Received: 19 March 2019 | Revised: 5 July 2019 | Accepted: 17 September 2019

DOI: 10.1002/prot.25832

PROTEINS WILEY

REVIEW

Machine learning techniques for protein function prediction

Rosalin Bonetta[1] | Gianluca Valentino[2]

[1]Centre for Molecular Medicine and Biobanking, University of Malta, Msida, Malta

[2]Department of Communications and Computer Engineering, University of Malta, Msida, Malta

Correspondence
Rosalin Bonetta, Centre for Molecular Medicine and Biobanking, University of Malta, Msida MSD2080, Malta.
Email: rosalin.bonetta@um.edu.mt

Peer Review
The peer review history for this article is available at https://publons.com/publon/10.1002/prot.25832.

Abstract

Proteins play important roles in living organisms, and their function is directly linked with their structure. Due to the growing gap between the number of proteins being discovered and their functional characterization (in particular as a result of experimental limitations), reliable prediction of protein function through computational means has become crucial. This paper reviews the machine learning techniques used in the literature, following their evolution from simple algorithms such as logistic regression to more advanced methods like support vector machines and modern deep neural networks. Hyperparameter optimization methods adopted to boost prediction performance are presented. In parallel, the metamorphosis in the features used by these algorithms from classical physicochemical properties and amino acid composition, up to text-derived features from biomedical literature and learned feature representations using autoencoders, together with feature selection and dimensionality reduction techniques, are also reviewed. The success stories in the application of these techniques to both general and specific protein function prediction are discussed.

KEYWORDS
deep learning, feature selection, machine learning, protein function prediction

1 | INTRODUCTION

Proteins are made up from 20 different types of amino acids, which occur in nature and are encoded by DNA sequences. Proteins perform essential roles in the cells of organisms. These include cell signaling, regulation, recognition, catalysis of reactions, membrane transport, and the provision of structure. The function performed by a protein depends on its structure, which is indirectly, a result of its DNA sequence.

A classical view of protein function focuses on the action of a single protein molecule. For example, the catalysis of a given reaction or the binding of a molecule, which may be small or large. Today this local function is occasionally termed the "molecular function" of the protein, such as to distinguish it from the expanded view of function (Figure 1). A protein is defined as an element in the network of its interactions in the case of an expanded view of protein function. Numerous terms such as "contextual function" or "cellular function," have been coined for this expanded view of function.[2] The idea conveyed is that each protein plays a role in an extended network of interacting molecules. Therefore, a function can be thought of as "anything that happens to or through a protein".[3]

The extent to which a protein's function is altered upon mutating an amino acid depends on the type and position of the amino acid that is mutated, for example whether the amino acid is found in an enzyme active site. Thus, numerous mutations may affect protein function in a complicated manner, and are therefore, difficult to predict. Due to limitations imposed by experimental methods,[4] predicting protein function by computational means has become crucial. Protein functions can be described at different levels of complexity, which include cellular, biochemical, physiological, and phenotypic levels. In addition, protein function may be defined in a hierarchical manner. For instance, at a high level, superoxide dismutase is an oxidoreductase, while at a lower level, it converts superoxide radicals into hydrogen peroxide and molecular oxygen.

Gene Ontology (GO) terms offer an accurate description of the several levels of protein function.[5] It is vital to comprehend that the molecular or biochemical function of a protein is demonstrated via

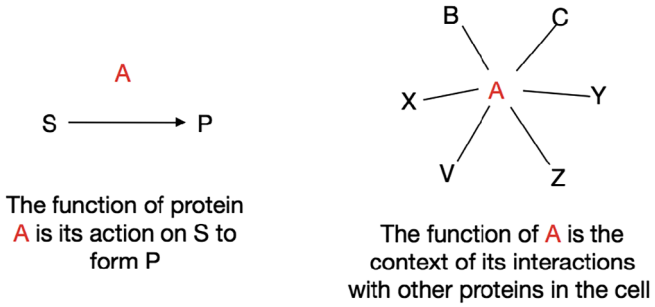

FIGURE 1 The evolution of the meaning of protein function. The traditional view is illustrated on the left, and the post-genomic view on the right. Adapted from Reference 1 [Color figure can be viewed at wileyonlinelibrary.com]

sequence and/or structural data. Therefore, in silico approaches can aid in the prediction of protein function.[6] As discussed by Lee et al, there are different interdependent levels of protein function, which may be divided into three major types of GO categories: molecular function, biological process, and cellular component (Figure 2).[7] Molecular function refers to activity at the molecular level (eg, catalysis), and is commonly predicted through computational methods, which identify homologues or orthologues. Biological process describes broader functions, which are performed by assemblies of molecular functions, such as a particular metabolic pathway. Genomic inference methods can identify the direct physical protein-protein interactions and indirect functional associations, which are found in biological processes. Finally, cellular component describes the location(s) within a cell in which the protein performs its function. Prediction of protein subcellular localization is an important component of bioinformatics based prediction of protein function and genome annotation, as it can aid the identification of drug targets.[8] This component can be predicted through methods that predict signal sequences, residue composition, membrane association, or post-translational modifications.

Protein information is stored in several databases, such as UniProt,[9] which is the leading protein sequence database or Pfam, which is a database of protein function families, for which the protein sequence is known but the function is unknown.[10] The gap between the amount of protein sequences and the functional annotations has been growing continuously (Figure 3). There is an order of magnitude more of protein sequences today than 10 years ago in the UniProt Knowledgebase (UniProtKB). However, the number of manually

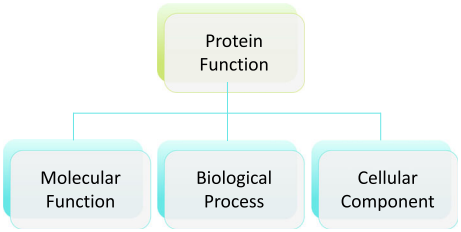

FIGURE 2 Classification of protein function according to GO: molecular function, biological process, and cellular component [Color figure can be viewed at wileyonlinelibrary.com]

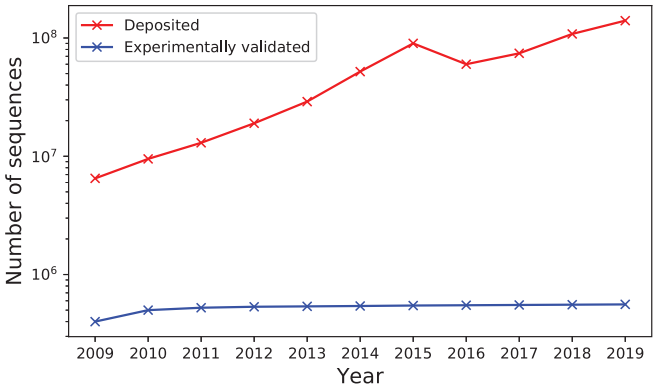

FIGURE 3 Number of sequences deposited and experimentally validated in UniProtKB over the past decade. The drop observed between 2015 and 2016 is due to procedures deployed by curators to identify and remove redundant proteomes

annotated and reviewed protein sequences (UniProtKB/SwissProt) has only marginally increased.

Therefore, a main challenge in bioinformatics involves predicting the role played by proteins in biological processes and disease, as well as predicting mechanisms by which such functions are performed. As new algorithms are developed to address these questions, it is essential to evaluate the performance of these different function prediction algorithms with respect to more traditional, manual methods. The bioinformatics community has sought to address the problem of automated protein function prediction through initiatives such as the Critical Assessment of Function Annotation (CAFA) challenge.[11] This is an experiment designed to provide large-scale assessment of computational methods used to predict protein function.

Since more than a decade ago, researchers have used or machine learning techniques to derive sequence-function relationships. Machine learning models of protein function have shown to provide good predictive performance, even when the underlying mechanisms were not well understood. Bernardes et al documented the growing critical mass of literature in which machine learning techniques were used to predict protein function in their review paper.[12] However, following the trend in other domains, besides the use of established methods like random forests, support vector machines (SVM) and neural networks, the use of deep learning has also caught on, with impressive results. Deep learning is well suited to big data problems, and is now within reach due to the rapid evolution in computational performance. Therefore, we extend the review of the literature beyond the one performed by Bernardes et al in 2013 to include novel sources of features and deep learning approaches, among others. Other reviews have focused on specific taxonomies and ontologies, such as enzyme functional class prediction[13] and subcellular localization,[14] whereas this review is intended to be more comprehensive to cover a wide array of features and techniques which may be interchangeable across different taxonomies.

The notion of protein function and a recapitulation of the existing techniques used for function prediction were already provided in this introduction. The next part of this review presents protein function

prediction as a problem which can be targeted using machine learning techniques. These techniques range from the generation and selection of suitable features, to algorithms and models, which can be trained to perform this task. In addition, the applications of these techniques to general and specific function prediction are also discussed. This review concludes with the future perspectives for these techniques in this domain.

2 | MACHINE LEARNING TECHNIQUES FOR PROTEIN FUNCTION PREDICTION

2.1 | Feature engineering and representation

The inputs to a predictive model, which is trained using machine learning techniques pertinent to a particular object, in this case a protein, are known as features. A key step in applying machine learning to any application is identifying suitable features. This can allow the model to discriminate between one category of data and another in a classification problem, or fit a suitable function to some data in a regression problem. Generating suitable features is also known as feature engineering. A group of features representing one particular object is known as a feature vector, while the n-dimensional space associated with the feature vector is termed the feature space.

Typical protein features include amino acid sequences, physicochemical properties, and protein-protein interactions. Amino acid sequences can be used to derive parameters such as amino acid composition, which refers to the occurrence of amino acids in a particular sequence; amino acid transition, which represents the frequency with which specific amino acid types are followed or preceded by other amino acid types within the sequence; and amino acid distribution, which captures the dissemination of specific amino acid types within specific portions of the sequence. A particular category of sequence-based features is the sequence motif, which consists of an amino acid sequence pattern, which is widespread, and is thought to have a certain biological significance. Therefore, the presence or absence of a particular sequence motif can be used as a binary feature. N-terminal targeting sequences have also been used as features.[15,16] Sequence-related features such as Auto Covariance, Conjoint triad, local descriptor, and Moran autocorrelation have proved useful in mining interaction information in the sequence.[17]

Physicochemical properties of protein residues include isoelectric points, molecular weights, polarity, hydrophobicity, normalized van der Waals volume, polarity, extinction coefficients, polarizability, charge, and surface tension. Protein-protein interaction (PPI) networks are mathematical representations of the physical contacts between proteins. The linkage-based assumption,[18] also known as the guilt-by-association rule, comes from the observation that immediate neighbor proteins and level-2 neighbors have a high probability of sharing functions. Therefore a protein's function could be determined from the majority of its neighbors' functions. In addition to considering neighboring proteins, it is also common to consider the weights of the interactions, which are proportional to the reliability of the experimental sources. PPI tools such as Cytoscape,[19] provide access to further network features, such as average shortest path length, neighborhood connectivity, radiality, and the topological coefficient. Features can also be generated based on the overall Composition, Transition and Distribution (CTD) of amino acid attributes such as physicochemical properties, secondary structure, and solvent accessibility.[20] This feature vector was used to classify protein locations in cellular sorting pathway.

