Proteins: Structure, Function, and Bioinformatics

WILEY

PROTEINS

RESEARCH ARTICLE

SUPERMAGO: Protein Function Prediction Based on Transformer Embeddings

Gabriel Bianchin de Oliveira | Helio Pedrini | Zanoni Dias

Institute of Computing, University of Campinas, Campinas, Brazil

Correspondence: Gabriel Bianchin de Oliveira (gabriel.oliveira@ic.unicamp.br)

Received: 20 September 2024 | **Revised:** 28 November 2024 | **Accepted:** 9 December 2024

Funding: This work was supported by the São Paulo Research Foundation (grant number 2017/12646-3), the National Council for Scientific and Technological Development (grant numbers 302530/2022-3, 304836/2022-2, 161015/2021-2), the Coordination for the Improvement of Higher Education Personnel, and Santander Bank—Brazil.

Keywords: local alignment | machine learning | neural network | protein function prediction | Transformer

ABSTRACT

Recent technological advancements have enabled the experimental determination of amino acid sequences for numerous proteins. However, analyzing protein functions, which is essential for understanding their roles within cells, remains a challenging task due to the associated costs and time constraints. To address this challenge, various computational approaches have been proposed to aid in the categorization of protein functions, mainly utilizing amino acid sequences. In this study, we introduce SUPERMAGO, a method that leverages amino acid sequences to predict protein functions. Our approach employs Transformer architectures, pre-trained on protein data, to extract features from the sequences. We use multilayer perceptrons for classification and a stacking neural network to aggregate the predictions, which significantly enhances the performance of our method. We also present SUPERMAGO+, an ensemble of SUPERMAGO and DIAMOND, based on neural networks that assign different weights to each term, offering a novel weighting mechanism compared with existing methods in the literature. Additionally, we introduce SUPERMAGO+Web, a web server-compatible version of SUPERMAGO+ designed to operate with reduced computational resources. Both SUPERMAGO and SUPERMAGO+ consistently outperformed state-of-the-art approaches in our evaluations, establishing them as leading methods for this task when considering only amino acid sequence information.

1 | Introduction

In recent decades, advancements in next-generation sequencing technologies have significantly increased the number of proteins sequenced in laboratory experiments [1]. Nevertheless, the quantity of proteins with established characteristics, such as functions, remains much lower. When it comes to protein function, determining it through a single laboratory experiment can be challenging, due to factors like biological complexities, budget constraints, and ethical considerations [2]. To address this, initiatives have been launched that leverage computational resources to reduce the gap between sequenced proteins and those with identified functions.

Protein functions are typically classified according to biological ontologies [3, 4]. The most utilized ontology for protein function prediction is Gene Ontology (GO) [5], a framework that describes functions from the molecular level up to the organism level. The GO is divided into: Biological Process Ontology (BPO), which categorizes larger processes, such as signal transduction, in which the protein is involved in; Cellular Component Ontology (CCO), which indicates where the protein performs its function, such as in the mitochondrion; and Molecular Function Ontology (MFO), which describes molecular-level activities, such as transporter activity. Each ontology is organized as a directed acyclic graph, with relationships between more general terms (super-classes) and

more specific ones (sub-classes). If a protein has a sub-class term, it also has all the ancestor terms (super-classes) up to the root term of the ontology. Moreover, a single protein can have multiple terms within one ontology. Therefore, this task is approached as a multi-label classification problem in computational models.

Various methods have been proposed in the literature to handle this task. Considering the features used by them, some utilize amino acid sequences [6–14], while others use protein–protein interaction networks [15, 16], biological information, such as families [17] and structures [18], or a combination of different features [19–23]. Due to the nature of the available data, methods that use only the amino acid sequences have an advantage over other methods, given the larger number of proteins with this information available compared to other types of features.

Looking at the methods based on amino acid sequences, we can divide them into three categories: methods based on local alignment tools, those based on machine learning, and ensembles of machine learning methods with local alignment tools. Local alignment-based methods [6–8] operate on the premise that proteins with similar sequences also have similar functions, and to predict, these approaches use tools like BLASTp [24] and DIAMOND [25]. However, methods purely based on local similarity may struggle with remote homology, where proteins with low sequence similarity perform the same function [26].

