehydration of the S-form of NAD(P)HX by using ATP. Class 4.2.1.93 only had 11 examples to learn from in our dataset. The enzyme had 3 isoforms of length 347, 390 and 126. HECNet correctly classified all 3 up to the fourth level.

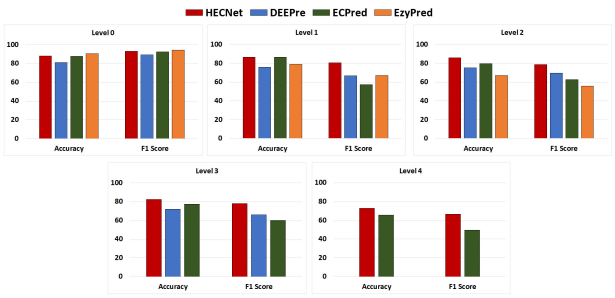

Fig. 5. Comparison of HECNet, DEEPre, EzyPred and ECPred on the isoforms dataset. DEEPre's F_1 score for levels 0, 1, 2 and 3 was 89.1%, 66.7%, 69.5% and 66.0%, respectively. HECNet achieved an F_1 score of 66.5% compared to 49.7% of ECPred on level 4. Similarly, on level 3, HECNet reported an F_1 score of 78.0% as compared to 60.0% and 66.0% achieved by ECPred and DEEPre, respectively.

Finally, Mitochondrial peptide methionine sulfoxide reductase is an enzyme that belongs to class 1.8.4.11 and is known to have 3 isoforms (Hansel *et al.*, 2002). The unique thing about this enzyme is that the third isoform shows no enzymatic activity. HECNet correctly classified the first and second isoform into class 1.8.4.11 while the third isoform was classified as a non-enzyme at the level 0 classification. Inferring the function of the third isoform using homology modelling would results in a wrong prediction as all isoforms have high sequence similarity.

Figure 6 shows the structures of a canonical enzyme and the modeled structure of the isoform using I-TASSER (Roy *et al.*, 2010) that does not show enzymatic activity. The modeled structure has a predicted Root Mean Square Deviation (RMSD) of 14 ± 3.9 Å, indicating that even though a template structure was available we could not model the structure of the inactive enzyme. As the structure could not be modeled, therefore predicting the function using the structure is not possible. However, our model is able to capture the functionally important details right from the sequence without having to model the structure of the enzymes.

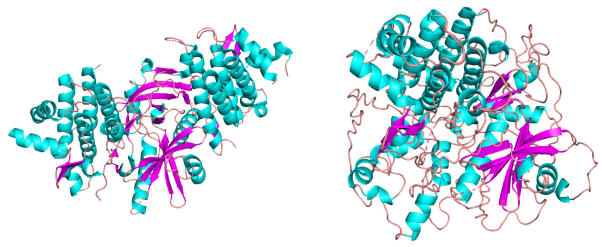

Fig. 6. 3D structure of an enzyme (left) and the modeled non-enzyme (right) from the isoforms dataset. The modeled structure did not have sufficient quality to infer function.

3.5 Number of Models trained

In hierarchical classification, the usual approach is to assign a model to each class whose lower levels are to be predicted. Using this approach, DEEPre employed 72 models for classification up to the 3rd level and if their approach was used on our dataset, it would have resulted in 163 models for classification up to the 4th level. On the other hand, we were able to classify enzymes up to the fourth level with only 22 models. Using the single network approach, we trained a total of 4 models. The initial model was used to classify a protein between an enzyme and a non-enzyme, followed by a model to classify an enzyme into one of the seven major classes. The final two models were applied on classes 2 and 3 to classify them into their respective sub-classes at level 2. Using the STNet approach, the total number of models trained were 18. For the major classes of the first level (1, 4, 5, 6 and 7), we trained one model each to classify them up till the fourth level. The next 13 models were used for the sub-classes of classes 2 and 3 to classify them up to level 4 (see Supplementary Section S1 for the distribution). This is significant as a smaller number of models indicate generic rules that are less prone to be dataset specific.

3.6 Annotation Completion

As hundreds of proteins are being discovered at a fast pace due to high throughput genomic studies and experimental studies to elucidate their function are costly and time consuming, a large number of proteins are either unannotated or partially annotated (Bairoch and Apweiler, 2000). Using our methodology, we tried to complete the annotation of partially annotated proteins. We acquired the original datasets (without redundancy reduction cutoffs). Proteins with partial annotations were fed to our model to complete their annotations. Prediction for a large number of enzymes (around 30000) up to the fourth level is given in Supplementary file 1.