After introducing the basic sources, we now discuss how features can be better represented. The concept of protein granularity and the possibility of extracting features was originally proposed in Reference 21. Protein granularity captures information about sequence-order effects and amino acid composition. As machine learning algorithms can only handle vectors, the Pseudo Amino Acid Composition (PseAAC)[22] was developed to formulate an amino acid sequence of arbitrary length, such as a vector. A protein sequence with length L amino acid residues $R_1R_2R_3...R_L$, where R_1 represents the residue at sequence position 1, R_2 represents the residue at position 2 and so on, may be denoted as a $(20 + \lambda)$-dimensional vector, defined by $20 + \lambda$ discrete numbers, that is

$$X = .[x_1...x_{20}x_{20+1}...x_{20+\lambda}] \quad (1)$$

The first 20 numbers above represent the classic amino acid composition, while the next lambda discrete numbers reflect the effect of sequence order.

The position-specific scoring matrix (PSSM) was first introduced for detecting distantly related proteins.[23] The original PSSM introduced by Gribskov et al consists of the following components: (a) position: indicates the sequentially increased index of each amino acid residue in a sequence after multiple sequence alignment; (b) probe: a group of typical sequences of functionally related proteins that have been aligned by sequence or structural similarity; (c) profile: a matrix consisting of 20 columns corresponding to 20 amino acids; (d) consensus: a sequence of amino acid residues that are closest to all of the alignment residues of probes at each position. It is generated by selecting the highest score in the profile at each position.

Therefore, a PSSM for a given protein consists of a $N \times 20$ matrix, where N is the length of the protein sequence. It assigns a score P_{ij} for the j^{th} amino acid in the i^{th} position of the query sequence with a large value indicating a highly conserved position, and a small value indicating a weakly conserved position. However, as machine learning algorithms typically require a fixed input size, the PSSMs need to be processed further.

A systematic study of three different feature sets extracted using PSSM was performed by Jeong et al.[24] The first feature set consisted of the averaged PSSM profiles over blocks, each with 5% of a sequence. A protein sequence, regardless of length, is divided into two blocks and each block consists of 20 features derived from the 20 columns in the PSSMs. In the second feature set, instead of considering the locations of domains in a sequence, the authors focused on the domains with similar conservation rates. In the third feature set, the physicochemical properties of probed residues using original protein sequences were considered. A total of nine physicochemical

properties, were categorized into two groups such as average and density groups. Hydrophobicity, isoelectric point, and mess scale were averaged, while hydrophobic, hydrophilic, polar, nonpolar, positive, and negative charge residues were used for calculating densities. Following training using machine learning models such as SVMs, Random Forests, and decision trees, the second feature set was found to be the most effective in protein function prediction. In Reference 25, the authors used protein granularity as one of the input features.

Machine learning algorithms generally require numerical features in order to develop a suitable model. While this is straightforward for most sequence-, physiochemical- and PPI-derived features, it is also possible to use text-based features if these are converted into a numerical format. Advances in Natural Language Processing (NLP) techniques have resulted in a greater exploitation of text-based features for protein function prediction from biomedical literature, such as abstracts or full-texts of journal articles.[26] NLP techniques are also well suited due to the nature of data storage in biological and biochemical databases.[27]

These techniques were previously used in representing amino acid sequences as text, and extracting features such as n-grams and term frequency-inverse document frequency (TFIDF). An n-gram is a contiguous sequence of n items from a sequential dataset, such as a protein sequence. In TFIDF, each document is represented by a vector of all terms in a controlled vocabulary. For each term in a document, a weight is calculated as the product of the TF and IDF, where TF is its frequency in this document, and IDF is its inverse document frequency in the full dataset of documents. The basic idea of TFIDF is to emphasize the terms with more occurrences in a document and less occurrences (more discriminable) in the document dataset. Another representation is document-to-vector, which is a dense, semantic representation for documents.[28] In NLP, text features are represented using vectors and techniques such as Word2Vec.[29] Asgari and Mofrad describe how they developed ProtVec to represent amino acid sequences.[30] Text mining applied to bioinformatics literature has been shown to be particularly useful in extracting protein-protein interactions and extracting the relationship between gene functions and diseases.

In the specific case of cellular component prediction, immunohistochemistry images have also been used as features. From September 2018 to January 2019, a Kaggle competition[31] was organized by the Human Protein Atlas to bring together computer scientists and biologists to identify protein locations from these images.

When using traditional machine learning algorithms and models, typically, feature generation needs to be guided to some extent by domain experts. However, deep learning algorithms have shown to be capable of extracting the relevant and salient features from a given input. Therefore, in this case the feature generation is said to be data-driven. A typical example of data-driven feature generation is a neural network autoencoder, which attempts to learn its own inputs. In this case, the features are extracted from the output of the neurons in middle of the network, and can then be used to train other classifiers. A pictorial example of this architecture is shown in figure of Reference 32. Some work has already been done to use autoencoders as feature generations for protein function prediction.[32,33] A summary of typical features used to represent proteins for functional classification is shown in Table 1.

2.2 | Feature selection

In several applications, such as biology, the usage of machine learning techniques suffers from the curse of dimensionality, which means that the feature space is so large that the available data become sparse, and in turn, a performance degradation results. Therefore, this wealth of information needs to be filtered out to obtain a final set of features, which are suitable for the problem at hand. This step is known as dimensionality reduction. Feature selection, in which subsets of the original set of features are kept, is a special case of dimensionality reduction.

Naively, one would test each possible subset of features, and select the subset, which minimizes the error with respect to the ground truth. However, this brute-force approach is computationally feasible for only small feature sets. Molina et al hold that Feature Selection Algorithms (FSAs) can be characterized as a search problem in the hypothesis space (ie, space of candidate feature subsets) in terms of three aspects: search strategy, which is the general strategy with which the space of hypothesis is explored; generation of successor candidates (the mechanism by which possible variants of the current hypothesis are proposed); and evaluation measure, which is the function by which successor candidates are evaluated, allowing to compare different hypotheses to guide the search process.[85] A general review of feature selection in bioinformatics, with specific applications of these techniques in sequence analysis, microarray analysis, and mass spectra analysis is available in Reference 86. On the other hand, Wang et al[87] categorize feature selection algorithms for big data bioinformatics into exhaustive search, heuristic search, and hybrid methods.

Feature selection algorithms are generally classified into three main categories: wrapper methods, filter methods, and embedded methods. Wrapper methods evaluate candidate feature subsets by using the same type of predictive model (eg, random forest or support vector machines) that will be applied to the selected features later, when the final classification model will be built. Each new feature subset is used to train a model, which is tested on a hold-out set to obtain an error-rate. As wrapper methods train a new model for each subset, they are computationally intensive, but tend to provide the best performing feature set for that particular model. Recursive Feature Elimination (RFE) is an example of a wrapper method. The predictive model is initially fitted with all available features, and the weakest feature is then removed until a predetermined minimum number is reached. Examples of work, which used RFE include References 69, 71. On the other hand, forward feature selection starts with the evaluation of each individual feature and selects the one, which results in the best performing model. Then, all possible combinations of that selected feature and subsequent features are evaluated in order to select a second feature. This is iteratively repeated until a maximum

TABLE 1 Summary of typical features used to represent proteins for functional classification

Feature	Advantages	Disadvantages	Usage in literature
Physicochemical properties	Simple and numeric	Do not capture enough information about the protein	25 34-44
Sequence-based	Capture plenty of information	Typically require a conversion process to numeric data for machine learning	2,7,15-17,33,34,36,38,40,42-72
PPI networks	Neighboring proteins have a high probability of sharing functions	Reliability of PPI data depends on the experimental source	45,68,73-76
Biomedical text	Provides a rich source of information which is currently under-utilzed	Results are strongly affected by how informative the selected terms are	16,77-82
Immunohistochemistry images	Rich in features, easy to visualize	Requires more computational power and larger datasets, only useful for subcellular localization tasks	83
Representation learning	Removes the need for manual feature engineering and selection	Requires more computational power and larger datasets	32,33,45,47,64,77,84

number. Forward feature selection was used in Reference 72. In Reference 41, feature ranking was performed in the WEKA tool[88] using SVM as an evaluator.

Filter methods evaluate candidate feature subsets by using a proxy measure instead of the error rate obtained by the algorithm to be applied to the selected features later. This measure is chosen as it is computationally inexpensive, but still captures the usefulness of the feature set. Common examples include mutual information and the Pearson product-moment correlation coefficient. The *t* test and ANalysis Of VAriance (ANOVA) are two examples of univariate parametric filter methods, while the Wilcoxon rank sum is an example of a univariate model-free method. The ANOVA method was used by Tang et al to rank 400 dipeptides, which were later used to train a SVM classifier to predict differences between growth hormone binding proteins.[89] Al-Shahib et al used a filter-based approach to select discriminatory features. For each feature, the Wilcoxon signed-rank test was performed for each comparison of functional classes.[2] Features were retained if for at least one comparison of classes a Wilcoxon *P*-value < .02 was achieved, that is, they contribute potentially discriminating information. A filter method called FrankSum was specifically developed for protein function prediction.[90] It uses a combination of the Wilcoxon rank test *P*-value to measure the statistical significance of a single feature in discriminating two functional classes, and correlation coefficients to examine redundancy between features.

The Information Gain Ratio measure can also be used to rank features. This was used in References 40, 43. XGBoost, a type of gradient boosted tree algorithm, was used as a filter method in Reference 68 to select 32 GO features from an initial 21 000 features. In Reference 42, the authors evaluated the use of rough set theory as well as Correlation Feature Selection, Fast Correlation-Based Filter and Artificial Immune System as feature selection algorithms for classifying protein function. In Reference 91, rough sets were used to rank the top 15 features from a feature set built based on compositional percentages of the 20 amino acids properties. The Minimum Redundancy Maximum Relevance (mRMR) feature selection algorithm[92] is an extension of maximum-relevance, in which the selected features are those, which correlate strongest to the classification variable. As biological data often contains relevant but redundant data, mRMR attempts to address this problem by removing these redundant subsets. Several works made use of mRMR[43,51,76] for protein function prediction.