Methods based on machine learning use information derived from amino acid sequences, partially addressing the problem of remote homology. For classification, approaches have proposed the application of convolutional neural networks [9, 19, 27], recurrent networks [27], and Transformer architectures [10–14] for this task. Transformer-based architectures have made significant advancements in predicting protein functions, as these models, pre-trained with millions or billions of amino acid sequences, can effectively analyze the sequences and extract better representations than approaches with other neural network architectures, establishing themselves as the state-of-the-art for this task.

Finally, methods based on ensemble of machine learning with local alignment tools can combine the functionalities of each approach and improve the results when compared to using them individually [9–14].

Despite achieving good results in protein function annotation tasks, machine-learning methods have largely underutilized the aggregation of diverse base classifiers, which can enhance robustness and provide insights into different aspects [28, 29]. In the context of ensembles combining machine learning and local alignment tools, the methods in the literature usually employ a simple linear combination of predictions, without considering term-specific weights, which can negatively impact the performance of these approaches.

In this paper, we introduce SUPERMAGO, a novel method that utilizes embeddings extracted from various layers of Transformer-based architectures and employs layer-specific classifiers, along with a multilayer perceptron to aggregate the predictions. We also present SUPERMAGO+, an ensemble method that combines SUPERMAGO with DIAMOND, where a neural network learns the weights for each term from each method, differentiating it from approaches in the literature. Finally, we introduce SUPERMAGO+Web, a web server-compatible method designed to run SUPERMAGO+ with reduced computational resources.

In the experiments, we evaluated SUPERMAGO and SUPERMAGO+, which outperformed the methods in the literature in the evaluated metrics and presented competitive results in the analyses involving the level of ontology, domain, similarity, term frequency, and filtered test sets. Our model also showed statistical significant difference compared to the other approaches in the literature, establishing it as the state-of-the-art for protein function prediction task.

The rest of the paper is organized as follows. In Section 2, we describe SUPERMAGO, SUPERMAGO+, and SUPERMAGO+Web, the comparison methods, the dataset, and the evaluation metrics. In Section 3, we present and discuss the results of SUPERMAGO and SUPERMAGO+ in comparison with the literature approaches, as well as the comparison between SUPERMAGO+ and SUPERMAGO+Web. In Section 4, we present our conclusions and suggest possible directions for future work.

2 | Materials and Methods

In this section, we describe SUPERMAGO, SUPERMAGO+, SUPERMAGO+Web, the comparison methods, the dataset, and the evaluation metrics.

2.1 | SUPERMAGO

In this section, we describe our approach based on machine learning, named SUPERb Magical classifier for protein function Annotation with Gene Ontology (SUPERMAGO). This method comprises three steps: embedding extraction, layer classification, and stacking. The SUPERMAGO pipeline is depicted in Figure 1.

2.1.1 | Embedding Extraction

The initial phase of SUPERMAGO involves embedding extraction. For this purpose, we utilized the ESM2 T36 [30] and ProtT5 [31] encoder architectures, extracting representations from the last five layers of each model. Specifically, we employed layers 36, 35, 34, 33, and 32 from ESM2 T36, and layers 24, 23, 22, 21, and 20 from ProtT5. A comparative analysis of the impact of using fewer versus more layers in the SUPERMAGO pipeline is presented in Section 3.

Due to the input length limitation of ESM2 T36, we split amino acid sequences exceeding 1022 residues into slices using a sliding window technique for both architectures. For example, a protein sequence of 2500 amino acids was divided into three slices: the first two comprising 1022 amino acids each, and the last containing 456 amino acids.

