Bioinformatics, 34(5), 2018, 760–769
doi: 10.1093/bioinformatics/btx680
Advance Access Publication Date: 23 October 2017
Original Paper

OXFORD

Sequence analysis

DEEPre: sequence-based enzyme EC number prediction by deep learning

Yu Li[1], Sheng Wang[1], Ramzan Umarov[1], Bingqing Xie[2], Ming Fan[3], Lihua Li[3] and Xin Gao[1,*]

[1]Electrical and Mathematical Sciences and Engineering Division (CEMSE), Computational Bioscience Research Center (CBRC), Computer, King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Saudi Arabia, [2]Computer Science Department, Illinois Institute of Technology, Chicago, IL 60616, USA and [3]Institute of Biomedical Engineering and Instrumentation, Hangzhou Dianzi University, Hangzhou 310018, China

*To whom correspondence should be addressed.
Associate Editor: John Hancock

Received on July 13, 2017; revised on October 11, 2017; editorial decision on October 17, 2017; accepted on October 20, 2017

Abstract

Motivation: Annotation of enzyme function has a broad range of applications, such as metagenomics, industrial biotechnology, and diagnosis of enzyme deficiency-caused diseases. However, the time and resource required make it prohibitively expensive to experimentally determine the function of every enzyme. Therefore, computational enzyme function prediction has become increasingly important. In this paper, we develop such an approach, determining the enzyme function by predicting the Enzyme Commission number.

Results: We propose an end-to-end feature selection and classification model training approach, as well as an automatic and robust feature dimensionality uniformization method, DEEPre, in the field of enzyme function prediction. Instead of extracting manually crafted features from enzyme sequences, our model takes the raw sequence encoding as inputs, extracting convolutional and sequential features from the raw encoding based on the classification result to directly improve the prediction performance. The thorough cross-fold validation experiments conducted on two large-scale datasets show that DEEPre improves the prediction performance over the previous state-of-the-art methods. In addition, our server outperforms five other servers in determining the main class of enzymes on a separate low-homology dataset. Two case studies demonstrate DEEPre's ability to capture the functional difference of enzyme isoforms.

Availability and implementation: The server could be accessed freely at http://www.cbrc.kaust.edu.sa/DEEPre.

Contact: xin.gao@kaust.edu.sa

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

1 Introduction

Enzymes, an essential kind of proteins in the human body, catalyzing reactions *in vivo*, play a vital role in regulating biological processes. Annotation of enzyme function has a broad range of applications, such as metagenomics, industrial biotechnology, and diagnosis of enzyme deficiency-caused diseases. The dysfunction of certain enzymes would cause serious metabolic diseases. For example, the deficiency of alpha-galactosidase, which hydrolyses the terminal alpha-galactosyl moieties from glycolipids and glycoproteins, would cause the Fabry disease, resulting in full body pain, kidney insufficiency, and cardiac complications (Hoffmann *et al.*, 2007). The deficiency of DNA repair enzymes, which recognize and correct the physical damage in DNA, can cause the accumulation of mutations, which may further lead to various cancers (Wood *et al.*,

2001). To investigate the causation of such diseases, an indispensable step of finding a way to cure them, it is crucial to understand the function of the related enzymes first. The most straightforward and accurate way of doing such investigation is through experimental techniques, such as enzymatic assays (Goddard and Reymond, 2004). However, conducting experiments requires significant amount of time and expert efforts, which may not cope with the rapid increase in the number of new enzymes. In this context, computational methods emerged to assist biologists in determining enzyme function and guiding the direction of setting up the validating experiments.

According to SWISS-PROT (Bairoch and Apweiler, 2000) (released on September 7, 2016), among the 539 566 manually annotated proteins, 258 733 proteins are enzymes. Such a large number of enzymes are usually classified using the Enzyme Commission (EC) system (Cornish-Bowden, 2014), the most well-known numerical enzyme classification scheme, which specifies the function of an enzyme by four digits. This classification system has a tree structure. After the root of the tree, there are two main nodes, standing for enzyme and non-enzyme proteins, respectively. The enzyme main node extends out six successor nodes, corresponding to the six main enzyme classes: (i) oxidoreductases, (ii) transferases, (iii) hydrolases, (iv) lyases, (v) isomerases and (vi) ligases, represented by the first digit. Each main class node further extends out several subclass nodes, specifying the enzyme's subclasses, represented by the second digit. With the same logic, the third digit indicates the enzyme's sub-subclasses and the fourth digit denotes the sub-sub-subclasses. Take Type II restriction enzyme, which is annotated as EC 3.1.21.4, as an example, the '3' denotes that it is an hydrolase; the '1' indicates that it acts on ester bonds; the '21' shows that it is an endodeoxyribonuclease producing 5-phosphomonoesters; and the '4' suggests that it is a Type II site-specific deoxyribonuclease. By predicting the EC numbers precisely, computational methods can annotate the function of enzymes. It should also be noted that a substantial number of enzymes annotated with some reactions in databases such as UniProt or Brenda do not have EC numbers associated, which is out of the scope of this study.

A number of computational methods have already been proposed to determine the enzyme function by predicting enzyme EC numbers. There have been three main research directions of this problem since (des Jardins *et al.*, 1997), who used machine learning methodologies and sequence information to investigate the problem for the first time. Firstly, because it is commonly believed that structures determine function, some researches, such as (Dobson and Doig, 2005; Nagao *et al.*, 2014; Roy *et al.*, 2012; Yang *et al.*, 2015; Zhang *et al.*, 2017), focused on predicting the enzyme function by predicting the structure of the enzyme first. After obtaining the structure, they scanned the database or the library, whose entries' EC numbers have already been determined and validated by experiments, and assigned the EC number of the template with the most similar structure to the query. However, structure prediction is still relatively immature and time-consuming. Besides, since both the structure prediction step and the EC number prediction step would cause errors, the accumulated error would have a negative effect on the final prediction result. Second, the common assumption that enzymes with high sequence similarity tend to have similar functionality leads to a number of studies utilizing sequence similarity (Arakaki *et al.*, 2009; Kumar and Skolnick, 2012; Quester and Schomburg, 2011; Tian *et al.*, 2004; Yu *et al.*, 2009). Although this category of methods is widely used in practice, they are unable to make a prediction when encountering a sequence without significant homologies in the current databases. Thirdly, extracting features from the sequence and classifying the enzyme using machine learning algorithms is the most extensively studied direction (Cai *et al.*, 2003, 2004, 2005; Cai and Chou, 2005; Chou, 2005; Chou and Elrod, 2003; De Ferrari *et al.*, 2012; Huang *et al.*, 2007; Kumar and Choudhary, 2012; Lee *et al.*, 2008; Li *et al.*, 2016; Lu *et al.*, 2007; Nasibov and Kandemir-Cavas, 2009; Qiu *et al.*, 2009, 2010; Sharif *et al.*, 2015; Shen and Chou, 2007; Volpato *et al.*, 2013; Wang *et al.*, 2010, 2011; Zhou *et al.*, 2007; Zou and Xiao, 2016). Although this direction has already been studied for over 15 years with a number of softwares and servers available, few of them combine the procedure of feature extraction and classification optimization together. Instead, previous studies rely heavily on manually crafted features, and consider feature extraction and classification as two separate problems. In spite of the success of such methods, with the rapid expansion of the known enzyme sequences, such manually designed features are very likely to be a suboptimal feature representation which may be unsustainable in the omic era.