Embedded methods are a catch-all group of techniques, which perform feature selection as part of the model construction process. The classical example is the LASSO method for constructing a linear model, which penalizes the regression coefficients with an L1 penalty, reducing many of them to zero. Any features, which have nonzero coefficients, are "selected" by the LASSO algorithm. Another example of an embedded method is the Random Forest, which can be used to obtain feature importance. This technique was used in Reference 55, to rank protein sequence features for enzyme function classification. After feature ranking by a random forest, Lou et al, performed wrapper-based feature selection using a best-first forward search strategy.[57]

On the other hand, techniques such as Principal Component Analysis (PCA) or Linear Discriminant Analysis (LDA) produce a smaller set of new synthetic features from a linear combination of the original ones. PCA was used in References 17, 25, 93-95, while multilabel LDA was used in Reference 59. Apart from reducing the dimensionality of the input features, it may also be desirable to reduce the space of possible output labels. Makrodimitris et al. developed two novel Label-Space Dimensionality Reduction (LSDR) techniques to improve the CAFA performance of several function prediction algorithms.[58] From a NLP point of view, non-negative matrix factorization was used in Reference 96, to transform the bag of words input features into a new, compressed space that captures the variability of the data.

2.3 | Machine learning algorithms and models

Machine learning techniques are used to determine the parameters of a data-driven model, which would translate a given input to the

correct output. Protein function prediction is a classification problem, as the input needs to be mapped to a discrete output. Classifier models can be trained to perform this task using supervised, unsupervised, or semi-supervised learning. In supervised learning, a training dataset is available with a series of output labels (known as the ground truth) corresponding to input vectors. On the other hand, in unsupervised learning no ground truth is provided. Therefore, unsupervised learning techniques are primarily concerned with finding patterns and structures (eg, clusters) in the data, which then may need to be analyzed further. Semi-supervised learning lies between the two previous learning paradigms, in that the training set typically contains a mixture of a small amount of labeled data and a large amount of unlabeled data.

A large variety of machine learning algorithms and models have been developed in the past decades, and have also been applied in many contexts and applications. Among the simplest of supervised learning algorithms is logistic regression, in which a sigmoid function is used as a squashing function to map a real-valued input to a range from 0 to 1. You et al trained a logistic regressor on text-based features (TFIDF and D2V), derived from the MEDLINE biomedical literature database. This was done to predict between molecular function, biological process, and cellular component.[77] A kernel logistic regression model based on diffusion kernels for protein interaction networks was developed in Reference 73. The model achieved better prediction accuracy when compared to a previous model based on Markov random fields. Similarly, the authors in Reference 74 also trained a logistic regressor to predict protein function based on protein-protein interactions.

Naive Bayes classifiers are a family of simple, probabilistic classifiers, which apply Bayes' theorem with the strong (naive) assumption that all features are independent from each other given the class variable. In Reference 97, the authors train a Naive Bayes classifier to predict protein-protein interaction sites, while in Reference 98, the Extended Local Hierarchical Naive Bayes algorithm[99] was used.

The SVM algorithm[100] seeks to maximize the separation between points corresponding to different classes in some n-dimensional space, and therefore, determines a maximum-margin hyperplane. As the data are often not linearly separable in the original feature space, they are typically mapped to a higher-dimensional space in which the separation should be easier. This is achieved by means of kernel functions, such as the polynomial or the radial basis function (Gaussian function). These kernels have different variables (known as hyperparameters), which need to be tuned in order to achieve better performance. In the case of the widely used radial basis function, these include γ (which controls the width of the gaussian) and C (a penalty factor which controls overfitting vs underfitting) hyperparameters. Due to its successes in other fields, it is the most commonly used algorithm in initial works, which attempted to use machine learning techniques for protein function prediction. Examples of prior work using SVMs include References 2, 15, 32, 35, 38, 39, 41, 42, 46, 51, 60, 65-67, 69, 72, 81, 82, 95, 101-107. Generally, the best hyperparameters are identified through a grid search in the parameter space. In Reference 37, other techniques such as genetic algorithms and particle swarm optimization were also attempted, however, the grid search still yielded the best values.

The k nearest neighbors (kNN) algorithm is a nonparametric method which classifies a given observation through a majority vote of the labels of the closest k points in a given feature space. No model training is required. However, the majority voting procedure suffers when the class distribution is skewed. Examples in the literature in which kNN was used include References 25, 54, 58, 76, 80.

Ensemble methods combine several base models in order to produce a better predictive model. There are two categories of ensemble methods. In sequential methods, "boosting" is used to incrementally build an ensemble, by training each model on the same dataset but adjusting the weights of individual data points according to the error of the last prediction. Examples of such methods include AdaBoost[108] and Gradient Boosting.[109] The XGBoost algorithm,[110] is a scalable tree boosting system, which was used in Reference 68 to classify human proteins as aging-related or nonaging-related.

On the other hand, parallel methods use "bagging" (also known as bootstrap aggregation), to generate multiple base models simultaneously. The random forest[111] uses bagging as one of the two main sources of randomization, the other being the fact that it randomly samples features to be used as candidate features for selecting the best feature to split the data at each tree node. This technique was used for protein function prediction in References 43, 55, 71, 112. In Reference 113, a protein function prediction method called transductive multilabel classifier was developed, based on a directed birelational graph that models the relationship between proteins and functions. This was extended in the same paper to transductive multilabel ensemble classification.

Pitting several machine learning models against each other, and then determining the prediction output based on voting can also be used to develop an ensemble method. In Reference 114, the majority vote was used together with the mean ensemble and top k ensemble algorithms in predicting human protein subcellular localization.

Decision trees are one of the simplest machine learning models. Each leaf in the tree represents a decision or output of the model. A decision would have been reached after traversing a particular path along the tree's branches. Several implementations of decision trees exist. The C4.5 decision tree,[115] is generally used for classification and has been used in Reference 40. A novel decision tree classifier presented in Reference 62, improved on the C4.5 technique by using the uncertainty measure for best attribute selection. In Reference 116, the Clus-HMC heuristic[117] was used to select the best attributes to construct the tree. Another novel implementation of a decision tree was the Recursive Maximum Contrast Tree developed in Reference 118.

A neural network consists of a series of interconnected layers of units called neurons. Neurons are also known as perceptrons, which give rise to the term "multilayer perceptron," a typical neural network architecture. The number of neurons in the input layer should match the input feature dimension, while the number of neurons in the output layer should match the number of outputs. In classification problems, it is desirable to represent the output from the network using

one hot encoding. In one hot encoding, categorical variables are represented by a binary vector having a length equivalent to the cardinality of the set of values of the categorical variables. The vector is filled with zeros, except at the index of the categorical value, which is assigned a 1. The output of a given neuron is computed via an activation function, which in turn, takes as an input the weighted sum from the previous layer of neurons. Typical activation functions include the sigmoid, tanh and rectified linear unit (ReLU). Therefore, the goal of training is to learn appropriate values for the weights so as to obtain a correct output for a given input. In order to learn more complex input-output mappings, the neural network architecture typically has a number of intermediate layers called hidden layers.

In Reference 119, the authors performed hierarchical multilabel classification using local multilayer perceptrons. This approach takes into account the fact that proteins may perform several functions, which may be further specialized into subfunctions. The large number of output labels (eg, several thousand) can hinder the performance of machine learning algorithms. Therefore, in Reference 93, an ensemble of 100 neural networks was trained to predict protein function, each with 100 outputs, rather than a single neural network. A hierarchical neural network was also trained in Reference 120, exploiting the inherent hierarchical nature of protein function. The authors trained both Adaline networks, composed of a single layer of adjustable weights, and multilayer perceptrons (two layers). The latter architecture achieved better performance. Multilabel hierarchical classification was performed using competitive neural networks in Reference 121. The difference with respect to standard multilayer perceptrons is that neurons of the output layer compete to be activated, such that only one output neuron will be declared the "winner" of the competition process.

Another algorithm, which seeks to mimic the function of the brain, is the neural response.[122] It simulates the neuronal behavior of the visual cortex, and was used for protein function prediction in Reference 61, by defining a distance metric that corresponds to the similarity of the amino acid subsequences. The latter was important to understand how the brain can distinguish different sequences. Probabilistic neural networks use the Bayes optimal decision rule for classification, and take into account the probability density function for each class. The latter can be estimated using the Parzen nonparametric estimator. Probabilistic neural networks were used in Reference 46, and offered better predictive performance than kNN and SVM in identifying protein functional families from sequence.

Rather than being limited to predicting continuous or discrete valued outputs, deep learning[123] is particularly concerned with learning data representations, that is, feature learning. This allows the model to automatically discover the required features, replacing the traditional feature engineering and selection process. Therefore, they are also known as end-to-end models. Deep learning is commonly associated with neural network architectures, which have several hidden layers. With the advances in computing power afforded by Graphical Processing Units (GPUs), training of deep learning models for a variety of tasks is now within reach. This holds only for the cases where a very large amount of data is available, in order to properly estimate the very large number of parameters of deep neural networks.[123]

In Reference 45, three separate models (one per each GO subontology) were trained using deep learning on amino acid sequence and protein-protein interaction data. Trigrams were built from the amino acid sequences and converted to dense embeddings, while the PPI network features were used to generate knowledge graph embeddings. The sequence features were then passed through a 1D convolutional layer, and max pooling was then performed. The output from max pooling was then combined with the PPI network features into a fully connected layer with 1024 neurons, which was subsequently passed to hierarchically structured neural networks with sigmoid activation functions for classification. The full architecture is shown in figure of Reference 45.

Three deep architectures were evaluated in Reference 84 to predict human protein function. A multitask deep neural network (MTDNN), which consisted of shared hidden layers and task-specific hidden layers, comprised the first architecture and was developed by the authors. Its performance was compared to a multilabel deep neural network (in which shared hidden layers are used all the way until the final output layer), and a single-task deep neural network. The MTDNN performed better than the other two architectures, as well as FFPred3 and BLAST.

Deep network fusion was used for protein function prediction in Reference 32. A multimodal deep autoencoder was used to extract features, which were then passed on to a SVM. In Reference 47, deep learning was used to learn embeddings for protein sequences, which were restricted to a maximum length of 2000. Each amino acid was represented as a 23-dimensional vector. A convolutional layer together with average pooling was then trained on a GPU. A similar approach was used in Reference 107, in which the output from a stacked denoising autoencoder was passed onto a binary-relevance SVM. The input dataset consisted of microarray expression data and phylogenetic profiles for yeast. The authors in Reference 64 focused on human protein subcellular localization, and also used a stacked autoencoder. They tried SVM, random forests and softmax regression in the last layer of the deep learning network to make predictions, and found that the best results were achieved with the latter. Protein-protein interaction was the subject of machine learning based prediction in Reference 48. The authors applied a stacked autoencoder to protein sequence autocovariance Pan's PPI dataset. A similar approach was used in Reference 49.