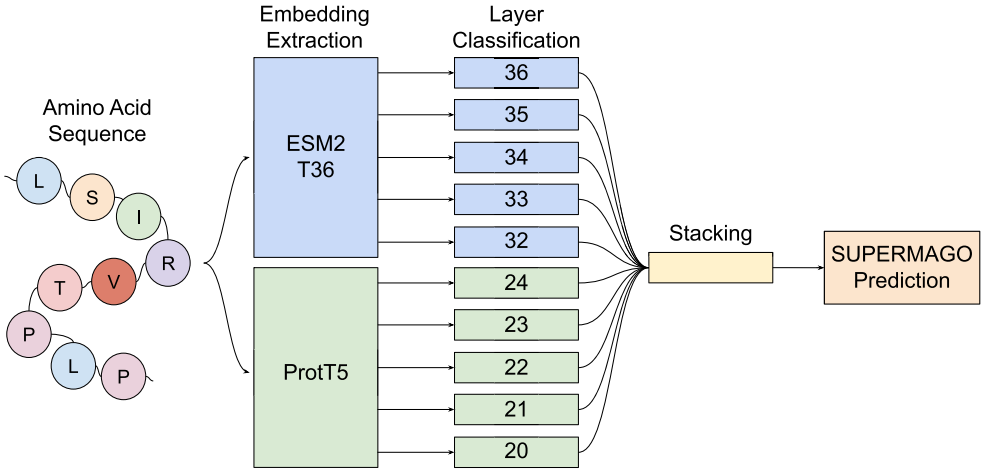

FIGURE 1 | SUPERMAGO pipeline. The process begins with the amino acid sequences being processed by ESM2 T36 and ProtT5 to extract embeddings from the last five layers of each model. These embeddings are subsequently used for training and classification in a per-layer multilayer perceptron model. Finally, the predictions from each classifier are aggregated using a stacking model to generate the final prediction.

Based on the embedding extraction, sequences longer than 1022 amino acids were processed using the sliding window technique, resulting in multiple slices per protein. With that, we aggregated this information using mean operation across all slices corresponding to the same protein. At the end of this process, each protein was represented by 2560 floating-point values per layer for ESM2 T36, and 1024 floating-point values per layer for ProtT5.

2.1.2 | Layer Classification

Using the extracted embeddings, we trained a multilayer perceptron for each configuration, considering both layer-based embeddings of ESM2 T36 or ProtT5 and ontology. We evaluated various configurations of multilayer perceptron models, varying the number of layers (from 1 to 3) and neurons per layer, assessing values from the set {512,1000,1024,1536,2048,4096}. The best model was selected based on the F_{max} score on the validation set, considering the results across the three ontologies.

The optimal configuration was a network with two layers, with each layer comprising 4096 neurons and using ReLU activation. The output layer, corresponding to the number of terms per ontology (500 for BPO, 498 for CCO, and 499 for MFO), utilized sigmoid activation. As a result, each ontology had ten classifiers, five based on ESM2 T36 embeddings and five based on ProtT5 embeddings.

For training each model, we used 50 epochs, with the binary cross-entropy loss function, employing early stopping configured to interrupt the training process if no improvement was observed after 5 epochs, and Adam [32] optimizer with a learning rate of 0.0001.

2.1.3 | Stacking

After obtaining predictions from each classifier based on a specific layer of ESM2 T36 or ProtT5, we trained a stacking-based classifier to aggregate the predictions. Instead of using a fully connected network, we designed a neural network with only a single output layer, where each neuron received the predictions of a specific term across all base models, as illustrated in Figure 2. In the stacking model, the number of neurons is equal to the number of terms per ontology (500 for BPO, 498 for CCO, and 499 for MFO).