In addition to those difficulties, another issue in the protein general function prediction field is the feature dimensionality non-uniformity problem, which usually lies in the sequence-length-dependent features, such as PSSM (position-specific scoring matrix). For example, in this paper, the dimensionality of PSSM can range from 50 by 20 to 5000 by 20, according to the corresponding sequence length. The feature uniformity requirement of mainstream classifiers has pushed out three strategies to this problem. First, avoiding using the sequence-length-dependent features is the most straightforward solution to the problem. Although this approach can work under some certain circumstances, it eliminates the possibility of taking advantage of some powerful representation, such as PSSM, which can provide evolutional information. The second solution is to manually derive sequence-length-independent features from the sequence-length-dependent features (Chen *et al.*, 2013, 2014, 2016). Pse-AAC (pseudo amino acid composition) and Pse-PSSM are typical examples of this category, which have been proved successful in a number of applications (Chou, 2009, 2011). The third solution is to systematically generate sequence-length-independent features, such as string kernels (Dai *et al.*, 2017, Leslie *et al.*, 2002, 2004; Rätsch *et al.*, 2005; Wang *et al.*, 2014), which, however, do not consider the classification problem when extracting features. Despite the previous success of these three strategies, they still heavily depend on either manually designed or pre-defined features, which are most likely to be suboptimal. To take full advantage of the bursting of data in recent years, a more robust, automatic framework to extract problem-specific sequence-length-independent features from the sequence-length-dependent ones for dealing with the dimensionality problem is desired.

To conquer the aforementioned limitations, which are homology requirement, feature design and feature dimensionality nonuniformity, here we propose a novel level-by-level prediction approach based on deep learning, by only utilizing the sequence information. The enzyme sequence is represented by two kinds of raw encoding, sequence-length-dependent encoding, such as raw sequence one-hot encoding and PSSM, and sequence-length-independent encoding, such as functional domain (FunD) encoding. Those two kinds of raw representations are combined into a deep learning model with a novel architecture to perform dimensionality uniformization, feature selection and classification model training simultaneously. This paper makes the following contributions: (i) We propose a framework for hierarchical EC number prediction, the idea of which can also be applied to hierarchical classification of protein general function. (ii) To solve the feature dimensionality nonuniformity problem, we propose a robust, automatic framework based on deep

learning to extract problem-specific sequence-length-independent features from the sequence-length-dependent ones. (iii) We propose a sequence-based enzyme EC number predictor, DEEPre, which is based on the above two frameworks. (iv) Two case studies demonstrate our tool's ability of performing functionality prediction of different enzyme isoforms caused by alternative splicing. (v) We investigate the importance of local information in determining the functionality of an enzyme.

2 Related work

In this section, we introduce four representative methods for enzyme function prediction, followed by a brief overview of deep learning and hierarchical classification.

2.1 EzyPred

EzyPred (Shen and Chou, 2007) is a three-level EC number predictor, which predicts whether an input protein sequence is an enzyme, and its main class and subclass if it is. It uses two features, pseudo PSSM (Pse-PSSM) and FunD encoding. Pse-PSSM is developed from the pseudo amino acid, a highly innovative manually designed feature which has already been proved successful in a number of problems (Chou, 2009; Hayat and Khan, 2012). It encodes the PSSM of proteins with different lengths using a uniform length matrix, which not only preserves the average score of the amino acid residues in the whole sequence that were changed to a certain type of amino acid during the evolution process but also avoids the complete loss of the sequence order information. FunD encoding captures the local FunD information, which could be referred to Section 3.2.5. With these two features, EzyPred uses optimized evidence-theoretic *k*-nearest neighbor (OET-KNN) as the classifier, which is an improved version of KNN. By considering not only the label of the KNN of the input query data point but also the distance of the neighbors to the query data as the supporting evidence, OET-KNN alleviates the problem of the original version of KNN for being too sensitive to noise. Although having been developed for 10 years, EzyPred still remains as one of the state-of-the-art methods in predicting enzyme function. Its server is easy-to-use with a user-friendly interface as well.

2.2 SVM-prot

SVM-Prot was proposed in 2004 (Cai *et al.*, 2003, 2004) and updated in 2016 (Li *et al.*, 2016). It can not only predict enzyme functional families but also non-enzyme functional families. It represents the protein sequence using 13 properties, including AAC, polarity, hydrophobicity, surface tension, charge, normalized Van der Waals volume, polarizability, secondary structure, solvent accessibility, molecular weight, solubility, number of hydrogen bond donors in side chain and number of hydrogen bond acceptors in side chain. Employing composition, translation and distribution to encode each of the above properties, SVM-Prot can make prediction irrespective of sequence similarity. Specifically, composition specifies the fraction of amino acids with a particular property; translation specifies the transition percentage of one amino acid with particular property to another amino acid with different properties; distribution specifies the distribution of amino acids with certain property within the first 25, 50, 75, and 100% of the sequence. The original version used support vector machines (SVM) as the classifier, while the 2016 update made two more classifiers, KNN and probabilistic neural networks, available.

2.3 COFACTOR

COFACTOR (Roy *et al.*, 2012; Zhang *et al.*, 2017) is a structure-based protein function annotation web-server. In terms of EC number prediction, for an input structural model, which can be obtained either by experiments or computational modeling, it threads the structure against the template library, whose entries' annotation has already been validated by experiments, to identify the template enzyme with the most similar folds and functional sites. Obtaining the template and assuming that structures determine function, the server assigns the EC number of the template enzyme to the query, with the confidence being evaluated by a function considering both the global similarity and the local similarity. In addition to enzyme function prediction, the server can predict the Gene Ontology (GO) terms and protein-ligand binding interactions as well. COFACTOR has been proved successful in protein–ligand binding interaction prediction in the CASP9 competition (Moult *et al.*, 2011).

2.4 EFICAz

EFICAz (Arakaki *et al.*, 2009; Kumar and Skolnick, 2012; Tian *et al.*, 2004) is an EC number prediction server using combined approaches. In addition to using the sequence similarity, it also incorporates the PROSITE and PFAM database information. The original version consists of four components: (i) pairwise sequence comparison-based enzyme function inference, (ii) conservation controlled hidden Markov model (HMM) iterative procedure for enzyme family classification-based functionally discriminating residue identification, (iii) multiple PFAM-based functionally discriminating residue recognition and (iv) multiple PROSITE pattern recognition. Those four components work independently, determining the final prediction by voting. In the later updates in 2009 and 2012, two more components, multiple PFAM family-based SVM evaluation and conservation controlled HMM iterative procedure for enzyme family classification-based SVM evaluation, and larger databases were added. Although it is unable to make EC number annotation if the query sequence has no homology, this server works pretty well in practice with completely four digits assigned.