Deep learning has also been applied to protein function prediction with text-based features. Deep semantic text representation was used in Reference 77 for biomedical literature. Multifunctional enzyme function prediction was achieved in Reference 78, with hierarchical multilabel deep learning.

As mentioned previously, immunohistochemistry images are a potential source of features in protein function prediction. Standard (so-called vanilla) neural network architectures run into problems when they need to be applied to images (in two dimensions or more when also considering, eg, color) or sequential data. In the case of the former, the high dimensionality due to the high image resolution

means that the neural network will have many parameters, which need to be learned. This leads to slow training and poor performance. Convolutional Neural Networks (CNNs),[124] on the other hand, take advantage of local spatial coherence in the images, and perform convolution operations, which result in fewer parameters, which need to be learned. Several CNN architectures developed recently, such as VGG16[125] or AlexNet,[126] are achieving high performance. In addition, it is possible to use these pretrained models on unseen data. In Reference 127, the author used a CNN in conjunction with both a SVM and a kNN classifier to predict protein function. On the other hand, in Reference 70, CNNs were trained to identify families of efflux proteins in transporters on features extracted from PSSM profiles.

Recurrent Neural Networks (RNNs)[128] are suitable for processing sequential data, as their architecture allows for maintaining an internal state. Long short-term memory (LSTM)[129] networks are an evolution of RNNs, which have been applied successfully in a number of domains, from speech recognition[130] to DNA sequences.[131] In Reference 53, a deep RNN was used to predict protein function from sequence, while in Reference 132 the authors used a three-unit LSTM together with neural machine translation.

In several cases, rather than just using a single machine learning model, results were achieved through a combination of algorithms. For instance, in Reference 63 the classification results from neural networks and SVM were fused via a heuristic fusion rule. In Reference 133, an ensemble multi-instance, multilabel learning neural network was trained. Multi-instance,[134] multilabel learning is useful when an observation is described by multiple instances and associated with multiple class labels.[135] Therefore, it is particularly applicable to protein function prediction, as proteins are often inherently multidomain and multifunctional, and each domain may fulfill its own function independently or in a concerted manner with its neighbors. A two-layer architecture was developed in Reference 133. In the first layer, training examples for each class label were clustered by invoking k-medoids, and then medoids of clustered groups were retained. Then neural nets were used to compute the basis functions between one example and the medoids.

Several other machine learning algorithms and models were used only sparingly in the literature. Rough set theory was developed in the early 1980s as a mathematical approach to intelligent data analysis and data mining.[136] It distinguishes between objects based on the concept of indiscernibility, and deals with the approximation of sets using binary relations constructed from empirical data. Rough sets were used in Reference 91 to predict between seven pectin lyase-like subfamilies. In Reference 137, the protein was modeled as a document, while the protein function label was the topic. A supervised topic model (labeled latent Dirichlet allocation[138]) was used to make predictions based on protein sequences organized into a bag of words. Bag of words from protein sequence was also used to generate features in Reference 59, with a model based on multilabel linear discriminant analysis then being trained. Finally, multilabel Gaussian kernel regression was used in Reference 139.

3 | MACHINE LEARNING MODEL IMPLEMENTATION, TUNING, AND EVALUATION

In the past decade, machine learning frameworks have evolved in response to increasing demand across various disciplines. The most commonly used frameworks are Scikit-Learn[140] for the Python programming language (which has been ranked as the most commonly used programming language by the IEEE Spectrum since 2017[141]), several packages in R and the Statistics and Machine Learning toolbox of MATLAB.[142] Due to the computationally intensive nature of deep learning, several libraries and frameworks have also been developed which allow models to be trained at faster speeds on GPUs and computing clusters, such as TensorFlow,[143] Keras,[144] Caffe,[145] and PyTorch.[146] With increasing amounts of training data available, from Gene Ontology to biomedical literature, as well as new, computationally intensive architectures, which use deep learning, the use of GPUs is becoming more prevalent in training machine learning algorithms to predict protein function. Examples of works in the literature, which made use of hardware acceleration include References 45, 47, 64, 107.

The machine learning models described in the previous section often need to be tuned to achieve a more satisfactory performance. This involves determining appropriate hyperparameters (which are set prior to training by the data scientist as opposed to the parameters, which are learnt during training), which also allow the model to generalize and perform well even with unseen data. There are only a few general hyperparameters, such as the optimizer (eg, Adam,[147] RMSprop,[148] or stochastic gradient descent) and the learning rate, while the rest are usually specific to a particular model or algorithm. In particular, in neural networks, these might be the activation function of the neuron, as well as the number of neurons in each layer and the number of hidden layers. Neural network performance can also be boosted by conducting the training over multiple epochs, or by increasing the number of times that the training data flows through the network.

Although there are suggested ranges of values and rules of thumb, there is no exact science of selecting the best hyperparameters prior to training. The most widely used method is to perform a search in the space of hyperparameters, either in a random fashion or using a systematic grid search. The hyperparameter search is often facilitated by several modern machine learning frameworks and combined with cross-validation. In k-fold cross-validation, its simplest form, the randomly shuffled training dataset is split into k groups, and in each group a percentage of samples is used as a training set, while the remainder is used as a test set. Therefore, an averaged performance result can be obtained, avoiding the so-called "lucky split."

Sometimes, the performance of machine learning classifiers can be boosted by mitigating class-imbalance, which occurs when the number of samples in each class is skewed. In Reference 149, the authors compared the performance of three class-balance strategies for SVM in relation to protein function prediction. These included under-

sampling (in which the extra samples of the majority class(es) are discarded), Synthetic Minority Over-sampling Technique (SMOTE), in which synthetic samples of the minority class are added to the dataset, and weighted SVM, which keeps the number of samples in each class but assigns appropriate weights during training to specifically improve the performance for the minority class. The latter two techniques achieved the best results, although weighted SVM was less computationally demanding.

As protein function prediction is generally treated as a classification problem, the metrics used to evaluate the performance of the machine learning models typically include accuracy, precision, recall (sensitivity), specificity, and the F1-score. The F1-score is defined as the harmonic average of the precision and the recall, and handles class imbalance better than accuracy (since accuracy can be trivially maximized by always predicting the majority class). In addition, for better visualization and performance understanding, a Receiver Operating Characteristic (ROC) curve can be obtained by plotting the true positive rate as a function of the false positive rate. The larger the Area Under Curve (AUC), the better the performance, as this normally means that a higher true positive rate is achieved for the same false positive rate. A similar metric can be derived from the precision-recall (PR) curve, known as Area Under PR (AUPR). Two further metrics used as the gold standard in the CAFA challenges are F_{max} and $S_{min.}$[11] The former is defined as the maximum F1-score obtained by varying the classifier threshold (and therefore the working point along, eg, a precision-recall curve), while the latter is obtained by minimizing the uncertainty and misinformation. A list of commonly used metrics found in the literature for evaluating the performance of classifiers for protein function prediction is shown in Table 2. Typically, several metrics tend to be calculated in a given work as most of them provide complementary information.

Although certain algorithms and models have proved to learn input-output mappings more effectively than others, and in a variety of domains, it is appropriate to train different machine learning models and see which results in the best performance. In Reference 152, the performance of logistic regression, Naive Bayes, SVMs, a decision tree and a neural network, was compared to evaluate the suitability of using dissimilarity representations. The SVM algorithm was found to give the best results in terms of F1-score and AUC metrics. In Reference 17, extreme machine learning was compared to a SVM, while in Reference 57, Gaussian Naive Bayes was trained together with a decision tree, random forest, logistic regression, kNN and SVM with both polynomial and RBF kernels. A SVM and the kNN algorithm were trained on sequence motifs for enzyme classification.[52] Finally, in Reference 153, the performance of a simple neural network was compared to that of a SVM for prediction of protein-protein interactions in human *Bacillus anthracis*.

4 | APPLICATIONS OF MACHINE LEARNING FOR PROTEIN FUNCTION PREDICTION

In most of the literature, machine learning algorithms are trained to predict protein function using a particular classification scheme as a ground truth. The most common taxonomies are Functional Catalogue (FunCat),[154] Enzyme Commission (EC),[155] and Gene Ontology.[5] The FunCat annotation scheme consists of 28 main categories that cover general features such as cellular transport, metabolism, and protein activity regulation. Each of the main branches has a hierarchical, tree-like structure. The EC is a hierarchical classification scheme for enzymes, based on the chemical reactions they catalyze. The top level consists of seven enzyme classes, such as oxidoreductases, hydrolases, and ligases. Gene ontology (GO) defines a representation of terms for gene product properties. The ontology covers three domains: cellular component (which refer to parts of a cell or its extracellular environment); molecular function (the elemental activities of a gene product at the molecular level, such as binding or catalysis); and biological process (operations or sets of molecular events with a defined beginning and end, relevant to the functioning of integrated living units such as cells and tissues). Each of the three GO ontologies is a Directed Acyclic Graph (DAG), where a node (GO term) can have multiple parents in the hierarchy, unlike the simpler tree-based hierarchies for the EC code and FunCat mentioned earlier.

Earlier works focused on classification schemes such as EC and FunCat. In Reference 36, the authors train a SVM and a Random Forest to predict top-level EC classes from seven features, such as amino acid sequence, molecular weight, and chain length. SVMs were also used in Reference 60 for the same class prediction, this time using two sequences per protein corresponding to the primary and secondary structures. In Reference 34, SVMs were trained on features such as physicochemical properties and sequence similarities to predict five different levels of enzymatic function. Very good results (F1-score of $\sim$0.99) were achieved. Protein functions were predicted according to the FunCat taxonomy in Reference 59, using multilabel linear discriminant analysis, and in Reference 75 using multilabel semi-supervised learning on graphs, which were based on input features from protein-protein interactions. The latter category of input features was also used in Reference 74 to train a logistic regressor and predict 17 FunCat classes.