3.7 Sensitivity Analysis

To test the robustness of our model, we conducted a sensitivity analysis on the hyperparameters, '$\boldsymbol{hw}$' and '$\boldsymbol{m}$'. Initially, our models were trained on the '$\boldsymbol{hw}$' values of 0.5, 0.7 and 1 for levels 4, 3 and 2, respectively. We modified these values and performed experiments on classes 1, 4 and 6 which satisfied two crucial conditions: a) an abundant number of enzymes were present and b) a wide class distribution on all levels. No significant change was observed in the F_1 score on a wide range of the hyperparameters, '$\boldsymbol{hw}$' and '$\boldsymbol{b}$', as long as the relative ratio on the hierarchical levels is preserved (Supplementary Section S4).

3.8 Limitations

In the current study, the enzymes that belonged to more than one class were removed. These enzymes are known as multi-functional enzymes. The classification of an enzyme to more than one class is a significantly challenging problem. One of the major contributions towards tackling

it has been achieved by mlDEEPre Zou *et al.* (2018). mlDEEPre trains two models; The first model determines whether an enzyme is a multi-functional enzyme or a mono-functional enzyme. If an enzyme is a multi-functional enzyme, it is passed on to the second model which determines all the classes which the enzyme belongs to at the first level. The second model is trained using a multi-label loss function. We plan to handle the multi-functional enzymes part of our dataset in future studies with an approach similar to the one adopted by mlDEEPre. Moreover, even though our algorithm has the ability to augment data, a minimum number of 10 enzymes are still required to generate enough triplets for adequate training of the network.

4 Conclusion

In this article, we proposed a novel deep learning architecture (STNet) that can handle the scarce nature of data at the lower levels of a hierarchical classification problem. Moreover, we also proposed HECNet, a computational framework that uses STNet for classifying an enzyme into its EC number till the most specific (4th) level. We have demonstrated the performance of HECNet on all enzymes in the Swiss-Prot database and the enzymes that were discovered subsequently. We also demonstrate the utility of HECNet in completing the annotation of partially annotated enzymes. A very stringent test of HECNet on predicting the function of isoforms of the same enzyme also produced adequate results. We compared our methodology with the state-of-art enzyme function prediction methods and show that our method performs significantly better than previous methods particularly on levels 3 and 4. This is the first study that has successfully tackled the problem of EC number prediction up to the 4th level. Our method is broadly applicable for genome-wide prediction of enzyme function that can subsequently lead to identification of better and cheaper enzymes for commercial activities.

Acknowledgements

We would like to thank Dr. Sibt ul Hussain and Mr. Atique ur Rehman for useful discussions on the deep learning models.

Funding

This research work was funded by the Higher Education Commission of Pakistan and the Ministry of Planning Development and Reforms under the umbrella of National Center in Big Data and Cloud Computing (NCBC).

References

Bairoch, A. and Apweiler, R. (2000). The swiss-prot protein sequence database and its supplement trembl in 2000. *Nucleic acids research*, **28**(1), 45–48.

Berg, J., Tymoczko, J., and Stryer, L. (2002). Enzymes are powerful and highly specific catalysts.

Blanco, A. and Blanco, G. (2017). Chapter 8 - enzymes. In A. Blanco and G. Blanco, editors, *Medical Biochemistry*, pages 153 – 175. Academic Press.

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., and Shah, R. (1994). Signature verification using a" siamese" time delay neural network. In *Advances in neural information processing systems*, pages 737–744.

Chopra, S., Hadsell, R., LeCun, Y., *et al.* (2005). Learning a similarity metric discriminatively, with application to face verification. In *CVPR (1)*, pages 539–546.

Cornish-Bowden, A. (2014). Current iubmb recommendations on enzyme nomenclature and kinetics. *Perspectives in Science*, **1**(1), 74 – 87. Reporting Enzymology Data – STRENDA Recommendations and Beyond.

Dalkiran, A., Rifaioglu, A. S., Martin, M. J., Cetin-Atalay, R., Atalay, V., and Doğan, T. (2018). Ecpred: a tool for the prediction of the enzymatic functions of protein sequences based on the ec nomenclature. *BMC bioinformatics*, **19**(1), 334.

Danielsson, P.-E. (1980). Euclidean distance mapping. *Computer Graphics and image processing*, **14**(3), 227–248.

El-Gebali, S., Mistry, J., Bateman, A., Eddy, S. R., Luciani, A., Potter, S. C., Qureshi, M., Richardson, L. J., Salazar, G. A., Smart, A., *et al.* (2018). The pfam protein families database in 2019. *Nucleic acids research*, **47**(D1), D427–D432.

Finn, R. D., Clements, J., and Eddy, S. R. (2011). Hmmer web server: interactive sequence similarity searching. *Nucleic acids research*, **39**(suppl_2), W29–W37.

Fu, L., Niu, B., Zhu, Z., Wu, S., and Li, W. (2012). Cd-hit: accelerated for clustering the next-generation sequencing data. *Bioinformatics*, **28**(23), 3150–3152.

Goutte, C. and Gaussier, E. (2005). A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In *European Conference on Information Retrieval*, pages 345–359. Springer.

