Neural Network and Random Forest Models in Protein Function Prediction

Kai Hakala, Suwisa Kaewphan, Jari Björne, Farrokh Mehryary, Hans Moen, Martti Tolvanen, Tapio Salakoski, and Filip Ginter

Abstract—Over the past decade, the demand for automated protein function prediction has increased due to the volume of newly sequenced proteins. In this paper, we address the function prediction task by developing an ensemble system automatically assigning Gene Ontology (GO) terms to the given input protein sequence. We develop an ensemble system which combines the GO predictions made by random forest (RF) and neural network (NN) classifiers. Both RF and NN models rely on features derived from BLAST sequence alignments, taxonomy and protein signature analysis tools. In addition, we report on experiments with a NN model that directly analyzes the amino acid sequence as its sole input, using a convolutional layer. The Swiss-Prot database is used as the training and evaluation data. In the CAFA3 evaluation, which relies on experimental verification of the functional predictions, our submitted ensemble model demonstrates competitive performance ranking among top-10 best-performing systems out of over 100 submitted systems. In this paper, we evaluate and further improve the CAFA3-submitted system. Our machine learning models together with the data pre-processing and feature generation tools are publicly available as an open source software at https://github.com/TurkuNLP/CAFA3.

Index Terms—Sequence analysis, protein function prediction, neural network, convolutional neural network, random forest

1 Introduction

Proteins play a pivotal role in many processes of living organisms, including, but not limited to, signal transduction, transmembrane transport and structural support. Determining protein functions experimentally is an expensive and labor-intensive undertaking. With the increasing number of sequences produced by high throughput sequencing methods, there is an urgent need for computational methods to assist in protein function annotation. Over the past decade, a research community focusing on automated function prediction (AFP) has formed, resulting in a number of AFP systems and the regular Critical Assessment of Functional Annotation (CAFA) challenge.

CAFA is a shared task organized by the AFP-Special Interest Group, aiming to establish a common platform and evaluation methods for measuring the performance of automated systems for the AFP task [1], [2]. In CAFA, each participating research group is asked to predict the functions for a large set of proteins with more than 100,000 individual sequences, under a tight time limit. Subsequently, the organizers gather experimental evidence to verify functional annotations for a subset of the sequences, usually within half a year after the predictions were submitted. The resulting verified annotations constitute a new test set, against which the predictions made by the participants can be evaluated.

In CAFA, the protein functions are assigned from the controlled vocabulary of terms defined in Gene Ontology (GO), i.e., the task is to annotate the given protein sequences with the relevant GO terms that describe their function. In addition to molecular functions (MF), GO includes terms related to the cellular components (CC) in which the proteins are active, and the biological processes (BP) such as pathways in which the proteins participate [3], [4]. All together, over 40,000 terms, organized in a hierarchy, exist in the current version of GO. This translates into a multi-class and multi-label classification task, where each protein sequence can be annotated with multiple terms from this large vocabulary, with strong statistical dependencies between the terms. Further, the distribution of GO terms is highly skewed in several distinct ways, as demonstrated in Swiss-Prot, a subset of the UniProt protein database [5] manually curated with GO terms. While for instance the human proteins are densely annotated with 20 GO terms on average, full 36 percent of GO terms do not have a single annotated example, and 18 percent have only one. This means that a proportionally small number of unique GO terms account for a large proportion of the annotations. All these factors combined make AFP a very challenging task from the machine learning perspective.

- *Kai Hakala, Farrokh Mehryary, and Tapio Salakoski are with the Department of Future Technologies, University of Turku, 20500 Turku, Finland, and also with the University of Turku Graduate School (UTUGS), 20500 Turku, Finland. E-mail: {kai.hakala, farrokh.mehryary, tapio.salakoski}@utu.fi.*
- *Jari Björne, Hans Moen, Martti Tolvanen, and Filip Ginter are with the Department of Future Technologies, University of Turku, 20500 Turku Finland, and also with the University of Turku Graduate School (UTUGS), 20500 Turku, Finland. E-mail: {jari.bjorne, hans.moen, martti.tolvanen, filip.ginter}@utu.fi.*
- *Suwisa Kaewphan is with the Department of Future Technologies, University of Turku, 20500 Turku, Finland, and with the University of Turku Graduate School (UTUGS), 20500 Turku, Finland, and also with the Turku Centre for Computer Science (TUCS), 20500 Turku, Finland. E-mail: suwisa.kaewphan@utu.fi.*

Manuscript received 4 Sept. 2019; revised 9 Nov. 2020; accepted 26 Nov. 2020. Date of publication 11 Dec. 2020; date of current version 3 June 2022. (Corresponding author: Kai Hakala.)
Digital Object Identifier no. 10.1109/TCBB.2020.3044230

The first two CAFA challenges have seen a variety of approaches applied to the problem [1], [2]. Among the top performing systems, the most common approach was the annotation transfer by homology, combined with a statistical or machine-learned scoring function. Cozetto *et al.* [6] uses a scoring function to combine and rank the predictions from various biological data analyses, including PSI-BLAST [7], profile-profile comparison, text mining, sequence features, protein-protein interactions and high-throughput data. This approach was ranked as the best performing system in CAFA1, which evaluated only the *molecular function* and *biological process* GO ontologies. The PANNZER method [8] uses weighted k-nearest neighbor to predict protein functions from the weighted sequence similarity scores using BLAST [9] search, and the taxonomic distance of the organisms originating the sequences. Argot2 [10] uses the semantic similarity of weighted ontology terms found through the BLAST sequence similarity search and the HMMER [11] tools. In CAFA2, GO-FDR [12], one of the best performing systems on all three GO ontologies, calculates the probability of a protein being associated with a target GO term, using predictions from the PSI-BLAST tool.

Recently, You *et al.* [13] suggested an approach based on an ensemble of logistic regression models, which resulted in the best overall performance among the participating teams in the CAFA3 challenge. For each GO term, a set of three logistic regression models are independently trained based on structural information from InterPro [14], biophysical attributes from ProFET [15] and amino acid n-gram features. Sequence alignment and GO annotation frequencies are used as additional features. All this information is aggregated by a separate machine learning (ML) model in a learning-to-rank setting.

Deep neural network architectures have been successfully applied to bioinformatics problems, such as fold recognition, functional classification and protein design [16], [17]. For the protein function prediction task, several architectures have been developed, aiming to replace hand-crafted features with ones directly extracted from the sequences. DeepGO [18] uses a deep convolutional neural network (CNN) architecture composed of 10 hidden layers, with inputs formed from amino acid tri-grams and the embedding (continuous vector representation) of the protein induced using its neighborhood in the STRING [19] database graph. ProLanGO [20] uses a neural machine translation system based on a recurrent neural network to "translate" proteins into the corresponding GO terms. Here the protein sequences are first converted to non-overlapping amino acid k-mers of length 3 to 5, forming the input sequence for the model. [21] use multi-task deep neural networks with features directly derived from the amino acid sequences as well as external analysis tools, focusing on human proteins only. In general, these deep neural network models have demonstrated a competitive performance, on par with other machine learning approaches with less feature engineering effort.