2.5 Deep learning and hierarchical classification

Since (Krizhevsky *et al.*, 2012), deep learning has become an extremely popular machine learning method. Its two main architectures, convolutional neural network (CNN) and recurrent neural network (RNN), have made a profound contribution to many bioinformatic problems, such as genetic analysis (Xiong *et al.*, 2015), sequence binding specificity prediction (Alipanahi *et al.*, 2015), and cryo-EM image processing (Wang *et al.*, 2016a). Instead of being a pure classifier that depends on the manually designed features such as SVM, CNN is considered as an end-to-end wrapper classifier, being able to perform feature extraction based on the classification result and improve the performance in a virtuous circle. As a complement to CNN's capability of capturing significant features from a 2D or 3D matrix, RNN has the potential of encoding long term interactions within the input sequence, which is usually a 1D vector, such as the encoding of English words. In our article, we combined the advantages of CNN and RNN, using CNN to conduct feature extraction and dimensionality compression starting from the raw 2D encoding matrices, and using RNN to extract the sequential, long-term interactions within the input sequence.

A classification problem with a tree structure in the label space, such as the enzyme function prediction problem discussed in this article, is often regarded as a hierarchical classification problem. Because this kind of problems can be regarded as multi-label

Table 1. Dataset summary

Dataset	*KNN* dataset	*NEW* dataset	*COFACTOR* dataset
Source	Shen and Chou (2007)	Self-constructed	Roy *et al.* (2012)
Enzymes	9832	22 168	284
Non-enzymes	9850	22 168	—

Note: The *KNN* dataset and *NEW* dataset are used for cross-fold validation. The *COFACTOR* dataset is used for cross-dataset validation.

classification and multi-class classification at the same time, the solutions to the problem can be classified into three categories based on different angles to the problem (Silla and Freitas, 2011), namely, flat classification approach, local classifier approach, and global classifier approach. According to the property of our problem, we chose the local classifier approach, which constructs one classifier for each internal node, to be the overall strategy.

3 Materials and methods

3.1 Datasets

We adopt three datasets in this paper. The first dataset is a widely used one from (Shen and Chou, 2007), constructed from the ENZYME database (released on May 1, 2007), with 40% sequence similarity cutoff. More details of that dataset could be referred to (Shen and Chou, 2007). This dataset is denoted as the *KNN* dataset in the rest of the paper.

Following the same procedure of constructing the *KNN* dataset, we constructed a larger dataset using up-to-date databases. The steps of constructing the dataset are as follows:

i. The SWISS-PROT (released on September 7, 2016) database was separated into enzymes and non-enzymes based on the annotation.
ii. To guarantee the uniqueness and correctness, enzyme sequences with more than one set of EC numbers or incomplete EC number annotation were excluded.
iii. To avoid fragment data, enzyme sequences annotated with 'fragment' or with <50 amino acids were excluded. Enzyme sequences with more than 5000 amino acids were also excluded.
iv. To remove redundancy bias, we used CD-HIT (Fu *et al.*, 2012) with 40% similarity threshold to sift upon the raw dataset, resulting in 22 168 low-homology enzyme sequences.
v. To construct the non-enzyme part, 22 168 non-enzyme protein sequences were randomly collected from the SWISS-PROT (released on September 7, 2016) non-enzyme part, which were also subject to the (ii–iv) steps.

This larger dataset would be referred to as the *NEW* dataset in the rest of this article.

Other than *KNN* and *NEW*, which will be used as the benchmark to evaluate the proposed method based on cross-fold validation, it is also important to test the generalization power of the proposed method. This can be done by training the model on one dataset, and testing it on an independent and non-overlapping dataset, to avoid being overfitted on a particular dataset. Thus, the third dataset, the benchmark dataset from (Roy *et al.*, 2012), is used for cross-dataset validation. This non-homologous dataset was collected from PDB, satisfying two requirements: (i) the pair-wise sequence similarity within the dataset is below 30%, and (ii) there is no self-BLAST hit within the dataset to ensure that there are no enzymes that are homologous to each other in this set (Roy *et al.*, 2012). All enzymes in this dataset have experimentally determined 3D structures. To avoid overlaps between the training and testing datasets, sequences contained in both our training dataset and this dataset were removed, which reduced the size of the dataset from 318 to 284. This benchmark dataset would be referred to as the *COFACTOR* dataset in the following. Table 1 summarizes the three datasets.

3.2 Sequence representation

The deep learning framework explained in Section 3.3 eliminates the necessity of performing manual dimensionality uniformization and building complex, manually designed features, which are unlikely to sustain the increasing amount and complexity of data, by conducting feature reconstruction and classifier training simultaneously. Therefore, we use the following raw features, constructed from the input sequence directly, to represent the sequences. Based on their dimensionality, they can be classified into two categories, sequence-length-dependent features and sequence-length-independent features. The first four features described below belong to the former while the last one belongs to the latter.

3.2.1 Sequence one-hot encoding

To preserve the original sequence information, we use one-hot encoding as the first raw representation of the input sequence. This encoding uses one 1 and nineteen 0 s to represent each amino acid. For example, A is encoded as $(1\ 0_1\ \ldots\ 0_{19})$, while C is encoded as $(0_1\ 1\ 0_2\ \ldots\ 0_{19})$. For each input protein sequence, the one-hot encoding would produce an L by 20 matrix, where L represents the sequence length, with each row representing a specific spot and each column representing the appearance of a certain amino acid. For those sequences with undetermined amino acid at a particular spot, a vector with 20 0s is used to represent that special position.

3.2.2 Position specific scoring matrix

To provide the evolutional information to the training model, we deploy PSSM as the second sequence representation, which was obtained through PSI-BLAST (Altschul *et al.*, 1997) from BLAST+ (Camacho *et al.*, 2009) with three iterations, E-value being 0.002, against SWISS-PROT (released on May 11, 2016).

3.2.3 Solvent accessibility

Solvent accessibility describes the openness of a local region. Because such information is unavailable directly from the database, we use DeepCNF (Wang *et al.*, 2016b) to predict it. Taking the protein sequence as the input, DeepCNF outputs the possibilities of each amino acid of the sequence being in the state of buried, medium or exposed, respectively. The three states are defined by two solvent accessibility thresholds. Buried is defined as less than 10%; exposed is defined as >40%; and medium is defined within the range of 10 and 40%. This encoding produces an L by 3 matrix. More details could be referred to (Wang *et al.*, 2016b).

3.2.4 Secondary structure one-hot encoding

An amino acid could be in one of the three main secondary structure states, alpha-helix, beta-sheet and random coil, which indicate the protein's local folding information. Similar to solvent accessibility, we take advantage of DeepCNF (Wang *et al.*, 2016b) to predict the secondary structure of a given sequence, whose result is an L by 3 matrix, each row of which shows the possibility of the amino acid folding into alpha-helix, beta-sheet or random coil, respectively. The details could be referred to (Wang *et al.*, 2016 b).