Protein function prediction based on GO terms is a more recent initiative. The various editions of the CAFA challenge have made available an increasing number of sequences to the community with the task of predicting GO annotations. A summary of the latest CAFA3 and CAFA-π challenges is available in Reference 156. The GOLabeler ensemble method,[50] which combines BLAST-kNN, logistic regression and a Naive GO term frequency computation to solve the problem of Learning To Rank (LTR) achieved the best performance when compared to other CAFA3 entries across the board (ie, for molecular function, biological process and cellular component ontologies). Machine learning techniques have been used to prediction functions related to one, two, or all three domains. According to CAFA, the prediction accuracy, which uses machine learning techniques is lowest for the Biological Process domain.[11] Around a dozen works attempt to predict protein function related to all three domains. The techniques used, which were already expanded upon earlier, range from deep neural networks[32,45,47,84,132] to kNN,[58] logistic regression,[77] and SVMs.[56,79,103] In Reference 80, a kNN classifier was

TABLE 2 List of commonly used metrics found in the literature for evaluating the performance of classifiers for protein function prediction

Metric	Advantages	Disadvantages	Usage in literature
Accuracy	Answers the question: how many samples were correctly labeled out of all samples?	Provides misleading information in the event of class imbalance	16,17,25,32,47,48,53,54,56,60,64-67,91,105,139,150,151
Precision	Answers the question: how many samples labeled as COI actually belong to the COI?	Does not consider false negatives	17,34,36,46-48,53-55,60,82,107,132,150
Recall	Answers the question: of all the samples which actually belong to the COI, how many were correctly predicted?	Does not consider false positives	15,17,34,36,39,46-48,53-56,60,74,82,107,132,139,150
Specificity	Answers the question: of all the samples which do not belong to the COI, how many were correctly predicted?	Does not consider false negatives	15,39,46,48,56,62,74,139,150
F1-score	Better suited for cases of class imbalance	Not as intuitive as other metrics	32,34,36,47,53,56,96,107,132
AUROC	Score is independent of the threshold set for the classifier	Provides misleading information in the event of class imbalance	36,50,56,69
AUPR	Score is independent of the threshold set for the classifier and not affected by class imbalance	Does not consider true negatives	50,58,77
F_{max}	Considers predictions across the full spectrum from high to low sensitivity	Penalizes specific predictions	45,50,77,82,84
S_{min}	Takes the structure of the ontology and the dependencies between terms induced by a hierarchical ontology into account	Assumes that a Bayesian network structured according to the underlying ontology will perfectly model the prior probability distribution of a target variable	50,77

Abbreviations: COI, class of interest; FN, false negative; FP, false positive; TN, true negative; TP, true positive.

used to predict protein function from text-based features derived from biomedical literature in both the molecular function and biological process domains.

Only molecular function was considered in the works of References 70, 89, 96, 149, 157. Transductive multilabel ensemble classification was used to determine protein functions related only to Biological Process in Reference 113. Most of the literature, which focused on predicting protein function pertinent to a sole domain, was relevant to the cellular component category. The most used machine learning model has been SVM, which was used in References 15, 16, 44, 65, 66, 72, 81, 101, 150, 158. Deep learning is used in References 64, 139. In Reference 83, immunohistochemistry images from the Human Protein Atlas database were used. An ensemble strategy was used in Reference 114 for human protein subcellular localization, while the authors in Reference 71, used a random forest to predict Golgi-resident protein types from non-Golgi resident.

In other instances in the literature, machine learning algorithms were trained to predict whether a given protein would fit into one of a select number of classes. In Reference 151, three different models (SVM, random forest, and kNN) were trained on short-linear motifs to predict whether a particular protein was a calmodulin-binding or a mitochondrial protein. A SVM was also used in Reference 104, to

TABLE 3 Performance comparison of various machine learning models and algorithms on the EC taxonomy

Taxonomy	Protein	ML method	Hyperparameter optimization	Result (metric)	Usage in literature
EC	Enzyme	Random forest	N/A	0.486 (F1-score)	36
	Enzyme	SVM	N/A	0.480 (F1-score)	34
	Enzyme	C4.5 Classifier	N/A	0.7213 (F1-score)	40
	Enzyme	SVM	GA	0.70 (F1-score)	37
	Enzyme	SVM	PSO	0.69 (F1-score)	37
	Enzyme	Deep neural network	N/A	0.965 (F1-score)	78

Abbreviations: GA, genetic algorithm; PSO, particle swarm optimization.

TABLE 4 Performance comparison of various machine learning models and algorithms on the FunCat taxonomy

Taxonomy	Protein	ML method	Hyperparameter optimization	Result (metric)	Usage in literature
FunCat	Yeast	MLDA	N/A	0.412 (F1-score)	[59]
	Yeast	MLDA + graph	N/A	0.437 (F1-score)	[59]
	Yeast	NMLDA + graph	N/A	0.440 (F1-score)	[59]
	Yeast	MCSL-d (PPI-weight, infor)	grid-search	0.4857 (F1-score)	[75]
	Yeast	MCSL-b (PPI-weight, infor)	grid-search	0.4865 (F1-score)	[75]

Abbreviations: MCSL, multilabel correlated semi-supervised learning; MLDA, multilabel linear discriminant analysis; NMLDA, L1-normalized MLDA.

determine whether a particular protein was an apolipoprotein or not. Apolipoproteins are crucial in cardiovascular systems and drug design. The authors in Reference 89 developed a tool (HBPred) to identify growth hormone-binding proteins. Dipeptide composition, which describes the correlation between the two most contiguous amino acid residues, was used as a feature on which a SVM was trained. The same type of machine learning model was also used in Reference 69 to classify signaling proteins based on molecular star graph descriptors. A plethora of techniques, including Gaussian Naive Bayes, decision trees, random forest, and logistic regression, were used in Reference 57, to develop a binary classifier between DNA-binding and non DNA-binding proteins. In Reference 62, a decision tree classifier was trained to predict between the five molecular classes of HPRD, namely defensin, cell surface receptor, DNA repair protein, heat shock protein, and voltage gated channel. In Reference 54, a kNN multilabel classifier was used to predict enzyme function at the level of chemical mechanism. A SVM was used to predict between RNA-binding, DNA-binding and EF-hand proteins in Reference 41. Rough sets were used to predict between seven pectin lyase-like subfamilies in Reference 91, based on features derived from amino acid composition.

Despite the significant body of work in which machine learning algorithms are trained to predict protein function, relatively little effort has been devoted to the issue of class imbalance in function labels. This imbalance is a result of the fact that for example, the GO

TABLE 5 Performance comparison of various machine learning models and algorithms on the GO taxonomy (molecular function)

Protein	ML method	Hyperparameter optimization	Result (metric)	Usage in literature
Yeast	Autoencoder + SVM	Manual adjustment of activation functions, number and sizes of hidden layers, batch sizes and learning rates for autoencoder Nested 5-fold CV via grid-search over γ and C for RBF kernel for SVM	0.27 (F1-score)	[32]
Human	Autoencoder + SVM	Same as above	0.18 (F1-score)	[32]
Human	Deep neural network	Manual tuning of minibatch size, number of convolution filters, filter size, number of neurons in fully connected layer and learning rate.	0.51 (F_{max})	[45]
Difficult proteins	LR and BLAST-kNN	N/A	0.62 (F_{max})	[77]
Difficult proteins	LR and BLAST-kNN	N/A	5.171 (S_{min})	[77]
Difficult proteins	LR and BLAST-kNN	N/A	0.567 (F_{max})	[50]
Difficult proteins	LR and BLAST-kNN	N/A	5.087 (S_{min})	[50]
Human	LR and BLAST-kNN	N/A	0.625 (F_{max})	[156]
Human	MTDNN	grid-search performed using HYPEROPT [160] to obtain number of shared layers, number of hidden units in each shared layer, number of specific layers, number of hidden units in each shared layer, drop-out rate, learning rate, L1/L2 regularization	0.311 (F_{max})	[84]
Human	MLDNN	grid-search performed using HYPEROPT [160] on number of hidden layers, number of units inside each hidden layer, batch size, learning rate, dropout rate and L1/L2 regularization	0.343 (F_{max})	[84]
Human	STDNN	Same as MLDNN	0.338 (F_{max})	[84]

Note: Difficult proteins have a global sequence identity of less than 60%.
Abbreviations: CV, cross-validation; LR, logistic regression.

TABLE 6 Performance comparison of various machine learning models and algorithms on the GO taxonomy (biological process)

Protein	ML method	Hyperparameter optimization	Result (metric)	Usage in literature
Yeast	Autoencoder + SVM	Activation functions, number and sizes of hidden layers, batch sizes and learning rates for autoencoder Nested 5-fold CV via grid-search over γ and C for RBF kernel for SVM	0.19 (F_{max})	[32]
Human	Autoencoder + SVM	Same as above	0.125 (F_{max})	[32]
Human	Deep neural network	Manual tuning of minibatch size, number of convolution filters, filter size, number of neurons in fully connected layer and learning rate.	0.42 (F_{max})	[45]
Difficult proteins	LR and BLAST-kNN	N/A	0.46 (F_{max})	[77]
Difficult proteins	LR and BLAST-kNN	N/A	16.82 (S_{min})	[77]
Difficult proteins	LR and BLAST-kNN	N/A	0.382 (F_{max})	[50]
Difficult proteins	LR and BLAST-kNN	N/A	14.538 (S_{min})	[50]
Human	MTDNN	grid-search performed using HYPEROPT [160] to obtain number of shared layers, number of hidden units in each shared layer, number of specific layers, number of hidden units in each shared layer, drop-out rate, learning rate, L1/L2 regularization	0.298 (F_{max})	[84]
Human	MLDNN	grid-search performed using HYPEROPT [160] on number of hidden layers, number of units inside each hidden layer, batch size, learning rate, dropout rate and L1/L2 regularization	0.287 (F_{max})	[84]
Human	STDNN	Same as MLDNN	0.288 (F_{max})	[84]

Note: Difficult proteins have a global sequence identity of less than 60%.
Abbreviations: CV, cross-validation; LR, logistic regression.

TABLE 7 Performance comparison of various machine learning models and algorithms on the GO taxonomy (cellular component)

Protein	ML method	Hyperparameter optimization	Result (metric)	Usage in literature
Yeast	Autoencoder + SVM	Activation functions, number and sizes of hidden layers, batch sizes and learning rates for autoencoder Nested 5-fold CV via grid-search over γ and C for RBF kernel for SVM	0.155 (F_{max})	[32]
Human	Autoencoder + SVM	Same as above	0.125 (F_{max})	[32]
Human	Deep neural network	Manual tuning of minibatch size, number of convolution filters, filter size, number of neurons in fully connected layer and learning rate.	0.60 (F_{max})	[45]
Difficult proteins	LR and BLAST-kNN	N/A	0.69 (F_{max})	[77]
Difficult proteins	LR and BLAST-kNN	N/A	4.45 (S_{min})	[77]
Difficult proteins	LR and BLAST-kNN	N/A	0.706 (F_{max})	[50]
Difficult proteins	LR and BLAST-kNN	N/A	5.344 (S_{min})	[50]
Human	LR and BLAST-kNN	N/A	0.6 (F_{max})	[156]
Human	MTDNN	grid-search performed using HYPEROPT [160] to obtain number of shared layers, number of hidden units in each shared layer, number of specific layers, number of hidden units in each shared layer, drop-out rate, learning rate, L1/L2 regularization	0.484 (F_{max})	[84]
Human	MLDNN	grid-search performed using HYPEROPT [160] on number of hidden layers, number of units inside each hidden layer, batch size, learning rate, dropout rate and L1/L2 regularization	0.449 (F_{max})	[84]
Human	STDNN	Same as MLDNN	0.425 (F_{max})	[84]

Note: Difficult proteins have a global sequence identity of less than 60%.
Abbreviations: CV, cross-validation; LR, logistic regression.

database rarely stores which proteins do not possess a particular function. In Reference 159, the authors developed two novel negative selection algorithms (Selection of Negatives through Observed Bias and Negative Examples from Topic Likelihood) to determine whether a protein does or does not perform a particular function.