Random forests, an ensemble of decision trees, have been used in a wide array of prediction tasks, including post-translational modification site prediction [22], fold recognition [23], SCOP structural classification [24], protein-protein interaction site prediction [25], and enzyme function classification [26]. Most models derive protein-related features, such as hydrophobicity, secondary structure, and amino acid composition, both by directly extracting the features from the sequence and by relying on established sequence analysis tools. Random forests have been a popular machine learning method in bioinformatics due to its simplicity in terms of modeling and interpretability of the results through feature importance analysis.

In this paper, we introduce our system based on neural network and random forest classifiers, both relying on a rich set of features and, individually, achieving competitive performance. Further, through an evaluation on our internal test dataset, we show that these individual approaches strengthen each others performance, resulting in an ensemble system outperforming the two individual classifiers. For comparison, we also experiment with a neural network relying purely on the sequence itself, in order to evaluate the feature generation ability of the neural networks. Most importantly, this neural network does not have access to the manually curated annotations of homologous proteins, a primary source of features for all systems performing well on the AFP task, nor to the biologically motivated features produced by established sequence analysis tools.

The official CAFA3 evaluation places our ensemble system in top-10 overall out of 68 participating teams and 144 submitted systems, with particularly strong performance on molecular function and cellular component categories of prokaryotic proteins, where the system placed 3rd and 2nd [27].

2 Methods

In this section we describe the protein datasets, sequence analyses used for generating features and the details of the neural network and random forest classifiers and the ensemble systems.

2.1 Protein Dataset

As the *training data*, we selected only those Swiss-Prot proteins, whose function annotation is based on a reliable experimental evidence, which we define as evidence codes `EXP`, `IDA`, `IPI`, `IMP`, `IGI`, `IEP`, the author statement code `TAS`, or the curatorial statement evidence code `IC` being assigned to the annotation. This restriction aims to discard annotations stemming from high-throughput and other noise-prone experimental techniques. The training data consists of 387,416 individual annotations of 67,118 proteins, from the total of 738,431 annotations of 112,279 proteins in Swiss-Prot that have other than purely computationally predicted annotation.

In addition to the exact GO terms manually assigned to a particular protein, we enriched the annotations by additionally assigning the ancestral terms from the ontology to the proteins, increasing the number of individual annotations to 3,955,953. As a result, the proteins which are on average annotated with 6 terms, now become annotated with 60 terms.

2.2 Sequence Analysis and Protein Features

In this section, we summarize the sequence-based features as used by the various classifiers throughout the paper.

2.2.1 BLAST Features

We use the Protein-Protein BLAST (BLASTP) program from a locally installed NCBI-BLAST+ version 2.5.0 [28] to search for similar proteins from the full Swiss-Prot database. The E-value of 0.001 is used as the inclusion threshold. We query with both BLOSUM45 and BLOSUM62 [29] scoring matrices, resulting in two distinct sets of similar proteins for each query sequence. These will be referred to as *blast45* and *blast62* hereafter. We use the default gap scores for each of the scoring matrices.

We also use DELTA-BLAST (Domain Enhanced Lookup Time Accelerated) for searching distantly related protein sequences [30]. DELTA-BLAST increases the sensitivity of BLAST-based sequence similarity search by constructing position-specific score matrices from the conserved domain database (CDD) [31]. BLOSUM62 is used as the scoring matrix with 0.001 as the cut-off E-value.

All BLAST data was converted to feature vectors with the following approach: The Uniprot ID of the matched protein was used as the feature name and the HSP (High-scoring Segment Pair) score as its value.

2.2.2 InterproScan

InterProScan [32] is a software package predicting structural motifs, functional domains, signatures, protein families and other features relevant to protein function analysis, based on the InterPro database. We use locally installed InterProScan 5 to predict InterPro profiles and, subsequently, GO terms for those cases where a mapping between an InterPro profile and GO is established in the database. These mappings are however neither up-to-date nor complete. To increase the coverage of the mappings, we trace the patterns and signatures to InterPro's upstream databases, and recover the GO terms from there. Features are generated from the matching GO terms (or a special feature is produced, signalling the absence of such mapping) such that scores given by InterProScan are converted to numerical features while the *GO* terms and profile *accession* identifiers are converted to binary features.

2.2.3 Taxonomy Features

The NCBI Taxonomy is a manually curated database of names and taxonomic lineages for organisms within the scope of the International Nucleotide Sequence Database Collaboration (INSDC) [33]. A binary feature is generated for each node in the NCBI Taxonomy and subsequently assigned to each protein based on its organism of origin.

2.2.4 Sequence Features

We use several additional tools to analyze the protein sequence and provide features potentially relevant to its function.

For nuclear localization, we use NucPred [34] to predict whether a protein enters the nuclear compartment. This analysis applies eukaryotic sequences only, as prokaryotes do not have a nucleus. For post-translational modifications, we use NetAcet [35] to predict proteins which are acetylated by N-acetyltransferase A (NatA). For GPI-anchored proteins, we use PredGPI [36] which is based on a support vector machine and a Hidden Markov Model predicting the anchoring signal and the most probable omega-site. The predictions of these three tools are encoded as numeric features.

2.2.5 Amino Acid Index

The Amino Acid Index is a set of numerical values characterizing the physicochemical and biochemical properties of each of the 20 amino acids. It has been used for numerous structure-function prediction tasks, e.g., human protein sub-cellular localization [37]. In this work, we obtain 544 amino acid indices from the Amino Acid Index database [38] and use their numerical values as features in our experiments with the sequence-only convolutional neural network (See Section 2.4).

2.3 Experimental Setting

In all experiments the input data for each protein was converted into numerical feature vectors and the GO terms into binary label vectors, one binary value for each GO term. We subsequently randomly divide the training data into three parts: The *training* set used for training the classifiers, the *validation* set used for hyperparameter optimization and the *test* set which is used for the final performance estimation. The validation set is also used during system development and for experiments, including feature selection, in order to avoid overfitting the test set. The training, validation and test subsets contain 60, 20 and 20 percent of the whole data.

Different feature groups were tested in combination to select the ones that gave the best classification performance. Finally, for computational reasons, the classifiers are trained to predict the 5,000 most common terms, a subset of GO which covers over 94 percent of all GO annotations in Swiss-Prot. We do not filter the targeted GO terms based on their depth in the GO hierarchy, but try to predict terms from all levels, if they belong to the 5,000 most common terms. Note that during testing, the classifiers are evaluated on the full set of GO terms (29,000 GO terms), i.e., all GO terms the models are unable to predict are counted as false negative errors. Thereby the 5,000 term subsetting choice does not artificially overestimate the performance.

We evaluate the performance of the systems using the F-score, unlike in the official CAFA evaluation, where the maximal F-score, calculated from precision-recall curves, is used as the primary metric. Thus, our internal evaluation is more strict and acts as a lower bound for the maximal F-score.

2.4 Feedforward Neural Network Classifier

As the first classifier in the ensemble, we train a standard feedforward neural network receiving a vector of the features described in Section 2.2. The feature values are scaled by dividing them by the maximum absolute value observed in the *training* set for the given feature, before they are utilized in the model. This scaling has favorable computational implications, as zero values are preserved as such, and feature matrices can be stored in the sparse format. To further reduce the computational cost, we reduce the number of features by removing those with variance below 0.0001 after scaling.