3.2.5 Functional domain

Usually, a protein sequence contains one or several FunDs, which provide distinct functional and evolutional information. Pfam (Finn *et al.*, 2016) is a collection of such FunDs, each represented by an HMM. Searching against the database and encoding in the following way generates the FunD encoding used in our model.

i. Pfam has its default searching engine as HMMER (Eddy, 2011). For each protein sequence, we use HMMER, with the inclusion *E*-value threshold as 0.01, to search against Pfam (Pfam 30.0 Released on July 1, 2016), which contains 16 306 entries.
ii. We employ a 16 306D vector to encode the searching result. If the *i*th entry in the database is reported as hit, 1 appears on the corresponding position of the vector, otherwise it is 0.

As a result, the FunD encoding of a protein sequence would be:

$$\mathbf{F}_{\mathrm{FuncD}} = [\, I_1 \quad I_2 \quad \ldots \quad I_i \quad \ldots \quad I_{16\,306} \,] \tag{1}$$

where

$$I_i = \begin{cases} 1, \text{ the } i\text{th entry in Pfam reported as hit,} \\ 0, \text{ otherwise.} \end{cases} \tag{2}$$

3.3 Classification model

The enzyme function prediction problem has a tree-structured label space, which makes it a typical hierarchical classification problem. To solve this kind of problems, we propose a level-by-level prediction framework, building a model for each internal label node. The model contains two main components, namely, the problem-specific feature extractor, which is able to perform dimensionality uniformity and feature extraction, and the classifier. Such a novel, end-to-end model can perform feature selection and classifier training simultaneously in a virtuous circle, making it more likely to achieve high performance.

3.3.1 Level-by-level strategy

As have been discussed in the hierarchical classification part, because of the relative small size (22 168 data points are assigned to 58 classes until the second digit) and, even worse, the extreme imbalance property (e.g. the *NEW* dataset contains 22 168 sequences belonging to non-enzyme while only 10 sequences belonging to subclass 1.20) of the data, we choose the local classifier approach for this problem. Particularly, the level-by-level prediction strategy is used. That is, given a sequence, the trained model would firstly predict whether it is an enzyme or not. If it is an enzyme, the model will further predict the first digit, which indicates its main class. Knowing the main class, our algorithm will choose the trained model for that specific main class and further predict the second digit, that is, the subclass. Corresponding to the label hierarchy, we build one model for determining whether the input is an enzyme, one model to determine the first digit, six models to determine the second digit. This prediction strategy could be referred to Figure 1A.

3.3.2 Deep neural network model

For each level of prediction, we build the end-to-end model based on several deep neural network components. In terms of the sequence-length-dependent features, such as PSSM, we build a feature extractor exploiting the CNN component to extract convolutional features from the input map and, after that, a RNN component, comprised of long short-term memory (LSTM) cells, to extract sequential features from the output of the previous component. As for the sequence-length-independent feature, i.e. the FunD

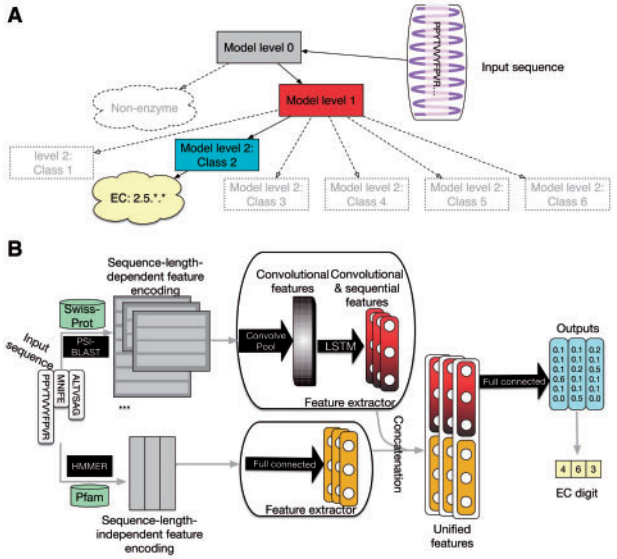

Fig. 1. (A) Strategy for predicting detailed function. Following the structure of the EC number system, we use this level-by-level classification approach. **(B)** Overview of the model. We use the CNN component to extract convolutional features and the RNN component to extract sequential features from each sequence-length-dependent raw feature encoding, followed by a fully connected component, which concatenates all extracted features together, serving as the classifier. Here we show the procedure of predicting the main class digit of three enzymes with different sequence lengths

encoding, which is a vector, we use a fully connected component to perform dimensionality reduction and feature extraction. We employ a fully connected component to combine those different pieces of information together, followed by a softmax layer for classification. The structure of the model could be referred to Figure 1B. More details of the model structure could be referred to Supplementary Section S1. It should be noted that the model we evaluate throughout the paper except for Section 4.4 uses only three input features, sequence one-hot encoding, PSSM and FunD encoding. We encode local features, i.e. secondary structure and solvent accessibility, only in Section 4.4 to evaluate the importance of local information. During training, the training error is back-propagated to each component. The error would guide the CNN component and RNN component to perform an end-to-end feature selection, weighing more on the features which would improve the final performance while weighing less on unimportant features automatically. At the same time, the weights of other components would be adjusted simultaneously to adopt the change. Such coupling effect of feature extraction and classifier training optimizes the performance dramatically.

The high complexity and flexibility of the proposed model bring high risks of overfitting. We adopt several methods to alleviate the problem. The first method is weight decay, which is a well-known method to handle the overfitting issue. The deep neural network model is likely to reproduce the detail of noise, which is usually non-smooth, by the usage of extreme weights. Modifying the objective function by adding an L-2 norm term of weights, we can reduce the probability of ending up with extreme weights and thus mitigate the overfitting issue. The second method is dropout (Srivastava *et al.*, 2014). The key idea of this technique is to randomly drop nodes during training so as to prevent them from co-adapting too much and reduce the model complexity while preserving its power. The third method is batch normalization (Ioffe and Szegedy, 2015). This approach extends the idea of normalization, which is usually done

during data pre-processing. In fact, because of the weight and parameter adjustment, the input of the internal layers of the model is possible to be too large or too small, known as 'internal covariate shift', which makes the preprocessing normalization meaningless. To conquer the issue, in addition to normalizing the data before inputting them in the model, we also normalize the input of each internal layer. In addition to the advantage of mitigating the overfitting problem, this manipulation would also reduce the strong dependency of knowledge-intensive initialization when training the model and allow a larger learning rate when tuning the model.