Hansel, A., Kuschel, L., Hehl, S., Lemke, C., AGRICOLA, H.-J., Hoshi, T., and Heinemann, S. H. (2002). Mitochondrial targeting of the human peptide methionine sulfoxide reductase (msra), an enzyme involved in the repair of oxidized proteins. *The FASEB Journal*, **16**(8), 911–913.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, **9**(8), 1735–1780.

Hoffer, E. and Ailon, N. (2015). Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, pages 84–92.

Jeon, M., Park, D., Lee, J., Jeon, H., Ko, M., Kim, S., Choi, Y., Tan, A.-C., and Kang, J. (2019). Resimnet: Drug response similarity prediction using siamese neural networks. *Bioinformatics*.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Koch, G., Zemel, R., and Salakhutdinov, R. (2015). Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2.

Krogh, A. and Hertz, J. A. (1992). A simple weight decay can improve generalization. In *Advances in neural information processing systems*, pages 950–957.

Lee, S., Lee, B.-c., and Kim, D. (2006). Prediction of protein secondary structure content using amino acid composition and evolutionary information. *Proteins: Structure, Function, and Bioinformatics*, **62**(4), 1107–1114.

Li, Y., Wang, S., Umarov, R., Xie, B., Fan, M., Li, L., and Gao, X. (2017). Deepre: sequence-based enzyme ec number prediction by deep learning. *Bioinformatics*, **34**(5), 760–769.

Madden, T. (2013). The blast sequence analysis tool. In *The NCBI Handbook [Internet]. 2nd edition*. National Center for Biotechnology Information (US).

Madeira, F., Lee, J., Buso, N., Gur, T., Madhusoodanan, N., Basutkar, P., Tivey, A., Potter, S. C., Finn, R. D., Lopez, R., *et al.* (2019). The embl-ebi search and sequence analysis tools apis in 2019. *Nucleic acids research*.

Markert, C. L. and Møller, F. (1959). Multiple forms of enzymes: tissue, ontogenetic, and species specific patterns. *Proceedings of the National Academy of Sciences of the United States of America*, **45**(5), 753.

Neculoiu, P., Versteegh, M., and Rotaru, M. (2016). Learning text similarity with siamese recurrent networks. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 148–157.

Rice, P., Longden, I., and Bleasby, A. (2000). Emboss: the european molecular biology open software suite. *Trends in genetics*, **16**(6), 276–277.

Roy, A., Kucukural, A., and Zhang, Y. (2010). I-tasser: a unified platform for automated protein structure and function prediction. *Nature protocols*, **5**(4), 725.

Sarac, O. S., Gürsoy-Yüzügüllü, Ö., Cetin-Atalay, R., and Atalay, V. (2008). Subsequence-based feature map for protein function classification. *Computational biology and chemistry*, **32**(2), 122–130.

Schäffer, A. A., Aravind, L., Madden, T. L., Shavirin, S., Spouge, J. L., Wolf, Y. I., Koonin, E. V., and Altschul, S. F. (2001). Improving the accuracy of psi-blast protein database searches with composition-based statistics and other refinements. *Nucleic acids research*, **29**(14), 2994–3005.

Schroff, F., Kalenichenko, D., and Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823.

Shen, H.-B. and Chou, K.-C. (2007). Ezypred: a top–down approach for predicting enzyme functional classes and subclasses. *Biochemical and biophysical research communications*, **364**(1), 53–59.

Taniguchi, H. and Honnda, Y. (2009). Amylases. encyclopedia of microbiology.

Van Bergen, N. J., Guo, Y., Rankin, J., Paczia, N., Becker-Kettern, J., Kremer, L. S., Pyle, A., Conrotte, J.-F., Ellaway, C., Procopis, P., *et al.* (2018). Nad (p) hx dehydratase (naxd) deficiency: a novel neurodegenerative disorder exacerbated by febrile illnesses. *Brain*, **142**(1), 50–58.

Wang, S., Peng, J., Ma, J., and Xu, J. (2016a). Protein secondary structure prediction using deep convolutional neural fields. *Scientific reports*, **6**, 18962.

Wang, S., Li, W., Liu, S., and Xu, J. (2016b). Raptorx-property: a web server for protein structure property prediction. *Nucleic acids research*, **44**(W1), W430–W435.

Zavialov, A. V., Gracia, E., Glaichenhaus, N., Franco, R., Zavialov, A. V., and Lauvau, G. (2010). Human adenosine deaminase 2 induces differentiation of

monocytes into macrophages and stimulates proliferation of t helper cells and macrophages. *Journal of leukocyte biology*, **88**(2), 279–290.

Zheng, W., Yang, L., Genco, R. J., Wactawski-Wende, J., Buck, M., and Sun, Y. (2018). Sense: Siamese neural network for sequence embedding and alignment-free comparison. *Bioinformatics*.

Zou, Z., Tian, S., Gao, X., and Li, Y. (2018). mldeepre: Multi-functional enzyme function prediction with hierarchical multi-label deep learning. *Frontiers in genetics*, **9**.