The input features are passed to a fully connected layer with dimensionality of 300 and hyperbolic tangent activation function before the output layer, which is a standard dense layer of dimensionality 5,000, corresponding to the number of unique output GO terms. Since the learning task is multi-label, i.e., the classifier can predict several classes for each instance, the output layer is trained using the binary cross entropy objective.

The network is regularized by applying a dropout with rate 0.5 to the input features [39]. The training is stopped early, once the *validation* set F-score is no longer improving.

The substantial imbalance towards the negative class in the multi-label mode of training results in a high precision and low recall model. To mitigate this, the recall is boosted by penalizing false negative predictions more than false positive ones. The magnitude of the penalty is a hyperparameter selected to optimize the performance on the *validation* dataset.

2.5 Random Forest Method

Random forests [40] are a prediction algorithm based on an ensemble of decision trees. Each decision tree in the ensemble is built based on a different random subset of the input features, and the final prediction of the ensemble is the majority vote among the trees. Random forests are particularly suitable for the current task as they have good classification performance and support multi-label classification on thousands of labels. The classifier was implemented using scikit-learn library version 0.18.1 [41].

2.6 Convolutional Neural Network

While the previous two methods can be seen as traditional classifiers relying on a carefully selected set of features, the third method, convolutional neural networks, will depart from this paradigm in not being presented with any complex features. Rather, the neural network is presented with the sequence itself as its sole input.

Convolutional neural networks (CNNs) have been successfully applied to sequential and multidimensional prediction tasks, especially in computer vision and text classification [42], [43]. In biological sequence analysis, they have been applied specifically to protein secondary structure prediction [44]. The strength of CNNs arises from the possibility of detecting fuzzy patterns locally from the input data, e.g., in our case we expect the convolutional kernels learn to detect amino acid n-grams relevant to a certain protein function.

Many suggested systems for AFP rely on structural information such as the presence of a certain domain or motif. However, this type of information is mostly gathered from other tools such as InterProScan, or by simply looking at the amino acid n-grams present in a given sequence, resulting in extremely sparse feature representations [18]. To overcome this issue, our neural network learns a latent feature vector (embedding) for each unique amino acid, and a protein is represented as a sequence of these embeddings. This gives the model an opportunity to measure whether two different amino acids tend to have a similar role in the sequence, leading to similar embeddings. To ease the task of learning these embeddings, we attempted to initialize the weights with the Amino Acid Index properties, but this did not have a significant influence on the model performance.

We use convolution window sizes of 3, 9, 27 and 81 amino acids and learn 50 kernels for each window size. The shortest window size of 3 is a common length used in amino acid n-gram features, whereas size 9 approximates the length of local segments analyzed in protein secondary structure prediction [44]. The larger window sizes of 27 and 81 should in turn be able to detect motifs and shorter domains [45].

The convolutional kernel activations are subsequently pooled by taking the maximum activations of each kernel across all positions in the sequence (max pooling). This procedure removes the location information of the detected patterns, producing a fixed-length, position-invariant model suitable as an input to a classification layer. A prior study shows that trying to preserve the location information with local max pooling does not provide any benefits in DNA-protein binding prediction [46], hence we have not experimented with other possible settings.

These maximum kernel activations then form an input of a fully connected output layer, producing the final predictions. The dimensionality of the output layer is 5,000, corresponding to the number of predicted GO terms.

Proteins longer than 2500 amino acids are truncated, i.e., we analyze only the first 2500 amino acids of each sequence. This truncation affects only 1 percent of the training sequences.

2.7 Homology Transfer

In our previous work on the CAFA2 challenge [2], we observed that oftentimes proteins received no prediction from the classifiers, despite there being homologous proteins with existing annotation. Moreover, our classifiers are trained with only the top-5000 most common annotated terms, leaving less frequent terms unattended. To address the issue, we use the following simple fallback homology-based transfer approach, inferring the functions from homologous proteins. For each protein, we extract the GO terms associated with homologous sequences from the *blast62* alignment, i.e., all sequences identified as similar at the BLAST E-value cut-off of 0.001. We subsequently rank the ontology terms by the number of associated homologous sequences, and the top 5 terms which are supported by at least 2 sequences form the prediction of the Homology Transfer fallback method.

2.8 Ensemble

As the final, combined output of the abovementioned methods, we take the union of their predicted GO term sets, i.e., if even one of the models has made a positive prediction, the given GO term is included in our final output. We evaluate all the model combinations and use the subset of models that leads to the best performance measured in terms of F-score as the final system. For completeness, we also report on results obtained using the intersection of the predictions, a distinctly high precision, low recall model. The whole system architecture is visualized in Fig. 1.

3 Results

In this section, we evaluate the methods from several distinct angles as well as report on the ranking of the ensemble system in the CAFA3 challenge.

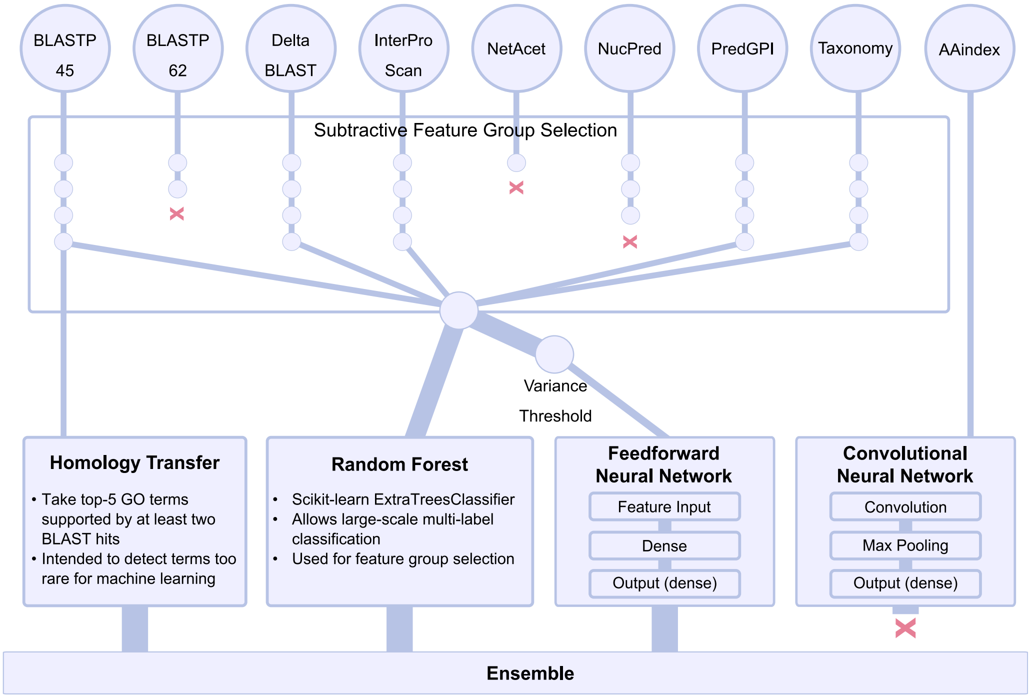
BLASTP 45
BLASTP 62
Delta BLAST
InterPro Scan
NetAcet
NucPred
PredGPI
Taxonomy
AAindex
Subtractive Feature Group Selection
Variance Threshold
Homology Transfer
• Take top-5 GO terms supported by at least two BLAST hits
• Intended to detect terms too rare for machine learning
Random Forest
• Scikit-learn ExtraTreesClassifier
• Allows large-scale multi-label classification
• Used for feature group selection
Feedforward Neural Network
Feature Input
Dense
Output (dense)
Convolutional Neural Network
Convolution
Max Pooling
Output (dense)
Ensemble

Fig. 1. The architecture of our ensemble system, based on three approaches, homology transfer, random forest and neural network. Features from Blast62, NetAcet and NucPred analyses are removed from the final feature sets. The final set of features are used in both random forest and machine learning systems. The final ensemble system is a union (boolean *OR* combination) of the predictions from the three approaches.