We choose adaptive moment estimation (Adam) as the optimizer (Kingma and Ba, 2014), which is an improved version of stochastic gradient descent, to minimize the weighted cross entropy loss. In this way, our method could handle the class imbalance issue by re-scaling predictions of each class by its weight. Instead of setting the learning rate as a hyper-parameter manually as in stochastic gradient descent and momentum, this method computes the adaptive learning rate of each individual parameter by estimating the first and second movement of the gradients at the cost of computational time and memory. Essentially, this optimizer combines the advantage of RMSprop (Tieleman and Hinton, 2012), which computes the adaptive learning rate during each step, and momentum, which reduces the oscillation problem of stochastic gradient descent by making the weight update considering both the gradient and the update of the previous step.

When training the second-digit prediction models, we adopt an idea that is similar to transfer learning. Since the limited number of data is further divided into six parts corresponding to the six main classes, the amount of data belonging to each main class is insufficient to produce a model with the ability to extract features and being generalized well. To solve this problem, we pre-train the CNN component and the RNN component by using all the training data. Then for training each second-digit prediction model, we fix the parameters of those components and only fine tune those fully connected components using the specific subset of the training data.

In practice, we use TensorFlow (Abadi, 2016) as the framework to construct the deep neural network. With two Pascal Titan X cards, it takes around 4 h to obtain a well-trained model. In Supplementary Section S2, we provide details on setting the model parameters.

4 Results and discussion

4.1 Evaluation criteria

For the enzyme or non-enzyme prediction, since it is a binary classification problem, we use accuracy, Cohne's Kappa Score (Viera and Garrett, 2005), precision, recall and F_1 score to evaluate the classifiers' performance. For other predictions, since they are multi-class classification problems, we use accuracy, Cohen's Kappa Score, Macro-precision, Macro-recall and Macro-F_1 score to evaluate the classifiers' performance, whose definitions are in Supplementary Section S3.

4.2 Compared methods

For the cross-fold validation, in which training and testing are based on different parts within the same dataset, we compare our method with five other methods, including two state-of-the-art methods, EzyPred (Shen and Chou, 2007) and SVM-Prot (Li *et al.*, 2016), and three baseline methods. One of the baseline methods uses SVM with the raw features used in our model; another baseline method uses SVM with Pse-PSSM; and the last baseline method uses the traditional neural network with our raw features. Due to the unchangeable database of EFICAz (Kumar and Skolnick, 2012) and COFACTOR (Zhang *et al.*, 2017), we do not include them in the cross-fold validation comparison. However, we perform cross-dataset validation, where the training and testing are performed on different datasets, to compare our method with EzyPred, SVM-Prot, COFACTOR and EFICAz.

4.3 Cross-fold validation

Here we report the 5-fold cross validation results, which are shown in Figure 2. Our method almost always outperforms the other methods in both the *KNN* dataset and the *NEW* dataset across the five criteria and across the three hierarchical levels of prediction. As for the *NEW* dataset, DEEPre outperforms the other five methods consistently in Levels 0 and 1 prediction across the five criteria. As for the Level 2 prediction, the only criterion that DEEPre does not improve over the existing methods is the Macro-Precision, which is an unweighted average of precision of each label. The appearance of very small classes (e.g. subclass 1.20 only has 10 enzymes) in the second level prediction might be the reason for this result. In terms of the *KNN* dataset, although the smaller dataset makes the improvement of DEEPre over the other methods in Level 0 prediction less significant, it still significantly outperforms the other methods in Levels 1 and 2 classification.

4.4 Feature importance analysis

It is believed that both global features and local features determine the function of a protein. For detailed function, local information would weigh even more in determining it. The features extracted by the convolutional component and the recurrent component from PSSM and sequence raw encoding could be considered as global features while the FunD encoding would be considered as a local feature. We remove the three input raw encoding one by one and show the comparison of their performance on the *NEW* dataset. The comparison is shown in Figure 3A. It is clear that as the level goes deeper, the importance of FunD is evidently increasing, which

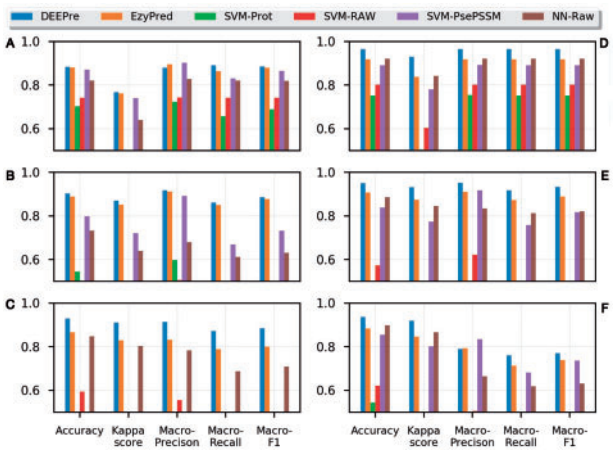

Fig. 2. Cross-fold validation results. **(A)** Performance comparison of Level 0 prediction (predicting whether the input is an enzyme or not) on the *KNN* dataset. **(B)** Performance comparison of Level 1 prediction (predicting the input enzyme's main class) on the *KNN* dataset. **(C)** Performance comparison of Level 2 prediction (predicting the input enzyme's subclass given the main class) on the *KNN* dataset. **(D)** Performance comparison of Level 0 prediction on the *NEW* dataset. **(E)** Performance comparison of Level 1 prediction on the *NEW* dataset. **(F)** Performance comparison of Level 2 prediction on the *NEW* dataset

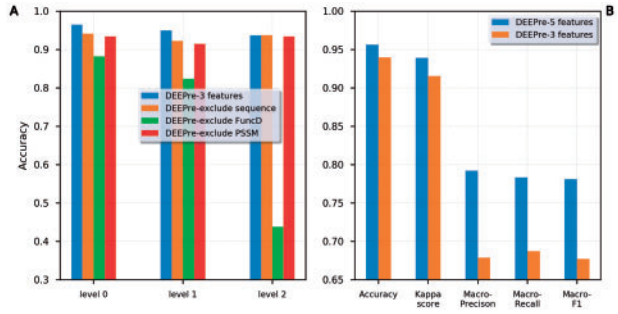

Fig. 3. **(A)** Feature contribution investigation considering sequence one-hot encoding (sequence), PSSM and FunD. **(B)** The performance change of the model before and after we input more local feature encoding. Macro-precision, Macro-recall and Macro-F1 score are improved by at least 11% by inputting solvent accessibility and secondary structure information

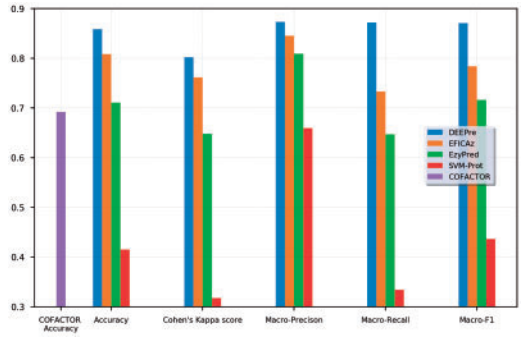

Fig. 4. The performance comparison of different servers on predicting the main class of the *COFACTOR* dataset. DEEPre improves the prediction accuracy over the other servers by at least 6%

demonstrates the well-recognized hypothesis. To further prove it, we design another experiment, in which we input more local feature encodings, including secondary structure and solvent accessibility, into our model. Details of this experiment could be referred to Supplementary Section S4. Figure 3B shows the performance comparison of this model and the previous model in Level 2 prediction. It is clear that the additional local features further improve the performance of our model, with accuracy improved by 1.8% while Macro-precision, Macro-recall and Macro-F1 score improved by at least 11%.