A summary of the comparison in performance between various machine learning models and algorithms is provided in Tables 3 and 4 for the EC and FunCat taxonomies respectively, and Tables 5–7 for the molecular function, biological process, and cellular component GO taxonomies respectively. As can be seen, machine learning methods applied to the EC and FunCat taxonomies generally did not disclose any hyperparameter optimization strategy, except for a grid search. In particular, in Reference 37 genetic algorithms and particle swarm optimization were used, however did not result in an increase in performance with respect to Reference 40, which used a decision tree. Multilabel correlated semi-supervised learning[75] with grid-search gave an improvement in terms of F1-score over multilabel linear discriminant analysis[59] for the FunCat taxonomy. The best performance for molecular function GO terms was given by GoLabeler which did not make use of deep learning or hyperparameter optimization. A similar model developed by the same authors also gave the best performance for biological process for difficult proteins (ie, proteins have a global sequence identity of less than 60%).[77] However, for the cellular component ontology, it performed as well as a deep neural network approach (which required hyperparameter optimization) for human proteins.[45]

5 | CONCLUSIONS AND FUTURE PERSPECTIVES

This paper has reviewed the evolution in features and machine learning techniques used to train data-driven models for protein function prediction. Although there has been a rise in the use of deep learning techniques to extract meaningful features and develop high performing predictors, methods using classical machine learning techniques such as logistic regression were still able to outperform deep learning approaches. In addition, methods which do not use machine learning still feature prominently in the top 10 performers of CAFA 3, as opposed to deep learning approaches. The fact that deep learning requires a very large amount of data remains a limitation, which probably reduces its success at least in some studies concerning protein function prediction. Nevertheless, the bioinformatics community has been quite successful in its efforts to bring machine learning and proteins together, through initiatives such as the CAFA challenge and Kaggle competitions. The community will keep this momentum going by facilitating the proliferation of databases and frameworks which are more appropriate for machine learning. Researchers are now also resorting to a much wider variety of input features, particularly those derived from biomedical text. Reliable data-driven models are key to narrowing the gap between the number of sequences with known and unknown function, which will ultimately help elucidate the effect of mutations in proteins on diseases and in the engineering of new proteins.

ORCID

Rosalin Bonetta https://orcid.org/0000-0003-4696-7770
Gianluca Valentino https://orcid.org/0000-0003-3864-7785

REFERENCES

1. Eisenberg D, Marcotte EM, Xenarios I, Yeates TO. Protein function in the post-genomic era. *Nature*. 2000;405:823-826.
2. Al-Shahib A, Breitling R, Gilbert DR. Predicting protein function by machine learning on amino acids sequences - a critical evaluation. *BMC Genomics*. 2007;8:78.
3. Rost B, Liu J, Nair R, Wrzeszczynski KO, Ofran Y. Automatic prediction of protein function. *Cell Mol Life Sci*. 2003; 60: 2637-2650.
4. Mills CL, Beuning PJ, Ondrechen MJ. Biochemical functional predictions for protein structures of unknown or uncertain function. *Comput Struct Biotechnol J*. 2015;02(13):182-191. https://www.ncbi.nlm.nih.gov/pubmed/25848497.
5. Ashburner M, Ball CA, Blake JA, et al. Gene ontology: tool for the unification of biology. *Nat Genet*. 2000;25:25-29.
6. Friedberg I. Automated protein function prediction - the genomic challenge. *Brief Bioinform*. 2006;7:225-242.
7. Lee D, Redfern O, Orengo C. Predicting protein function from sequence and structure. *Nat Rev Mol Cell Biol*. 2007;8:995-1005.
8. Gardy JL, Brinkman FS. Methods for predicting bacterial protein subcellular localization. *Nat Rev Microbiol*. 2006;4:741-751.
9. The UniProt Consortium. UniProt: the universal protein knowledgebase. *Nucleic Acids Res*. 2017;45:D158-D169.
10. Punta M, Coggill PC, Eberhardt RY, et al. The Pfam protein families database. *Nucleic Acids Res*. 2012;40:D290-D301.
11. Jiang Y, Oron TR, Clark WT, et al. An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome Biol*. 2016;17:184.
12. Bernardes JS, Pedreira CE. A review of protein function prediction under machine learning perspective. *Recent Pat Biotechnol*. 2013;7:122-141.
13. Sharma M, Garg P. Computational approaches for enzyme functional class prediction: a review. *Curr Proteomics*. 2014;11(1):17-22. https://www.ingentaconnect.com/content/ben/cp/2014/00000011/00000001/art00003.
14. Wang Z, Zou Q, Jiang Y, Ju Y, Zeng X. Review of protein subcellular localization prediction. *Curr Bioinformatics*. 2014;9(3):331-342. https://www.ingentaconnect.com/content/ben/cbio/2014/00000009/00000003/art00015.
15. Hoglund A, Donnes P, Blum T, Adolph HW, Kohlbacher O. MultiLoc: prediction of protein subcellular localization using N-terminal targeting sequences, sequence motifs and amino acid composition. *Bioinformatics*. 2006;22:1158-1165.
16. Shatkay H, Hoglund A, Brady S, Blum T, Donnes P, Kohlbacher O. SherLoc: high-accuracy prediction of protein subcellular localization by integrating text and protein sequence data. *Bioinformatics*. 2007;23:1410-1417.
17. You ZH, Lei YK, Zhu L, Xia J, Wang B. Prediction of protein-protein interactions from amino acid sequences with ensemble extreme learning machines and principal component analysis. *BMC Bioinformatics*. 2013;14:S10.
18. Schwikowski B, Uetz P, Fields S. A network of protein-protein interactions in yeast. *Nat Biotechnol*. 2000;18:1257-1261.

19. Shannon P, Markiel A, Ozier O, et al. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res*. 2003;13:2498-2504.
20. Govindan G, Nair AS. Composition Transition and Distribution (CTD)? A dynamic feature for predictions based on hierarchical structure of cellular sorting. 2011 Annual IEEE India Conference, 2011. p. 1–6.
21. Liu ZX, Liu SL, Yang HQ, Bao LH. Using protein granularity to extract the protein sequence features. *J Theor Biol*. 2013;331:48-53.
22. Chou KC. Prediction of protein signal sequences and their cleavage sites. *Proteins*. 2001;42:136-139.
23. Gribskov M, McLachlan AD, Eisenberg D. Profile analysis: detection of distantly related proteins. *Proc Natl Acad Sci USA*. 1987;84:4355-4358.
24. Jeong JC, Lin X, Chen XW. On position-specific scoring matrix for protein function prediction. *IEEE/ACM Trans Comput Biol Bioinform*. 2011;8:308-315.
25. Wang W, Zhang X, Meng J, Luan Y. Protein function prediction based on physiochemical properties and protein granularity. Proceedings of IEEE International Conference on Granular Computing Beijing, China, 2013. p. 342–346.
26. Verspoor KM. Roles for text mining in protein function prediction. *Methods Mol Biol*. 2014;1159:95-108.
27. Zeng Z, Shi H, Wu Y, Hong Z. Survey of natural language processing techniques in bioinformatics. *Comput Math Methods Med*. 2015;2015:1-10.
28. Mikolov T, Sutskever I, Chen K, Corrado G, Deap J. Distributed representations of words and phrases and their compositionality. Proceedings of 26th International Conference on Neural Information Processing Systems Lake Tahoe, USA, 2013. p. 3111–3119.
29. Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space, 2013.
30. Asgari E, Mofrad MR. Continuous distributed representation of biological sequences for deep proteomics and genomics. *PLoS One*. 2015;10:e0141287.
31. Kaggle, Human Protein Atlas Image Classification. 2018. https://www.kaggle.com/c/human-protein-atlas-image-classification.
32. Gligorijevic V, Barot M, Bonneau R. deepNF: deep network fusion for protein function prediction. *Bioinformatics*. 2018;34:3873-3881.
33. Wang J, Zhang L, Jia L, Ren Y, Yu G. Protein-protein interactions prediction using a novel local conjoint triad descriptor of amino acid sequences. *Int J Mol Sci*. 2017;18:E2373.
34. Dalkiran A, Rifaioglu A, Martin M, Cetin-Atalay R, Atalay V, Dogan T. ECPred: a tool for the prediction of the enzymatic functions of protein sequences based on the EC nomenclature. *BMC Bioinformatics*. 2018;19:334.
35. Rahman S, Bakar A, Hussein Z. Data mining framework for protein function prediction. Proceedings of IEEE International Symposium on Information Technology Kuala Lumpur, Malaysia, 2008.
36. Srivastava A, Mahmood R, Srivastava R. A comparative analysis of SVM random forest methods for protein function prediction. Proceedings of IEEE International Conference on Current Trends in Computer, Electrical, Electronics and Communication Mysore, India, 2018. p. 1008–1010.
37. Silva M, Leijoto L, Nobre C. Algorithms analysis in adjusting the SVM parameters: an approach in the prediction of protein function. *J Appl Artif Intell*. 2017;31:316-331.
38. Cai CZ, Han LY, Ji ZL, Chen X, Chen YZ. SVM-Prot: Web-based support vector machine software for functional classification of a protein from its primary sequence. *Nucleic Acids Res*. 2003;31:3692-3697.
39. Cai CZ, Wang WL, Sun LZ, Chen YZ. Protein function classification via support vector machine approach. *Math Biosci*. 2003;185:111-122.
40. Lee B, Ryu K. Feature extraction from protein sequences and classification of enzyme function. Proceedings of IEEE International Conference on Biomedical Engineering and Informatics Sanya, China, 2008. p. 138–142.
41. Lee B, Lee H, Kim D, Ryu K. Feature extraction in spatially-conserved regions and protein functional classification. Proceedings of Frontiers in the Convergence of Bioscience and Information Technologies Jeju City, Korea, 2007. p. 165–170.
42. Rahman S, Bakar A, Hussein Z. Experimental study of different FSAs in classifying protein function. Proceedings of IEEE International Conference of Soft Computing and Pattern Recognition Malacca, Malaysia, 2009. p. 516–521.
43. Li F, Li C, Wang M, et al. GlycoMine: a machine learning-based approach for predicting N-, C- and O-linked glycosylation in the human proteome. *Bioinformatics*. 2015;31:1411-1419.
44. Acquaah-Mensah GK, Leach SM, Guda C. Predicting the subcellular localization of human proteins using machine learning and exploratory data analysis. *Genomics Proteomics Bioinformatics*. 2006;4:120-133.
45. Kulmanov M, Khan MA, Hoehndorf R, Wren J. DeepGO: predicting protein functions from sequence and interactions using a deep ontology-aware classifier. *Bioinformatics*. 2018;34:660-668.
46. Li Y et al. SVM-Prot 2016: a web-server for machine learning prediction of protein functional families from sequence irrespective of similarity. *PLoS One*. 2016;11:e0155290.
47. Nauman M, Rehman H, Politano G, Benso A. Beyond homology transfer: deep learning for automated annotation of proteins. *J Grid Comput*. 2018;17:225–237.
48. Sun T, Zhou B, Lai L, Pei J. Sequence-based prediction of protein protein interaction using a deep-learning algorithm. *BMC Bioinformatics*. 2017;18:277.
49. Wang YB, You ZH, Li X, et al. Predicting protein-protein interactions from protein sequences by a stacked sparse autoencoder deep neural network. *Mol Biosyst*. 2017;13:1336-1344.
50. You R, Zhang Z, Xiong Y, Sun F, Mamitsuka H, Zhu S. GOLabeler: improving sequence-based large-scale protein function prediction by learning to rank. *Bioinformatics*. 2018;34:2465-2473.
51. You ZH, Zhu L, Zheng CH, Yu HJ, Deng SP, Ji Z. Prediction of protein-protein interactions from amino acid sequences using a novel multi-scale continuous and discontinuous feature set. *BMC Bioinformatics*. 2014;15:S9.
52. Ben-Hur A, Brutlag D. Sequence motifs: highly predictive features of protein function. In: Guyon I, Nikravesh M, Gunn S, Zadeh L, eds. *Feature Extraction*. Berlin, Heidelberg: Springer; 2006:625-645.
53. Liu X. Deep Recurrent Neural Network for Protein Function Prediction from Sequence, 2017.
54. Ferrari LD, Mitchell J. From sequence to enzyme mechanism using multi-label machine learning. *BMC Bioinformatics*. 2014;15:150.
55. Kumar C, Li G, Choudhary A. Enzyme function classification using protein sequence features and random forest. Proceedings of IEEE International Conference on Bioinformatics and Biomedical Engineering Beijing, China, 2009.
56. Lee B, Shin M, Young J, Hae O, Ryu K. Identification of protein functions using a machine-learning approach based on sequence-derived properties. *Proteome Sci*. 2009;7:27.
57. Lou W, Wang X, Chen F, Chen Y, Jiang B, Zhang H. Sequence based prediction of DNA-binding proteins based on hybrid feature selection using random forest and Gaussian naive Bayes. *PLoS One*. 2014;9:e86703.
58. Makrodimitris S, van Ham R, Reinders M. Improving protein function prediction using protein sequence and GO-term similarities. *Bioinformatics*. 2018;35:1116-1124.
59. Wang H, Yan L, Huang H, Ding C. From protein sequence to protein function via multi-label linear discriminant analysis. *IEEE/ACM Trans Comput Biol Bioinform*. 2017;14:503-513.