3.1 Feature Group Selection

Feature selection is a process of selecting a subset of relevant features, or removing irrelevant features, in order to simplify the system, reduce the training time and improve the generalization of the models [47]. We performed feature selection by repeatedly removing one feature group at a time, so as to increase the overall performance on the development dataset, stopping when the performance of the classifier no longer increased (see Supplementary File, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TCBB.2020.3044230, for detailed results). The optimal performance using the random forest classifier on the development data was achieved by removing the *blast62*, *netacet* and *nucpred* feature groups. The same feature subset is used also for the neural model.

Among the remaining features, *taxonomy* contributes the most to the system performance, i.e., removing *taxonomy* features results in a substantial drop in F-score. For BLAST-based homology features, the fact that the *blast45* features outperformed the *blast62* features can be attributed to the BLOSUM45 matrix allowing more distant hits, thus increasing the recall compared to BLOSUM62. These results also emphasize that choosing the right scoring matrix can have an impact on the performance of the classifiers. In general, the selection of a BLAST scoring matrix depends on the proteins at hand and the downstream application.

3.2 Method Evaluation

We evaluate the performance of all systems against the test dataset using the precision/recall/F-score metric for both single and ensemble systems. The evaluation results on all of the 29,190 unique annotated GO terms are summarized in Table 1.

The performances of all the approaches, feedforward neural network (FFN), convolutional neural network (CNN), random forest classifier (RF) and homology transfer (HT) are shown in Table 1. The performance of all classifiers, except for CNN, surpass HT that is based solely on inferring known protein functions from similar sequences. FFN is the best performing method with an F-score of 0.480, followed by RF with 0.424.

The union of the predictions of the individual methods outperforms each method individually. The best result in terms of F-score is achieved by combining the predictions of FFN and RF. This demonstrates that the classifiers, even though provided with the same features, learn different aspects of the task. Also, as the RF classifier performs at the

TABLE 1
The Performance of all System Combinations in This Work

NN	RF	HT	mode	F	P	R
CNN				0.347	0.316	0.385
FFN				**0.480**	0.492	0.468
	RF			0.424	0.609	0.326
		HT		0.387	0.550	0.298
FFN	RF		OR	**0.493**	0.472	0.517
FFN		HT	OR	0.487	0.460	0.518
	RF	HT	OR	0.471	0.527	0.426
FFN	RF		AND	0.398	0.707	0.277
FFN		HT	AND	0.363	0.675	0.248
	RF	HT	AND	0.312	0.740	0.198
FFN	RF	HT	OR	**0.493**	0.445	0.553
FFN	RF	HT	AND	0.296	0.765	0.184

All ensemble combinations of the three methods: random forest classifier (RF), feedforward neural network (FFN), and the homology transfer (HT), using either intersection AND *or union* OR *of the predictions. The performance is evaluated as the micro-averaged F-score (F), precision (P) and recall (R).*

TABLE 2
The Performance of the Submitted Systems to the CAFA3 Challenge

NN	RF	HT	mode	F	P	R
NN				0.449	0.485	0.419
	RF	HT	OR	0.471	0.527	0.426
NN	RF	HT	OR	**0.483**	0.442	0.532

high precision - low recall point, it adds a small number of predictions, which are on average more likely correct than FFN, thereby benefiting the overall numerically stronger FFN method. Even though adding the predictions from HT improves F-score of both RF (+4.7pp) and FFN (+0.7pp) methods in isolation, it no longer improves the F-score of the RF-FFN ensemble. As expected, the intersection-based ensemble produces numerically inferior F-score as it drastically decreases the already low recall of the models. Nevertheless, this low recall is matched with a comparatively high precision, which could be a desirable property in some applications.

Finally, the CNN method has the lowest performance of the four methods in isolation, and also decreased the performance of all tested ensembles. These are therefore excluded from Table 2 (see Supplementary File, available online, for these results). Nevertheless, keeping in mind the minimal input information presented to the CNN classifier — the raw sequence itself — we find it very encouraging, even surprising that it can reach a performance which, in terms of F-score is roughly comparable to the HT method (CNN=0.347, HT=0.387) and a mere 15 percent points behind an ensemble of several strong methods with large feature sets. We have experimented with more complex CNN architectures, but neither increasing the kernel window, nor stacking several convolutional layers resulted in an improvement.

3.3 CAFA3

The ensemble system as submitted to the CAFA3 challenge differs from the methods above in not treating the FFN and CNN methods independently. Rather the outputs from the max-pooled CNN layer and the first fully connected layer of the FFN method are concatenated and serve as an input to a single output layer. In subsequent experiments we however found that an improvement of +3.1pp F-score can be achieved by removing the CNN component. The performance of the system submitted to CAFA3 is reported in Table 2. Note that the values in Tables 1 and 2 are numerically comparable.

The primary evaluation of CAFA3 is based on maximal F-score obtained from precision-recall curves for predicted terms in each ontology. Our ensemble method has been ranked in the top-10 best performing systems for all three ontologies [27]. As shown in Fig. 2, competitive performance, differing from the best performing system by 1-3pp maximal F-score is achieved for ontologies of cellular component (0.60 versus 0.61) and biological process (0.37 versus 0.40). For molecular function, the top performing system [13], which is based on an ensemble of logistic regression classifiers, outperforms ours and other top performing systems by ten percentage points (0.52 versus 0.62). The performance of the participating systems are significantly lower for biological process ontology, which accounts for the majority of the annotations (4).

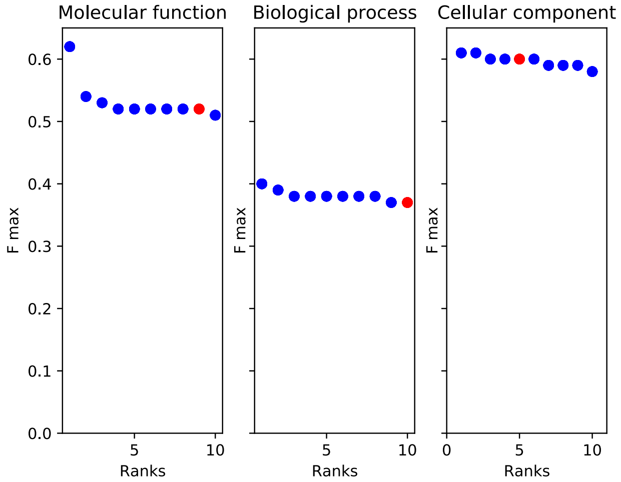

Fig. 2. The performance of the top-10 teams (out of 68 teams) for each ontology in the CAFA3 challenge. The red circles represent the performance of our submitted systems compared to the performance of the other systems (blue circle). The maximal F-score is shown on y-axis where the ranks of top-10 systems are shown on x-axis.