4.5 Cross-dataset validation

In this experiment, we directly compare the performance of different servers in predicting the first digit and the second digit of an enzyme. We use the *COFACTOR* benchmark dataset, which is proved to be a difficult dataset in the enzyme function prediction field (Roy *et al.*, 2012), as the test dataset. First, we eliminate the sequences in the COFACTOR benchmark data which overlap with the DEEPre's training database (*NEW*) by 40% sequence similarity filtering, reducing the data size from 318 to 284, to ensure that there is no bias in the DEEPre's results. Then we input the remaining sequences to each server manually and collect the prediction results. For COFACTOR, since it is quite time-consuming to run the server, about 4 h to obtain the result for one query, we report the results from the original paper. As shown in Figure 4, for the first-digit prediction, DEEPre outperforms the other servers consistently across the five criteria, improving the accuracy by at least 6% over the other servers, including COFACTOR. This is significant because COFACTOR requires 3D structures of enzymes whereas DEEPre only requires the sequence information. On the other hand, we should admit that we have changed the original *COFACTOR* dataset to some extent by reducing the overlap between the training and testing sets, which might explain some of the performance difference between COFACTOR and DEEPre. We should also notice that all of those five servers have different training datasets, but those training datasets highly overlap with each other and each method was optimized on its corresponding dataset. In addition, although we removed overlapping enzymes from our training set and the *COFACTOR* test set, there may still be homologs of the enzymes in *COFACTOR* in our training set. However, DEEPre is a sequence-based statistical method, which explores the statistical properties of training data and does not benefit from knowing enzyme structures. Therefore, although the performance of DEEPre on *COFACTOR* may still be biased, the influence is not as much as that by nearest neighbor-based methods. We also perform comparison of different servers' performance on the second-digit prediction (Supplementary Section S5). The results show that DEEPre and EFICAz both perform well on the second-digit prediction on the *COFACTOR* dataset. It is worth noting that EC numbers have regular corrections, such as deletions and transfers. We check all the corrections that are related to the test enzymes in the *COFACTOR* dataset and find that none of them influences the comparison reported here.

4.6 Third-digit and fourth-digit prediction

Using the same framework described above, we are also able to predict the enzyme's third digit, which represents its sub-subclass, on the *NEW* dataset. The accuracy across all the sub-subclasses is 0.9415; the Kappa score is 0.8918; the macro-precision is 0.8942; the macro-recall is 0.8578; and the macro-F1 score is 0.8665. Regarding the fourth-digit prediction, more data are needed to perform normal machine learning training-and-testing procedure. For example, within the sub-subclass 1.1.1 in the *NEW* dataset, there are 188 classes. Each of those classes has <40 enzyme sequences, with 175 classes having <10 enzyme sequences. Using the current dataset with such distribution would lead to unreliable results.

4.7 Case study

Glutaminase is a phosphate-activated enzyme, which catalyzes the first step of glutaminolysis, hydrolysing glutamine into glutamate (Curthoys and Watford, 1995). The alternative splicing of its messager RNA results in its three isoforms, with Isoforms 1 and 3 being capable of catalyzing while Isoform 2 lacking the catalytic activity (Li *et al.*, 2017). To validate our model's ability to distinguish the different functionality of different isoforms, we obtained the sequences of the three Glutaminase isoforms from the UniProt and put them into our model. Our model predicted that Isoforms 1 and 3 of Glutaminase were hydrolases acting on carbon-nitrogen bonds, being consistent with the experimental results. Our model also recognized Isoform 2 as non-enzyme, which is consistent with the experimental result as well.

Aurora kinases B is a key enzyme regulating chromosomal segregation during mitosis, ensuring correct chromosome alignment and segregation as well as chromatin-induced microtubule stabilization and spindle assembly (Carmena *et al.*, 2009). Over-expression of it is possible to cause unequal distribution of genetic information, resulting in aneuploid cells, which may become cancerous (Sorrentino *et al.*, 2005). Aurora kinases B has five isoforms resulted from alternative splicing. Four of them have roughly equal length with high similarity, while Isoform 3, having high expression in the metastatic

liver with no expression in the normal liver, is only half of the length of the" canonical" isoform (142 amino acids versus 344 amino acids). Despite its much shorter length, the isoform does not lose its functionality. To further validate our model's ability of handling isoforms' functionality prediction, we collected the sequence of the five isoforms from the database and put them into our model. Our model's result is consistent with the experimental results. Particularly, our model predicted the functionality of the Isoform 3 successfully, despite its sequence's large difference from the 'canonical' sequence.

The detailed performance comparison of different servers on these two case studies could be referred to Supplementary Section S6. Among the five compared methods, only our method and EzyPred produced correct predictions for both cases.

5 Conclusion

In this article, we proposed a novel end-to-end feature extraction and classifier training method for enzyme function prediction. The method proposed in this paper would force the model to learn to extract features by itself and adapt the parameters of the classifier simultaneously so that it can improve the performance in a virtuous circle. The thorough experiments conducted on two datasets demonstrate the high performance of our method in both a smaller dataset from 10 years ago and a larger dataset constructed half a year ago. The cross-dataset validation experiment proves the performance of our model in handling sequences with no close homologs. Although it is just a starting point, the user-friendly server, DEEPre, will provide users a good guess of enzyme function and help them set up downstream experiments. Since DEEPre predicts a score for each candidate value of a certain EC digit, it can be potentially used to detect the enzyme promiscuity (Carbonell and Faulon, 2010; Mellor *et al.*, 2016), which means that some enzymes show multiple activities by either accepting multiple substrates or catalyzing multiple reactions. Our webserver provides the predicted scores for all candidate EC values. In addition to providing the server in the enzyme function prediction field, we believe the idea proposed in this paper can be quite helpful in handling the feature length nonuniformity problem and the dataset evolvement in a wide spectrum of computational biology problems.

Among the global features, the most important one is the FunD (Fig. 3A). This sequence-length-independent feature cannot be replaced by sequence-length-dependent features. Nevertheless, the global and local features explored in this paper provide complementary information and together provide improved performance (Fig. 3B), in spite of leading to the higher dimensionality of the predictor.

A large number of protein function prediction problems are hierarchical classification problems, such as GO term (Camon *et al.*, 2004), transporter classification (Saier *et al.*, 2016) and G-protein-coupled receptors (GPCR) hierarchy (Davies *et al.*, 2007). The high extensibility and flexibility of our level-by-level prediction framework make it possible to adopt our framework in those problems. Furthermore, the robust, automatic framework based on deep learning to extract problem-specific sequence-length-independent features from the sequence-length-dependent features can also be extended to other features in addition to the features mentioned in this article.