60. Resende W, Nascimento R, Xavier C, Lopes I, Nobre C. The use of support vector machine and genetic algorithms to predict protein function. Proceedings of IEEE International Conference on Systems, Man and Cybernetics Seoul, South Korea, 2012. p. 1773–1778.
61. Yalamanchili HK, Wang J, Xiao Q. NRProF: neural response based protein function prediction algorithm. Proceedings of IEEE International Conference on Systems Biology Zhuhai, China, 2011. p. 33–40.
62. Singh M, Singh P, Singh H. Decision tree classifier for human protein function prediction. Proceedings of IEEE International Conference on Advanced Computing and Communications Surathkal, India, 2006. p. 564–568.
63. Amidi S, Amidi A, Vlachakis D, Paragios N, Zacharaki EI. Automatic single- and multi-label enzymatic function prediction by machine learning. *PeerJ*. 2017;5:e3095.
64. Wei L, Ding Y, Su R, Tang J, Zou Q. Prediction of human protein subcellular localization using deep learning. *J Parallel Distr Comput*. 2018;117:212-217.
65. Yu CS, Lin CJ, Hwang JK. Predicting subcellular localization of proteins for gram-negative bacteria by support vector machines based on n-peptide compositions. *Protein Sci*. 2004;13:1402-1406.
66. Park KJ, Kanehisa M. Prediction of protein subcellular locations by support vector machines using compositions of amino acids and amino acid pairs. *Bioinformatics*. 2003;19:1656-1663.
67. Zhou X, Chen C, Li Z, Zou X. Using Chou's amphiphilic pseudo-amino acid composition and support vector machine for prediction of enzyme subfamily classes. *J Theor Biol*. 2007;248:546-551.
68. Kerepesi C, Daroczy B, Sturm A, Vellai T, Benczur A. Prediction and characterization of human ageing-related proteins by using machine learning. *Sci Rep*. 2018;8:4094.
69. Fernandez-Lozano C, Cuinas RF, Seoane JA, Fernandez-Blanco E, Dorado J, Munteanu CR. Classification of signaling proteins based on molecular star graph descriptors using machine learning models. *J Theor Biol*. 2015;384:50-58.
70. Taju SW, Nguyen TT, Le NQ, Kusuma R, Ou YY. DeepEfflux: a 2D convolutional neural network model for identifying families of efflux proteins in transporters. *Bioinformatics*. 2018;34:3111-3117.
71. Yang R, Zhang C, Gao R, Zhang L. A novel feature extraction method with feature selection to identify golgi-resident protein types from imbalanced data. *Int J Mol Sci*. 2016;17:218.
72. Lin H, Ding H, Guo FB, Huang J. Prediction of subcellular location of mycobacterial protein using feature selection techniques. *Mol Divers*. 2010;14:667-671.
73. Lee H, Tu Z, Deng M, Sun F, Chen T. Diffusion kernel-based logistic regression models for protein function prediction. *OMICS*. 2006;10:40-55.
74. Ni Q, Wang Z, Han Q, Li G, Wang X, Wang G. Using logistic regression method to predict protein function from protein-protein interaction data. Proceedings of IEEE International Conference on Bioinformatics and Biomedical Engineering Beijing, China, 2009.
75. Jiang J, McQuay L. Predicting protein function by multi-label correlated semi-supervised learning. *IEEE/ACM Trans Comput Biol Bioinform*. 2012;9:1059-1069.
76. Hu L, Huang T, Shi X, Lu WC, Cai YD, Chou KC. Predicting functions of proteins in mouse based on weighted protein-protein interaction network and protein hybrid properties. *PLoS One*. 2011;6:e14556.
77. You R, Huang X, Zhu S. DeepText2GO: improving large-scale protein function prediction with deep semantic text representation. *Methods*. 2018;145:82-90.
78. Zou Z, Tian S, Gao X, Li Y. mIDEEPre: multi-functional enzyme function prediction with hierarchical multi-label deep learning. *Front Genet*. 2019;9:714.
79. Rice SB, Nenadic G, Stapley BJ. Mining protein function from text using term-based support vector machines. *BMC Bioinformatics*. 2005;6:S22.
80. Wong A, Shatkay H. Protein function prediction using text-based features extracted from the biomedical literature: the CAFA challenge. *BMC Bioinformatics*. 2013;14:S14.
81. Zheng W, Blake C. Using distant supervised learning to identify protein subcellular localizations from full-text scientific articles. *J Biomed Inform*. 2015;57:134-144.
82. Funk CS, Kahanda I, Ben-Hur A, Verspoor KM. Evaluating a variety of text-mined features for automatic protein function prediction with GOstruct. *J Biomed Semant*. 2015;6:9.
83. Shao W, Liu M, Zhang D. Human cell structure-driven model construction for predicting protein subcellular location from biological images. *Bioinformatics*. 2016;32:114-121.
84. Fa R, Cozzetto D, Wan C, Jones DT. Predicting human protein function with multi-task deep neural networks. *PLoS One*. 2018;13:e0198216.
85. Molina L, Belanche L, Nebot A. Feature selection algorithms: a survey and experimental evaluation. Proceedings of IEEE International Conference on Data Mining Maebashi City, Japan, 2002. p. 306–313.
86. Saeys Y, Inza I, Larranaga P. A review of feature selection techniques in bioinformatics. *Bioinformatics*. 2007;23:2507-2517.
87. Wang L, Wang Y, Chang Q. Feature selection methods for big data bioinformatics: a survey from the search perspective. *Methods*. 2016;111:21-31. http://www.sciencedirect.com/science/article/pii/S1046202316302742 big Data Bioinformatics.
88. Frank E, Hall MA, Witten IH. The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques". Morgan Kaufmann; 2016.
89. Tang H, Zhao YW, Zou P, et al. HBPred: a tool to identify growth hormone-binding proteins. *Int J Biol Sci*. 2018;14:957-964.
90. Al-Shahib A, Breitling R, Gilbert DR. Franksum: new feature selection method for protein function prediction. *Int J Neural Syst*. 2005;15:259-275.
91. Rahman S, Bakar A, Hussein Z. Feature selection and classification of protein subfamilies using rough sets. Proceedings of IEEE International Conference on Electrical Engineering and Informatics Selangor, Malaysia, 2009. p. 32–35.
92. Ding C, Peng H. Minimum redundancy feature selection from microarray gene expression data. Proceedings of IEEE Conference on Computational Systems Bioinformatics Stanford, USA, 2003.
93. Clark WT, Radivojac P. Analysis of protein function and its prediction from amino acid sequence. *Proteins*. 2011;79:2086-2096.
94. Moreira IS, Koukos PI, Melo R, et al. SpotOn: high accuracy identification of protein-protein interface hot-spots. *Sci Rep*. 2017;7:8007.
95. Santos BD, Nobre C, Zarate L. Multi-objective genetic algorithm for feature selection in a protein function prediction context. Proceedings of IEEE Congress on Evolutionary Computation Rio de Janeiro, 2018.
96. Fodeh S, Tiwari A, Yu H. Exploiting PubMed for protein molecular function prediction via NMF based multi-label classification. Proceedings of IEEE International Conference on Data Mining Workshops New Orleans, USA, 2017. p. 446–451.
97. Maheshwari S, Brylinski M. Prediction of protein-protein interaction sites from weakly homologous template structures using metathreading and machine learning. *J Mol Recognit*. 2015;28:35-48.
98. Fabris F, Freitas A. An efficient algorithm for hierarchical classification of protein and gene functions. Proceedings of IEEE International Workshop on Database and Expert Systems Applications Munich, Germany, 2014. p. 64–68.
99. Merschmann L, Freitas A. *An Extended Local Hierarchical Classifier for Prediction of Protein and Gene Functions*. Berlin: Springer; 2013.
100. Boser B, Guyon I, Vapnik V. A training algorithm for optimal margin classifiers. Proceedings of 5th Annual ACM workshop on computational learning theory. Proceedings of 5th Annual ACM Workshop