3.4 Comparison to DeepGO

After the CAFA3 challenge, others have pursued sequence models based on neural networks as well. As a point of comparison representing these NN-based systems, we report in Table 3 the results of the well-known DeepGO [18] system. We retrained DeepGO on our training dataset, and report its performance in Table 3. The DeepGO system is based on the CNN architecture, combining amino acid trigrams and protein-protein interaction network embeddings. The F-score of the DeepGO model trained with network and ngram embeddings is slightly lower than our best-performing system (0.493 versus 0.437) in terms of overall F-score. In Table 3, *ngrams* for DeepGo and *aa-index* for our system are predictions based purely on the sequence itself, while *Network* for

TABLE 3
The Performance of Our Systems Compared to DeepGO

System	features	Ontology	F	P	R
DeepGO	ngrams	MF	0.261	0.197	0.385
		BP	0.272	0.280	0.264
		CC	0.482	0.466	0.499
		All	0.313	0.303	0.324
our CNN	aa-index	MF	0.286	0.261	0.316
		BP	0.304	0.283	0.327
		CC	0.506	0.429	0.615
		All	0.347	0.316	0.385
DeepGO	Network	MF	0.414	0.393	0.438
		BP	0.384	0.432	0.346
		CC	0.604	0.577	0.634
		All	0.437	**0.464**	0.414
our ensemble	sequence feature	MF	**0.536**	0.480	**0.608**
		BP	**0.446**	0.411	**0.488**
		CC	0.620	0.530	**0.748**
		All	**0.493**	0.445	**0.553**

TABLE 4
The Distribution of Annotations Across the Three Different Ontologies

Ontology	Non-propagated		Propagated	
	[%]	count	[%]	count
MF	17.5%	67,903	9.3%	370,485
BP	52.7%	204,212	69.8%	2,763,702
CC	29.7%	115,301	20.7%	821,766
All	100%	386,693	100%	3,955,230

Non-propagated *refers to the distribution of direct annotations, while* propagated *refers to the distribution after the annotations are propagated to the root. The latter is the basis for the evaluation in the CAFA3 challenge.*

DeepGo and *sequence feature* for our system take into account also features derived through homology and other external knowledge about the sequence in question. As seen in Table 3, our system demonstrates better performance across all categories. Further, in Table 4 we report the class distribution, showing that Biological Process (BP) is the most frequent annotation in the data, followed by Cellular Component (CC) and Molecular Function (MF).

3.5 Evaluation on Model Organisms and Different Ontologies

The ultimate goal for automated function prediction is to develop universal and reliable algorithms capable of predicting the function of any protein sequence. This is of course a very difficult task. As shown in all CAFA challenges [1], [2], [27] the accuracy of the predictions varies greatly among the organisms and different ontologies. To look beyond overall system performance, we next focus on the results of our methods on these two aspects: ontologies and organisms.

There are 10 organisms with over 1,000 manually annotated sequences, hereafter called *model organisms*. The list includes both domains of life, eukaryota (*Arabidopsis thaliana*, *Mus musculus*, *Rattus norvegicus*, *Homo sapiens*, *Drosophila melanogastor*, *Caenorhabditis elegans*, *Schizosaccharomyces pombe* 972h- and *Saccharomyces cerevisiae* S288c,) and prokaryota (*Escherichia coli* K-12, *Mycobacterium tuberculosis* H37Rv). The number of annotated sequences from these 10 organisms, ranging from 1,500 for *M. tuberculosis* to 14,000 sequences for human, account for 83 percent of the whole protein dataset. We compare the performance of the classification methods with the homology transfer approach on the *model organisms* in the Swiss-Prot dataset.

On one hand, the improvement of the performance is only minor to moderate for multiple cellular organisms, ranging from 4pp to less than 15pp. As shown in Fig. 3, proteins from *C. elegans* and mouse are more difficult to predict, as the systems add only +5pp on top of the HT F-score. On the other hand, the systems show higher performance improvement on bacterial sequences, increasing the F-score by 4–23pp, with the best overall improvement seen on *E. coli K-12* proteins. Despite having less annotated sequences, predicting functions of prokaryote proteins seems an easier task for machine learning systems. This is probably due to the fact that prokaryotes are simpler organisms with fewer functions and shallower GO ontologies.