There are two directions of the future work. First, more robust methods for the fourth-digit prediction are needed. The increasing number of enzymes that have experimentally validated functions, as well as the advance in method development for learning from imbalanced-data and small samples (Maadooliat *et al.*, 2016), provide potential solutions to the problem. Second, instead of predicting the EC numbers for enzymes, it is practically useful to predict enzymatic reactions of the enzymes. The use of reaction fingerprints, for instance, could be one viable solution for this (Segler and Waller, 2017). Another possible solution is through the use of descriptors of the reaction centers as in (Rahman *et al.*, 2014).

Acknowledgements

We would like to thank Prof. Kuo-Chen Chou for kindly providing the *KNN* dataset.

Funding

This work was supported by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR) under Award No URF/1/1976-04 and URF/1/3007-01, National Natural Science Foundation of China (61401131 and 61731008).

Conflict of Interest: none declared.

References

Abadi, M. (2016) Tensorflow: Learning functions at scale. *Acm Sigplan. Notices*, **51**, 1–1.

Alipanahi, B. *et al.* (2015) Predicting the sequence specificities of dna- and rna-binding proteins by deep learning. *Nat. Biotechnol.*, **33**, 831–838.

Altschul, S.F. *et al.* (1997) Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.

Arakaki, A.K. *et al.* (2009) Eficaz2: enzyme function inference by a combined approach enhanced by machine learning. *BMC Bioinformatics*, **10**, 107.

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

Cai, C.Z. *et al.* (2003) Svm-prot: Web-based support vector machine software for functional classification of a protein from its primary sequence. *Nucleic Acids Res.*, **31**, 3692–3697.

Cai, C.Z. *et al.* (2004) Enzyme family classification by support vector machines. *Proteins*, **55**, 66–76.

Cai, Y.D. and Chou, K.C. (2005) Predicting enzyme subclass by functional domain composition and pseudo amino acid composition. *J. Proteome Res.*, **4**, 967–971.

Cai, Y.D. *et al.* (2005) Predicting enzyme family classes by hybridizing gene product composition and pseudo-amino acid composition. *J. Theor. Biol.*, **234**, 145–149.

Camacho, C. *et al.* (2009) Blast+: architecture and applications. *BMC Bioinformatics*, **10**, (1), 421.

Camon, E. *et al.* (2004) The gene ontology annotation (goa) database: sharing knowledge in uniprot with gene ontology. *Nucleic Acids Res.*, **32**, 262D–2626. D262. –

Carbonell, P. and Faulon, J.-L. (2010) Molecular signatures-based prediction of enzyme promiscuity. *Bioinformatics (Oxford, England)*, **26**, 2012–2019.

Carmena, M. *et al.* (2009) Making the auroras glow: regulation of aurora a and b kinase function by interacting proteins. *Curr. Opin. Cell Biol.*, **21**, 796–805.

Chen, P. *et al.* (2013) Accurate prediction of hot spot residues through physicochemical characteristics of amino acid sequences. *Proteins*, **81**, 1351–1362.

Chen, P. *et al.* (2014) Ligandrfs: random forest ensemble to identify ligand-binding residues from sequence information alone. *BMC Bioinformatics*, **15**, S4.

Chen, P. *et al.* (2016) A sequence-based dynamic ensemble learning system for protein ligand-binding site prediction. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, **13**, 901–912.

Chou, K.C. (2005) Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes. *Bioinformatics*, **21**, 10–19.

Chou, K.C. (2009) Pseudo amino acid composition and its applications in bioinformatics, proteomics and system biology. *Curr. Proteomics*, **6**, 262–274.

Chou, K.C. (2011) Some remarks on protein attribute prediction and pseudo amino acid composition. *J. Theor. Biol.*, **273**, 236–247.

Chou, K.C. and Elrod, D.W. (2003) Prediction of enzyme family classes. *J. Proteome Res.*, **2**, 183–190.

Cornish-Bowden, A. (2014) Current iubmb recommendations on enzyme nomenclature and kinetics. *Perspect. Sci.*, **1**, 74–87.

Curthoys, N.P. and Watford, M. (1995) Regulation of glutaminase activity and glutamine metabolism. *Annu. Rev. Nutr.*, **15**, (1), 133–159.

Dai, H. *et al.* (2017) Sequence2vec: a novel embedding approach for modeling transcription factor binding affinity landscape. *Bioinformatics*, doi: 10.1093/bioinformatics/btx480.

Davies, M.N. *et al.* (2007) On the hierarchical classification of g protein-coupled receptors. *Bioinformatics*, **23**, 3113–3118.

De Ferrari, L. *et al.* (2012) Enzml: multi-label prediction of enzyme classes using interpro signatures. *BMC Bioinformatics*, **13**, 61.

Des Jardins, M. *et al.* (1997) Prediction of enzyme classification from protein sequence without the use of sequence similarity. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, **5**, 92–99.

Dobson, P.D. and Doig, A.J. (2005) Predicting enzyme class from protein structure without alignments. *J. Mol. Biol.*, **345**, 187–199.

Eddy, S.R. (2011) Accelerated profile hmm searches. *PLoS Comput. Biol.*, **7**, e1002195.

Finn, R.D. *et al.* (2016) The pfam protein families database: towards a more sustainable future. *Nucleic Acids Res.*, **44**, D279–D285.

Fu, L. *et al.* (2012) Cd-hit: accelerated for clustering the next-generation sequencing data. *Bioinformatics*, **28**, 3150–3152.

Goddard, J.P. and Reymond, J.L. (2004) Enzyme assays for high-throughput screening. *Curr. Opin. Biotechnol.*, **15**, 314–322.

Hayat, M. and Khan, A. (2012) Discriminating outer membrane proteins with fuzzy k-nearest neighbor algorithms based on the general form of chou's pseaac. *Protein Pept. Lett.*, **19**, 411–421.

Hoffmann, B. *et al.* (2007) Nature and prevalence of pain in fabry disease and its response to enzyme replacement therapy-a retrospective analysis from the fabry outcome survey. *Clin. J. Pain*, **23**, 535.

Huang, W.L. *et al.* (2007) Accurate prediction of enzyme subfamily class using an adaptive fuzzy k-nearest neighbor method. *Biosystems*, **90**, 405–413.

Ioffe, S. and Szegedy, C. (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, **37**, 448–456.

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

Krizhevsky, A. *et al.* (2012) Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems 25*, pp. 1097–1105.

Kumar, C. and Choudhary, A. (2012) A top-down approach to classify enzyme functional classes and sub-classes using random forest. *EURASIP J. Bioinform. Syst. Biol.gy*, **2012**, 1–14.

Kumar, N. and Skolnick, J. (2012) Eficaz2.5: application of a high-precision enzyme function predictor to 396 proteomes. *Bioinformatics*, **28**, 2687–2688.