on Computational Learning Theory Pittsburgh, Pennsylvania, USA, 1992. p. 144–152.
101. Cai YD, Liu XJ, Xu X, Zhou GP. Support vector machines for predicting protein structural class. *BMC Bioinformatics*. 2001;2:3.
102. Lanckriet GR, Deng M, Cristianini N, Jordan MI, Noble WS. Kernel-based data fusion and its application to protein function prediction in yeast. Pacific Symposium on Biocomputing Hawaii, USA, 2004. p. 300–311.
103. Cozzetto D, Minneci F, Currant H, Jones DT. FFPred 3: feature-based function prediction for all Gene Ontology domains. *Sci Rep*. 2016;6:31865.
104. Tang H, Zou P, Zhang C, Chen R, Chen W, Lin H. Identification of apolipoprotein using feature selection technique. *Sci Rep*. 2016;6:30441.
105. Zhang SB, Tang QR. Predicting protein subcellular localization based on information content of gene ontology terms. *Comput Biol Chem*. 2016;65:1-7.
106. Badal VD, Kundrotas PJ, Vakser IA. Natural language processing in text mining for structural modeling of protein complexes. *BMC Bioinformatics*. 2018;19:84.
107. Miranda L, Hu J. A deep learning approach based on stacked denoising autoencoders for protein function prediction. Proceedings of IEEE 42nd Annual Computer Software and Applications Conference Tokyo, Japan, 2018. p. 480–485.
108. Freund Y, Shapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. *J Comput Syst Sci*. 1997;55:119-139.
109. Friedman JH. Greedy function approximation: a gradient boosting machine. *Ann Stat*. 2001;29:1189-1232.
110. Chen T, Guestrin C. XGBoost: a scalable tree boosting system. Proceedings of the 22nd ACM Conference on Knowledge Discovery and Data Mining San Francisco, USA, 2016. p. 785–794.
111. Breiman L. Random forests. Machine Learning, 2001.
112. Peled S, Leiderman O, Charar R, Efroni G, Shav-Tal Y, Ofran Y. De-novo protein function prediction using DNA binding and RNA binding proteins as a test case. *Nat Commun*. 2016;7:13424.
113. Yu G, Rangwala H, Domeniconi C, Zhang G, Yu Z. Protein function prediction using multilabel ensemble classification. *IEEE/ACM Trans Comput Biol Bioinform* 2013;10:1045–1067.
114. Guo X, Liu F, Ju Y, Wang Z, Wang C. Human protein subcellular localization with integrated source and multi-label ensemble classifier. *Sci Rep*. 2016;6:28087.
115. Quinlan J. *C4.5: Programs for Machine Learning*. Boston: Morgan Kaufmann Publishers; 1993.
116. Cerri R, Basgalupp M, Mantovani R, de Carvalho A. Multi-label feature selection techniques for hierarchical multi-label protein function prediction. Proceedings of IEEE International Joint Conference on Neural Networks Rio de Janeiro, Brazil, 2018.
117. Vens C, Struyf J, Shetgat L, Dzeroski S, Blockeel H. Decision trees for hierarchical multi-label classification. *Mach Learn*. 2008;73:185-214.
118. Yang J, Yang M. Assessing protein function using a combination of supervised and unsupervised learning. Proceedings of IEEE Symposium on Bioinformatics and Bioengineering Arlington, USA, 2006. p. 35–44.
119. Cerri R, Barros RC, de Carvalho A, Jin Y. Reduction strategies for hierarchical multi-label classification in protein function prediction. *BMC Bioinformatics*. 2016;17:373.
120. Nievola J, Paraiso E, Freitas A. A hierarchical neural network for predicting protein functions. Proceedings of IEEE International Conference on Bioinformatics and Bioengineering Belgrade, Serbia, 2015.
121. Borges H, Nievola J. Multi-label hierarchical classification using a competitive neural network for protein function prediction. Proceedings of International Joint Conference on Neural Networks Brisbane, Australia, 2012. p. 172–177.
122. Smale S, Rosasco L, Bouvrie J, Caponnetto A, Poggio T. Mathematics of the neural response. *Found Comput Math*. 2010;10:67-91.
123. LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature*. 2015;521:436-444.
124. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. *Proc IEEE*. 1998;86:2278-2324.
125. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition; 2015.
126. Krizhevsky A, Sutskever I, Hinton G. ImageNet classification with deep convolutional neural networks. Proceedings of Neural Information Processing Systems Conference Lake Tahoe, USA, 2012. p. 1106–1114.
127. Zacharaki E. Prediction of protein function using a deep convolutional neural network ensemble. *PeerJ Comput Sci*. 2017;3:e124.
128. Pearlmutter B. Learning state space trajectories in recurrent neural networks. *Neural Comput*. 1989;1:263-269.
129. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput*. 1997;9:1735-1780.
130. Graves A, Mohamed A, Hinton G. Speech recognition with deep recurrent neural networks. Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing Vancouver, Canada, 2013. p. 6645–6649.
131. Quang D, Xie X. DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences. *Nucleic Acids Res*. 2016;44:e107.
132. Cao R, Freitas C, Chan L, Sun M, Jiang H, Chen Z. ProLanGO: protein function prediction using neural machine translation based on a recurrent neural network. *Molecules*. 2017;22:E1732.
133. Wu JS, Huang SJ, Zhou ZH. Genome-wide protein function prediction through multi-instance multi-label learning. *IEEE/ACM Trans Comput Biol Bioinform*. 2014;11:891-902.
134. Dietteric R, Lathrop R, Lozano-Perez T. Solving the multiple instance learning problem with axis-parallel rectangles. *Artif Intell*. 1997;89:31-71.
135. Zhou Z, Zhang M, Huang S, Li Y. Multi-instance multi-label learning. *Artificial Intelligence. Artificial Intelligence*. 2012;176:2291-2320.
136. Pawlak Z. Rough sets. *Int J Comput Inf Sci*. 1982;11:341-356.
137. Liu L, Tang L, He S, Yao S, Zhou W. Predicting protein function via multi-label supervised topic model on gene ontology. *Biotechnol Biotechnol Equip*. 2017;31:630-638.
138. Ramage D, Hall D, Nallapati R, Manning C. Labeled LDA: a supervised topic model for credit attribution in multi-labeled corpora. Proceedings of Conference on Empirical Methods in Natural Language Singapore, 2009. p. 248–256.
139. Cheng X, Lin WZ, Xiao X, Chou KC. pLoc_bal-mAnimal: predict subcellular localization of animal proteins by balancing training dataset and PseAAC. *Bioinformatics*. 2019;35:398-406.
140. Pedregosa F et al. Scikit-learn: machine learning in Python. *J Mach Learn Res*. 2011;12:2825-2830.
141. Spectrum I, The Top Programming Languages in 2018; 2018. https://spectrum.ieee.org/static/interactive-the-top-programming-languages-2018.
142. The MathWorks I, MATLAB and Statistics Toolbox Release 2018b; 2018.
143. Adabi M, et al. TensorFlow: a system for large-scale machine learning. Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation Savannah, USA, 2016. p. 265–283.
144. Chollet F, et al.; 2015. https://keras.io.
145. Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, et al. Caffe: convolutional architecture for fast feature embedding. Proceedings of ACM International Conference on Multimeda Orlando, USA, 2014. p. 675–678.
146. Paszke A, et al. Automatic differentiation in PyTorch. Proceedings of Neural Information Processing Systems Conference. Proceedings of Neural Information Processing Systems Conference Long Beach, USA, 2017.

147. Kingma D, Ba J. Adam: a method for stochastic optimization. Proceedings of International Conference on Learning Representations San Diego, USA, 2015.
148. Tielman T, Hinton G. Lecture 6.5 - rmsprop: Divide the Gradient by a Running Average of its Recent Magnitude, 2012.
149. Mercado-Diaz L, Navarro-Garcia J, Jaramillo-Garzon J. A comparison of class-balance strategies for SVM in the problem of protein function prediction. Proceedings of 20th Symposium on Signal Processing, Images and Computer Vision Bogota, Colombia, 2015.
150. Lu Z, Szafron D, Greiner R, et al. Predicting subcellular localization of proteins using machine-learned classifiers. *Bioinformatics*. 2004; 20:547-556.
151. Li Y, Maleki N, Carruthers N, Rueda L, Stemmer P, Ngom A. Prediction of calmodulin-binding proteins using short-linear motifs. Proceedings of International Conference on Bioinformatics and Biomedical Engineering Granada, Spain, 2017. p. 107–117.
152. Santis ED, Martino A, Rizzi A, Mascioli F. Dissimilarity space representation and automatic feature selection for protein function prediction. Proceedings of International Joint Conference on Neural Networks Rio de Janeiro, Brazil, 2018.
153. Ahmed I, Witbooi P, Christoffels A. Prediction of human-*Bacillus anthracis* protein-protein interactions using multi-layer neural network. *Bioinformatics*. 2018;34:4159-4164.
154. Ruepp A, Zollner A, Maier D, et al. The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes. *Nucleic Acids Res*. 2004;32:5539-5545.
155. Nomenclature Committee of the International Union of Biochemistry and Molecular Biology on the Nomenclature and Classification of Enzymes. *Enzyme Nomenclature*. San Diego, CA: Elsevier; 1992.
156. Zhou N, Jiang Y, Bergquist TR, Lee AJ, Kacsoh BZ, Crocker AW, et al. The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. bioRxiv 2019;https://www.biorxiv.org/content/early/2019/05/29/653105.
157. Wu J, Zhu W, Jiang Y, Sun G, Gao Y. Predicting protein functions of bacteria genomes via multi-instance multi-label active learning. Proceedings of IEEE International Conference on Integrated Circuits and Microsystems Shanghai, China 2018. p. 302–307.
158. Tung CH, Chen CW, Sun HH, Chu YW. Predicting human protein subcellular localization by heterogeneous and comprehensive approaches. *PLoS One*. 2017;12:e0178832.
159. Youngs N, Penfold-Brown D, Bonneau R, Shasha D. Negative example selection for protein function prediction: the NoGO database. *PLoS Comput Biol*. 2014;10:e1003644.
160. Bergstra J, Yamins D, Cox DD. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28 ICML'13, JMLR.org; 2013. p. I–115–I–123. http://dl.acm.org/citation.cfm?id=3042817.3042832.

How to cite this article: Bonetta R, Valentino G. Machine learning techniques for protein function prediction. *Proteins*. 2020;88:397–413. https://doi.org/10.1002/prot.25832