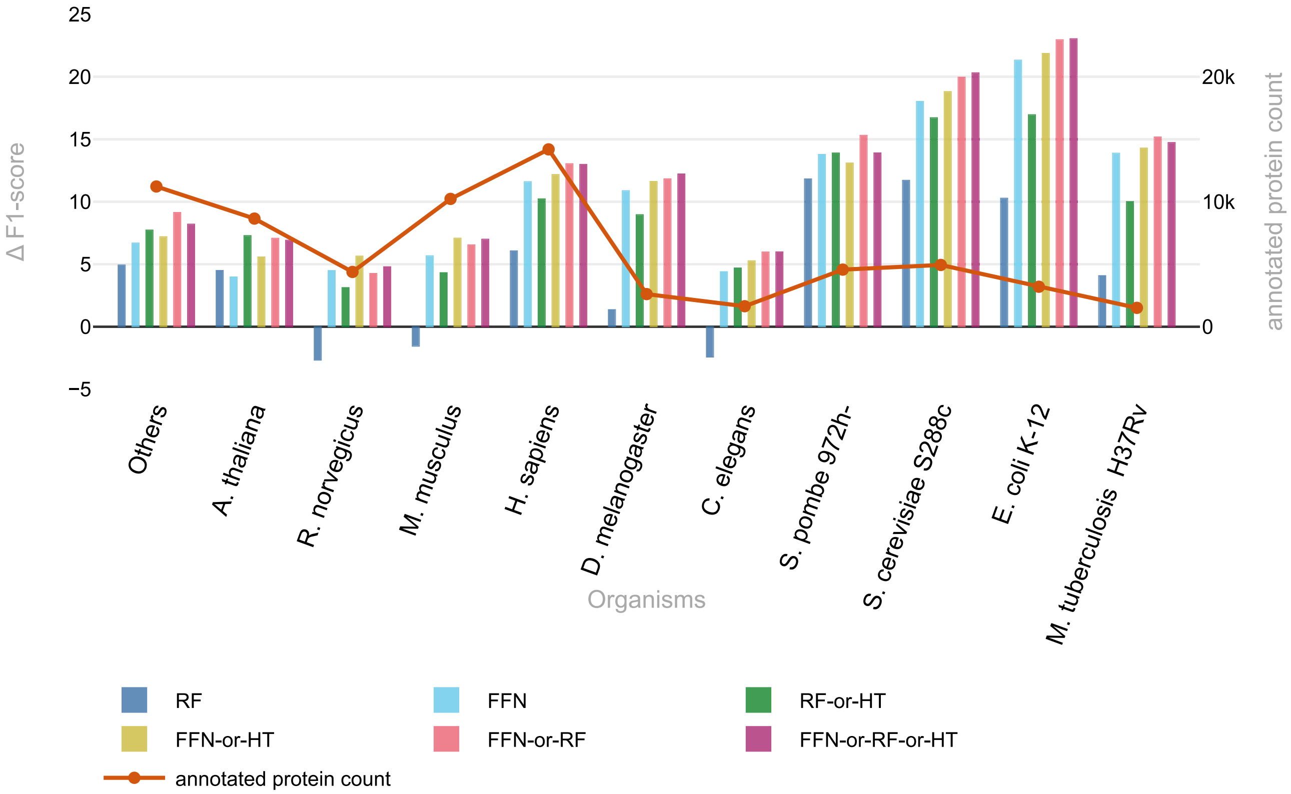

Fig. 3. The performance of the methods on the 10 organisms that have more than 1,000 annotated protein sequences in the training data. *Others* represents a group of organisms with less than 1,000 annotated protein sequences. The vertical bars plot the difference of each tested method to the Homology Transfer fallback (left vertical scale). The red connecting line plots the number of annotated proteins for each organism (right vertical scale).

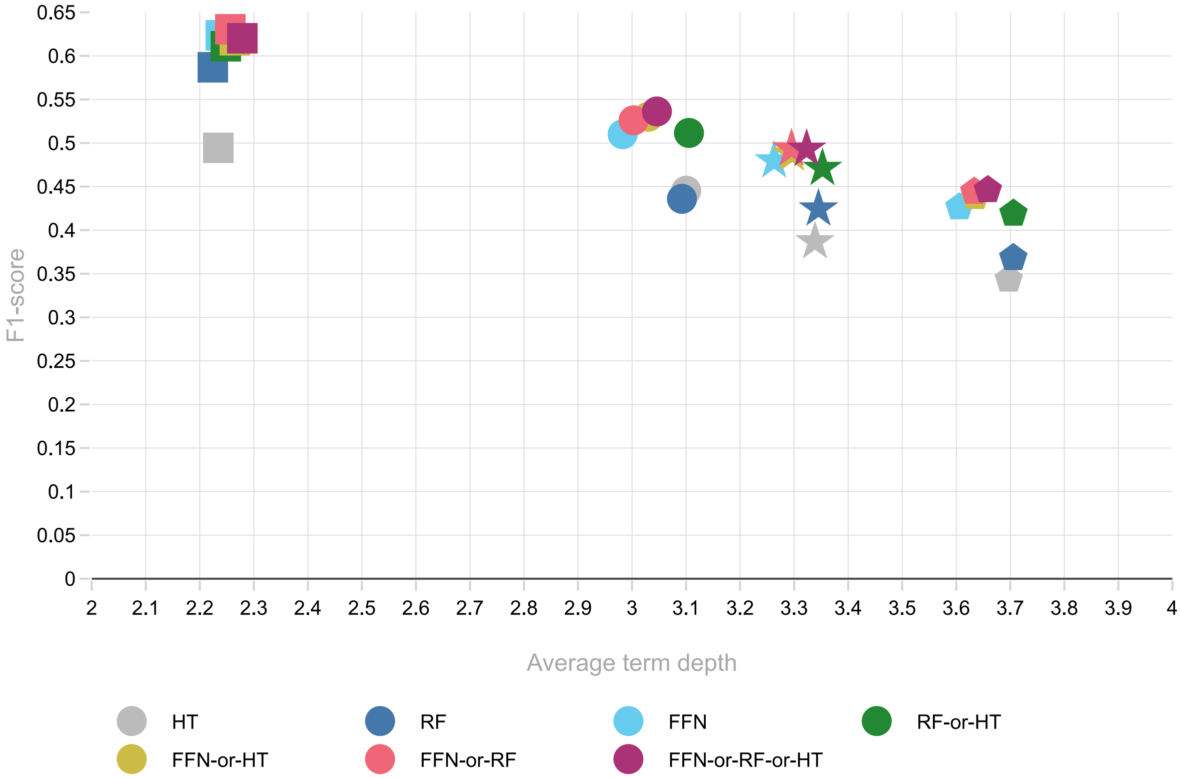

Fig. 4. The performance of the systems for each model on cellular component (square), molecular function (circle), biological process (pentagon) and all ontologies combined (star). The average term depths are calculated from the predictions in each ontology.

Considering the different ontologies, predicting biological processes remains a challenge for the methods, compared to cellular component and molecular function. As shown in Fig. 4, all of the ensembles exhibit the highest performance when predicting cellular components, followed by molecular function and biological process, a trend common to many AFP methods [2], [13], [18], [27]. The difficulty of predicting *biological process* terms is probably due to the fact that its terms are the least correlated with sequence similarity [48], [49]. Thus using only sequence similarity to infer functions can be insufficient or misleading, e.g., paralogs which occur from evolutionary gene duplication processes are often recruited to different pathways [50].

Fig. 4 also shows a near-linear dependence of the F-score on the average term depth in the three GO ontologies, which correlates with the complexity and richness of these ontologies. Overall, the FFN predictions seem marginally less specific as the average term depth of FFN predictions is lower by 0.1 compared to the HT and RF approaches.

In the official evaluation of the CAFA3 challenge our ensemble system has demonstrated good overall performance with a particularly strong performance for molecular functions and cellular component of prokaryotic proteins (top-3) compared to eukaryotic proteins (top-10). As CAFA3 evaluation set emphasizes eukaryotes, the overall ranking of the system is similar to the ranking with the eukaryotic protein subset.

4 Conclusions and Future Work

We have presented methods for automated protein function prediction, evaluated both on Swiss-Prot and also through the CAFA3 challenge. These methods have demonstrated competitive performance among more than 100 CAFA3 entries, especially on the prediction of prokaryotic molecular functions and cellular components. Nevertheless, the absolute performance of the AFP methods show that the task remains a challenge.

We can chart two main directions for further development. First, features derived from sequences related in other ways than solely through homology, e.g., through co-expression or binding, can be potentially beneficial especially for the prediction of biological processes, as demonstrated for instance by Piovesan *et al.* [51] and Kulmanov *et al.* [18]. Of the three GO ontologies, biological process currently exhibits the lowest absolute performance, and therefore is the most impactful target for further development.

Second, structure and sequence-based features without doubt play an important role in determining the function of a protein. However, having to employ the large number of external tools needed to obtain the relevant features is a surprisingly tedious task. As a potential remedy to this practical problem, but also as a research task in its own right, we experimented with using a CNN to derive features directly from the sequence, without any external analysis tools. While the absolute performance of the CNN method can not currently compete with the feature-based methods, the CNN achieved what we believe to be a surprisingly good performance given the simple format of its input. The CNN performing on par with purely BLAST-based predictions suggests that, with further development, the reliance on homology — an important source of features in much of the current AFP work — could potentially be omitted, with a neural model analyzing the amino acid sequences directly.