Lee, B.J. *et al.* (2008) Design of a novel protein feature and enzyme function classification. In: *8th IEEE International Conference on Computer and Information Technology Workshops: Cit Workshops 2008, Design of a Novel Protein Feature and Enzyme Function Classification*, pp. 450–455.

Leslie, C. *et al.* (2002) The spectrum kernel: a string kernel for SVM protein classification. In: *Proceedings of the Pacific Symposium on Biocomputing*, pp. 564–575, Singapore. World Scientific Publishing.

Leslie, C.S. *et al.* (2004) Mismatch string kernels for discriminative protein classification. *Bioinformatics*, **20**, 467–476.

Li, Y. *et al.* (2017) Serial deletion reveals structural basis and stability for the core enzyme activity of human glutaminase 1 isoforms: relevance to excitotoxic neurodegeneration. *Transl. Neurodegener.*, **6**, 10.

Li, Y.H. *et al.* (2016) Svm-prot 2016: a web-server for machine learning prediction of protein functional families from sequence irrespective of similarity. *PLoS One*, **11**, e0155290.

Lu, L. *et al.* (2007) Ecs: an automatic enzyme classifier based on functional domain composition. *Comput. Biol. Chem.*, **31**, 226–232.

Maadooliat, M. *et al.* (2016) Collective estimation of multiple bivariate density functions with application to angular-sampling-based protein loop modeling. *J. Am. Stat. Assoc.*, **111**, 43–56.

Mellor, J. *et al.* (2016) Semisupervised gaussian process for automated enzyme search. *ACS Synth. Biol.*, **5**, 518–528.

Moult, J. *et al.* (2011) Critical assessment of methods of protein structure prediction (casp)-round ix. *Proteins*, **79**, 1–5.

Nagao, C. *et al.* (2014) Prediction of detailed enzyme functions and identification of specificity determining residues by random forests. *PLoS One*, **9**, e84623.

Nasibov, E. and Kandemir-Cavas, C. (2009) Efficiency analysis of knn and minimum distance-based classifiers in enzyme family prediction. *Comput. Biol. Chem.*, **33**, 461–464.

Qiu, J.D. *et al.* (2009) Using support vector machines to distinguish enzymes: Approached by incorporating wavelet transform. *J. Theor. Biol.*, **256**, 625–631.

Qiu, J.D. *et al.* (2010) Using the concept of chou's pseudo amino acid composition to predict enzyme family classes: An approach with support vector machine based on discrete wavelet transform. *Protein Pept. Lett.*, **17**, 715–722.

Quester, S. and Schomburg, D. (2011) Enzymedetector: an integrated enzyme function prediction tool and database. *BMC Bioinformatics*, **12**, 376.

Rahman, S.A. *et al.* (2014) Ec-blast: a tool to automatically search and compare enzyme reactions. *Nat. Methods*, **11**, 171–174.

Rätsch, G. *et al.* (2005) RASE: recognition of alternatively spliced exons in *C.elegans*. *Bioinformatics*, **21** (Suppl 1), i369–i377.

Roy, A. *et al.* (2012) Cofactor: an accurate comparative algorithm for structure-based protein function annotation. *Nucleic Acids Res.*, **40**, W471–W477.

Saier, M.H.J. *et al.* (2016) The transporter classification database (tcdb): recent advances. *Nucleic Acids Res.*, **44**, D372–D379.

Segler, M.H.S. and Waller, M.P. (2017) Neural-symbolic machine learning for retrosynthesis and reaction prediction. *Chemistry*, **23**, 5966–5971.

Sharif, M.M. *et al.* (2015) Enzyme function classification based on sequence alignment. *Inform. Syst. Des. Intell. Appl.*, **340**, 409–418.

Shen, H.B. and Chou, K.C. (2007) Ezypred: a top-down approach for predicting enzyme functional classes and subclasses. *Biochem. Biophys. Res. Commun.*, **364**, 53–59.

Silla, C.N. and Freitas, A.A. (2011) A survey of hierarchical classification across different application domains. *Data Mining Knowl. Discov.*, **22**, 31–72.

Sorrentino, R. *et al.* (2005) Aurora b overexpression associates with the thyroid carcinoma undifferentiated phenotype and is required for thyroid carcinoma cell proliferation. *J. Clin. Endocrinol. Metab.*, **90**, 928–935.

Srivastava, N. *et al.* (2014) Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, **15**, 1929–1958.

Tian, W. *et al.* (2004) Eficaz: a comprehensive approach for accurate genome-scale enzyme function inference. *Nucleic Acids Res.*, **32**, 6226–6239.

Tieleman, T. and Hinton, G. (2012) Lecture 6.5-rmsprop: divide the gradient by a running average of its recent magnitude. *COURSERA*, **4**,

Viera, A.J. and Garrett, J.M. (2005) Understanding interobserver agreement: the kappa statistic. *Fam. Med.*, **37**, 360–363.

Volpato, V. *et al.* (2013) Accurate prediction of protein enzymatic class by n-to-1 neural networks. *BMC Bioinformatics*, **14**, 1–7.

Wang, F. *et al.* (2016a) Deeppicker: a deep learning approach for fully automated particle picking in cryo-em. *J. Struct. Biol.*, **195**, 325–336.

Wang, S. *et al.* (2016b) Protein secondary structure prediction using deep convolutional neural fields. *Sci. Rep.*, **6**, 18962.

Wang, X. *et al.* (2014) Modeling DNA affinity landscape through two-round support vector regression with weighted degree kernels. *BMC Syst. Biol.*, **8**, S5.

Wang, Y.C. *et al.* (2010) Prediction of enzyme subfamily class via pseudo amino acid composition by incorporating the conjoint triad feature. *Protein Pept. Lett.*, **17**, 1441–1449.

Wang, Y.C. *et al.* (2011) Support vector machine prediction of enzyme function with conjoint triad feature and hierarchical context. *BMC Syst. Biol.*, **5**, S6.

Wood, R.D. *et al.* (2001) Human dna repair genes. *Science*, **291**, 1284–1289.

Xiong, H.Y. *et al.* (2015) Rna splicing. the human splicing code reveals new insights into the genetic determinants of disease. *Science*, **347**, 1254806.

Yang, J. *et al.* (2015) The i-tasser suite: protein structure and function prediction. *Nat. Methods*, **12**, 7–8.

Yu, C. *et al*. (2009) Genome-wide enzyme annotation with precision control: catalytic families (catfam) databases. *Proteins*, **74**, 449–460.

Zhang, C. *et al*. (2017) Cofactor: improved protein function prediction by combining structure, sequence and protein-protein interaction information. *Nucleic Acids Res*., page gkx366.

Zhou, X.B. *et al*. (2007) Using chou's amphiphilic pseudo-amino acid composition and support vector machine for prediction of enzyme subfamily classes. *J. Theor. Biol*., **248**, 546–551.

Zou, H.L. and Xiao, X. (2016) Classifying multifunctional enzymes by incorporating three different models into chou's general pseudo amino acid composition. *J. Membr. Biol*., **249**, 551–557.