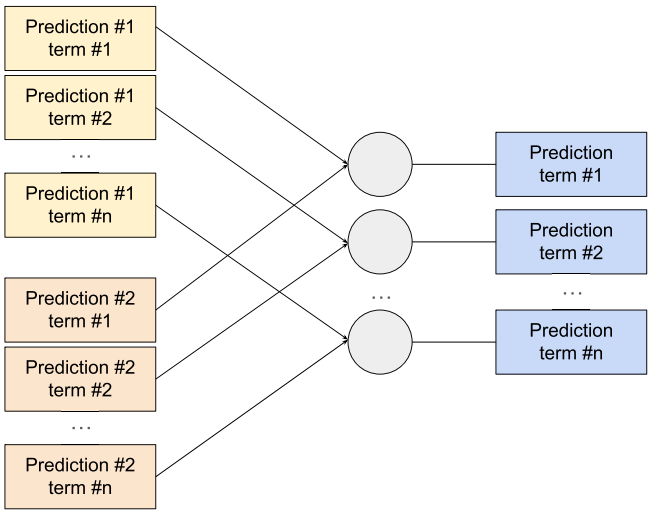

FIGURE 2 | The stacking classifier used in SUPERMAGO connects the predictions for the same term across different layer-based classifiers. The sum of the weights connected to each specific output neuron (shown in gray) is equal to 1.

Regarding the connections, we applied softmax operation to the weights linking the predictions to the output neurons, ensuring that their sum was equal to 1. This operation allows the network to perform a linear combination of the predicted values across layers, with specific weights for each set of terms and classifiers. For the output neurons, we employed linear activation due to the linear combination context applied to the weights connected to them.

To train the stacking model, we used 80% of the predictions on the validation set for training, with the remaining 20% used for

validation. We adopted this approach because the predictions on the training set tend to be more accurate than those on the validation and test sets, given that each layer-specific classifier is trained and optimized using this set. The stacking model was trained for 50 epochs, with the binary cross-entropy loss function, employing early stopping after 5 epochs with no improvement, and using Adam optimizer with a learning rate of 0.001.

2.2 | SUPERMAGO+

In this section, we present SUPERMAGO+, our method based on ensemble of machine learning (SUPERMAGO) with local alignment tool (DIAMOND [25]).

2.2.1 | DIAMOND

During the utilization of DIAMOND, we configured it with an *E*-value of 0.001, which is the commonly used threshold in the literature [9, 10, 14], to compare the proteins in the evaluation set with those in the training set. We compare the use of BLASTp instead of DIAMOND in Section 3.

To generate the predictions of DIAMOND, we utilized Equation (1), where $S(p,f)$ represents the score S of a protein in the evaluation test p for a specific term t. Here, E denotes the set of proteins in the training set aligned with p, s denotes a specific protein in the E set, T_s represents the ground-truth of protein s, $\text{bitscore}(p,s)$ signifies the bitscore result of the alignment between p and s, and I is the identity function, returning 1 if true or 0 if false.

$$S(p,t) = \frac{\sum_{s\in E} I\left(t \in T_s\right) \times \text{bitscore}(p,s)}{\sum_{s\in E} \text{bitscore}(p,s)} \tag{1}$$

2.2.2 | Ensemble

To create SUPERMAGO+, we performed the ensemble of SUPERMAGO with DIAMOND. For this, we employed the same stacking classifier approach described for SUPERMAGO with minor modification to account for the possibility that DIAMOND might not generate predictions for some proteins due to the absence of homologous proteins in the training database. With that, in addition to the predictions from SUPERMAGO and DIAMOND, the adapted stacking model also receives a binary matrix indicating whether SUPERMAGO and DIAMOND made predictions for the analyzed protein.

In the case of SUPERMAGO, it always generates predictions, therefore, if only SUPERMAGO provides predictions, we apply a mask to the weight connection corresponding to each term in the DIAMOND predictions. This ensures that prediction of SUPERMAGO has a weight of 1, while the prediction of DIAMOND has a weight of 0, meaning that only SUPERMAGO's prediction is used. If both methods have predicted the protein functions, we applied the weighted combination of the predictions, that is, without mask for the DIAMOND predictions.

Similar to the stacking process in SUPERMAGO, we used 80% of the validation predictions to train this model and 20% for validation. The training was conducted over 50 epochs, with the binary cross-entropy loss function, with early stopping applied after 5 epochs, and using Adam optimizer with a learning rate of 0.001.