However, training a well-performing neural model is nontrivial. As future work, we plan to improve our methods by testing other neural network architectures [18], [21] and by pretraining the used protein sequence encoders in a similar fashion as is common in neural computer vision and natural language processing systems [52], [53].

The trained prediction models and source code of the system are publicly available under an open license at https://github.com/TurkuNLP/CAFA3.

Acknowledgments

We would like to thank Sampo Pyysalo for his insightful comments on the manuscript. We also thank CSC - IT Center for Science Ltd, Espoo, Finland for providing the computational resources used in this work. Kai Hakala, Suwisa Kaewphan, and Jari Björne contributed equally to this work.

References

[1] P. Radivojac *et al.*, "A large-scale evaluation of computational protein function prediction," *Nat. Methods*, vol. 10, no. 3, 2013, Art. no. 221.

[2] Y. Jiang *et al.*, "An expanded evaluation of protein function prediction methods shows an improvement in accuracy," *Genome Biol.*, vol. 17, no. 1, 2016, Art. no. 184.

[3] The Gene Ontology Consortium, "The gene ontology resource: 20 years and still GOing strong," *Nucleic Acids Res.*, vol. 47, no. D1, pp. D330–D338, 2018. [Online]. Available: https://dx.doi.org/10.1093/nar/gky1055

[4] M. Ashburner *et al.*, "Gene ontology: Tool for the unification of biology," *Nat. Genet.*, vol. 25, no. 1, 2000, Art. no. 25.

[5] The UniProt Consortium, "UniProt: The universal protein knowledgebase," *Nucleic Acids Res.*, vol. 45, no. D1, pp. D158–D169, 2017. [Online]. Available: http://dx.doi.org/10.1093/nar/gkw1099

[6] D. Cozzetto, D. W. Buchan, K. Bryson, and D. T. Jones, "Protein function prediction by massive integration of evolutionary analyses and multiple data sources," *BMC Bioinf.*, vol. 14, no. 3, Feb. 2013, Art. no. S1. [Online]. Available: https://doi.org/10.1186/1471-2105-14-S3-S1

[7] D. J. Lipman *et al.*, "Gapped BLAST and PSI-BLAST: A new generation of protein database search programs," *Nucleic Acids Res.*, vol. 25, no. 17, pp. 3389–3402, 1997. [Online]. Available: https://dx.doi.org/10.1093/nar/25.17.3389

[8] J. Nokso-Koivisto, L. Holm, P. Koskinen, and P. Törönen, "PANNZER: High-throughput functional annotation of uncharacterized proteins in an error-prone environment," *Bioinformatics*, vol. 31, no. 10, pp. 1544–1552, 2015. [Online]. Available: https://doi.org/10.1093/bioinformatics/btu851

[9] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *J. Mol. Biol.*, vol. 215, no. 3, pp. 403–410, 1990. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022283605803602

[10] M. Falda *et al.*, "Argot2: A large scale function prediction tool relying on semantic similarity of weighted gene ontology terms," *BMC Bioinf.*, vol. 13, no. 4, Mar. 2012, Art. no. S14. [Online]. Available: https://doi.org/10.1186/1471–2105-13-S4-S14

[11] S. R. Eddy, "Profile hidden Markov models," *Bioinformatics*, vol. 14, no. 9, pp. 755–763, 1998. [Online]. Available: https://doi.org/10.1093/bioinformatics/14.9.755

[12] Q. Gong, W. Ning, and W. Tian, "GoFDR: A sequence alignment based method for predicting protein functions," *Methods*, vol. 93, pp. 3–14, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1046202315300487

[13] R. You, Z. Zhang, Y. Xiong, F. Sun, H. Mamitsuka, and S. Zhu, "GOLabeler: Improving sequence-based large-scale protein function prediction by learning to rank," *Bioinformatics*, vol. 34, no. 14, pp. 2465–2473, 2018. [Online]. Available: http://dx.doi.org/10.1093/bioinformatics/bty130

[14] A. L. Mitchell *et al.*, "InterPro in 2019: Improving coverage, classification and access to protein sequence annotations," *Nucleic Acids Res.*, vol. 47, no. D1, pp. D351–D360, 2018. [Online]. Available: https://dx.doi.org/10.1093/nar/gky1100

[15] D. Ofer and M. Linial, "ProFET: Feature engineering captures high-level protein functions," *Bioinformatics*, vol. 31, no. 21, pp. 3429–3436, 2015. [Online]. Available: https://dx.doi.org/10.1093/bioinformatics/btv345

[16] J. Hou, B. Adhikari, and J. Cheng, "DeepSF: Deep convolutional neural network for mapping protein sequences to folds," *Bioinformatics*, vol. 34, no. 8, pp. 1295–1303, 2017.

[17] J. Wang, H. Cao, J. Z. Zhang, and Y. Qi, "Computational protein design with deep learning neural networks," *Sci. Rep.*, vol. 8, no. 1, 2018, Art. no. 6349.

[18] M. Kulmanov, M. A. Khan, and R. Hoehndorf, "DeepGO: Predicting protein functions from sequence and interactions using a deep ontology-aware classifier," *Bioinformatics*, vol. 34, no. 4, pp. 660–668, 2017. [Online]. Available: https://dx.doi.org/10.1093/bioinformatics/btx624

[19] A. Roth *et al.*, "STRING v10: Protein-protein interaction networks, integrated over the tree of life," *Nucleic Acids Res.*, vol. 43, no. D1, pp. D447–D452, 2014. [Online]. Available: https://doi.org/10.1093/nar/gku1003

[20] R. Cao, C. Freitas, L. Chan, M. Sun, H. Jiang, and Z. Chen, "ProLanGO: Protein function prediction using neural machine translation based on a recurrent neural network," *Molecules*, vol. 22, no. 10, 2017, Art. no. 1732.

[21] R. Fa, D. Cozzetto, C. Wan, and D. T. Jones, "Predicting human protein function with multi-task deep neural networks," *PLOS One*, vol. 13, no. 6, pp. 1–16, 2018. [Online]. Available: https://doi.org/10.1371/journal.pone.0198216

[22] S. Teng, H. Luo, and L. Wang, "Random forest-based prediction of protein sumoylation sites from sequence features," in *Proc. 1st ACM Int. Conf. Bioinf. Comput. Biol.*, 2010, pp. 120–126.

[23] T. Jo and J. Cheng, "Improving protein fold recognition by random forest," *BMC Bioinf.*, vol. 15, 2014, Art. no. S14.

[24] P. Jain and J. D. Hirst, "Automatic structure classification of small proteins using random forest," *BMC Bioinf.*, vol. 11, no. 1, Jul. 2010, Art. no. 364. [Online]. Available: https://doi.org/10.1186/1471–2105-11-364

[25] M. Liu and X.-W. Chen, "Prediction of protein-protein interactions using random decision forest framework," *Bioinformatics*, vol. 21, no. 24, pp. 4394–4400, 2005. [Online]. Available: https://dx.doi.org/10.1093/bioinformatics/bti721

[26] C. Kumar, G. Li, and A. Choudhary, "Enzyme function classification using protein sequence features and random forest," in *Proc. 3rd Int. Conf. Bioinf. Biomed. Eng.*, 2009, pp. 1–4.

[27] N. Zhou *et al.*, "The cafa challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens," *Genome Biol.*, vol. 20, no. 1, pp. 1–23, 2019.

[28] C. Camacho, G. Coulouris, V. Avagyan, N. Ma, J. Papadopoulos, K. Bealer, and T. L. Madden, "BLAST+: Architecture and applications," *BMC Bioinf.*, vol. 10, no. 1, 2009, Art. no. 421.

[29] S. Henikoff and J. G. Henikoff, "Amino acid substitution matrices from protein blocks," *Proc. Nat. Acad. Sci. USA*, vol. 89, no. 22, pp. 10 915–10 919, 1992. [Online]. Available: https://www.pnas.org/content/89/22/10915

[30] G. M. Boratyn, A. A. Schäffer, R. Agarwala, S. F. Altschul, D. J. Lipman, and T. L. Madden, "Domain enhanced lookup time accelerated BLAST," *Biol. Direct*, vol. 7, no. 1, Apr. 2012, Art. no. 12. [Online]. Available: https://doi.org/10.1186/1745–6150-7-12

[31] C. Zheng *et al.*, "CDD/SPARCLE: Functional classification of proteins via subfamily domain architectures," *Nucleic Acids Res.*, vol. 45, no. D1, pp. D200–D203, 2016. [Online]. Available: https://dx.doi.org/10.1093/nar/gkw1129

[32] P. Jones *et al.*, "InterProScan 5: Genome-scale protein function classification," *Bioinformatics*, vol. 30, no. 9, pp. 1236–1240, 2014.

[33] S. Federhen, "The NCBI taxonomy database," *Nucleic Acids Res.*, vol. 40, no. D1, pp. D136–D143, 2011.

[34] A. Heddad, M. Brameier, and R. M. MacCallum, "Evolving regular expression-based sequence classifiers for protein nuclear localisation," in *Workshops on Applications of Evolutionary Computation*. Berlin, Germany: Springer, 2004, pp. 31–40.

[35] L. Kiemer, J. D. Bendtsen, and N. Blom, "NetAcet: Prediction of n-terminal acetylation sites," *Bioinformatics*, vol. 21, no. 7, pp. 1269–1270, 2005. [Online]. Available: + http://dx.doi.org/10.1093/bioinformatics/bti130

[36] A. Pierleoni, P. L. Martelli, and R. Casadio, "PredGPI: A GPI-anchor predictor," *BMC Bioinf.*, vol. 9, no. 1, 2008, Art. no. 392.

[37] C.-H. Tung, C.-W. Chen, H.-H. Sun, and Y.-W. Chu, "Predicting human protein subcellular localization by heterogeneous and comprehensive approaches," *PLoS One*, vol. 12, no. 6, pp. 1–14, 2017. [Online]. Available: https://doi.org/10.1371/journal.pone.0178832

[38] S. Kawashima, P. Pokarowski, M. Pokarowska, A. Kolinski, T. Katayama, and M. Kanehisa, "AAindex: Amino acid index database, progress report 2008," *Nucleic Acids Res.*, vol. 36, pp. D202–D205, 2008. [Online]. Available: http://dx.doi.org/10.1093/nar/gkm998

[39] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[40] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[41] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[42] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Advances Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[43] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Proc. Advances Neural Inf. Process. Syst.*, 2015, pp. 649–657.

[44] S. Wang, J. Peng, J. Ma, and J. Xu, "Protein secondary structure prediction using deep convolutional neural fields," *Sci. Rep.*, vol. 6, 2016, Art. no. 18962.

[45] D. Xu and R. Nussinov, "Favorable domain size in proteins," *Folding Des.*, vol. 3, no. 1, pp. 11–17, 1998.

[46] H. Zeng, M. D. Edwards, G. Liu, and D. K. Gifford, "Convolutional neural network architectures for predicting DNA–protein binding," *Bioinformatics*, vol. 32, no. 12, pp. i121–i127, 2016.

[47] A. L. Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Artif. Intell.*, vol. 97, no. 1, pp. 245–271, 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0004370297000635

[48] P. W. Lord, R. D. Stevens, A. Brass, and C. A. Goble, "Semantic similarity measures as tools for exploring the gene ontology," in *Proc. Pacific Symp. Biocomput.*, 2003, pp. 601–612.

[49] P. W. Lord, R. D. Stevens, A. Brass, and C. A. Goble, "Investigating semantic similarity measures across the gene ontology: The relationship between sequence and annotation," *Bioinformatics*, vol. 19, no. 10, pp. 1275–1283, 2003.

[50] J. A. Gerlt and P. C. Babbitt, "Divergent evolution of enzymatic function: Mechanistically diverse superfamilies and functionally distinct suprafamilies," *Annu. Rev. Biochem.*, vol. 70, no. 1, pp. 209–246, 2001. [Online]. Available: https://doi.org/10.1146/annurev.biochem.70.1.209

[51] D. Piovesan, M. Giollo, C. Ferrari, and S. C. E. Tosatto, "Protein function prediction using guilty by association from interaction networks," *Amino Acids*, vol. 47, no. 12, pp. 2583–2592, Dec. 2015. [Online]. Available: https://doi.org/10.1007/s00726–015-2049-3

[52] S. Kornblith, J. Shlens, and Q. V. Le, "Do better ImageNet models transfer better?," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2656–2666, doi: 10.1109/CVPR.2019.00277.

[53] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. 2019 Conf. North Amer. Chapter Assoc. Comput. Linguistics Human Lang. Tech.*, pp. 4171–4186, 2019. [Online]. Available: https://www.aclweb.org/anthology/N19-1423

Kai Hakala received the MSc degree in software engineering from the University of Turku and is currently working toward the PhD degree in computer science at the Department of Future Technologies, University of Turku.

Suwisa Kaewphan received the MSc degree in bioinformatics from the University of Turku and is currently working toward the PhD degree in computer science at the Department of Future Technologies, University of Turku.

Jari Björne received the PhD degree in computer science from the University of Turku. He is a postdoctoral researcher with the Department of Future Technologies, University of Turku.

Farrokh Mehryary received the MSc degree in bioinformatics from the University of Turku and is currently working toward the PhD degree in computer science at the Department of Future Technologies, University of Turku.

Hans Moen received the PhD degree in computer science from the Norwegian University of Science and Technology. He is a postdoctoral researcher with the Department of Future Technologies, University of Turku.

Martti Tolvanen received the PhD degree in bioinformatics from the University of Tampere. He is a University lecturer with the Department of Future Technologies, University of Turku.

Tapio Salakoski received the PhD degree in computer science from the University of Turku. He is the dean of the Faculty of Science and Engineering and a professor of Language and Speech Technology with the Department of Future Technologies, University of Turku. His area of expertise includes artificial intelligence especially in language technology and machine learning, bio and health informatics, learning analytics and technology, and digital humanities.

Filip Ginter received the PhD degree in computer science from the University of Turku. He is an associate professor of language and Speech Technology with the Department of Future Technologies, University of Turku.