2.3 | SUPERMAGO+Web

For our web version, we replaced ESM2 T36 with ESM2 T12 and quantized the models to `int4` [33] format, utilizing LoRA [34] and `bfloat16` computation type. This process significantly improved the model's efficiency in terms of memory usage, enabling it to operate with limited computational resources. We compare the results of SUPERMAGO+ and SUPERMAGO+Web in Section 3.

2.4 | Comparison Methods

In this subsection, we briefly describe the state-of-the-art approaches that we compared to our method. During the experiments, we executed each method using the configurations outlined in their respective papers and online repositories. We only compared SUPERMAGO and SUPERMAGO+ with approaches that have available code and use amino acid sequences as input features for their models.

2.4.1 | DeepGO

This method [19] utilizes trigrams to characterize amino acid sequences. It transforms the inputs, passing the features through convolutional layers before proceeding to neurons organized according to the ontology structure. DeepGO also has the capability to incorporate protein–protein interaction network features, but we disregard this input to ensure a fair comparison with other methods.

2.4.2 | DeepGOCNN and DeepGOPlus

As an advancement from DeepGO, DeepGOCNN [9] adopts a one-hot encoding approach for input information. It employs convolutional layers, but eliminates the organization of the neurons based on the ontology from the previous version. The authors also introduced DeepGOPlus, which is an ensemble of DeepGOCNN with DIAMOND [25], utilizing the formula specified in Equation (2) when DIAMOND provides predictions, otherwise relying solely on DeepGOCNN predictions. In the equation, $S(p,f)$ represents the score S of a protein in the evaluation test p for a specific term t for DeepGOCNN (DG), DIAMOND (D), and DeepGOPlus (DGP).

$$S_{DGP}(p,t) = \alpha \times S_{DG}(p,t) + (1-\alpha) \times S_D(p,t) \tag{2}$$

2.4.3 | TALE and TALE+

This approach [10] was among the pioneering methods to utilize the Transformer for protein function prediction tasks. It employs the encoder of the vanilla Transformer [35] with GO organization. For each ontology, the authors applied an ensemble

of predictions with various configurations of the Transformer architecture. Additionally, they introduced TALE+, which combines TALE with DIAMOND [25], employing the formula described in Equation (2) when DIAMOND predicts for a specific protein, and relying on TALE predictions otherwise.

2.4.4 | PFmulDL

In this approach [27], the authors presented a method based on one-hot encoding, convolutional layers, and recurrent layers. This method includes a pre-training step for the convolutional part of the architecture. Consequently, the features are processed through this segment before being passed to recurrent layers.

2.4.5 | PU-GO and PU-GO+Diamond

PU-GO [11] utilizes ESM2 T48 [30] embeddings to train a classifier based on fully-connected layers for predicting protein functions. To optimize the classifier, the authors applied ranking using positive labels and unlabeled samples. Additionally, they introduced PU-GO+Diamond, an ensemble of PU-GO with DIAMOND [25]. Unlike other approaches, the authors simply averaged the predictions of PU-GO and DIAMOND.

2.4.6 | ATGO and ATGO+

ATGO [13] employs the extraction of embeddings from the last three layers of the ESM-1b [36] architecture. Subsequently, it calculated the average embedding considering the representations of the three last layers and passed the final feature of each protein to train a classifier comprising fully-connected neurons. Furthermore, they introduced ATGO+, an ensemble of ATGO with BLASTp [24], which utilizes Equation (2) when BLASTp provides predictions, otherwise, it relies on ATGO predictions.

2.4.7 | TEMPROT and TEMPROT+

In this study [12], the authors presented an approach based on Transformer for predicting protein functions. The method utilizes data augmentation by duplicating the original training set, preprocesses the sequences using the sliding window technique, fine-tunes the ProtBERT-BFD [31] architecture, extracts the embeddings from the last layer of the fine-tuned architecture, and trains a meta-classifier with fully-connected neurons. Additionally, they introduced an ensemble of TEMPROT with BLASTp [24], named TEMPROT+, which applies the formula of Equation (2) when BLASTp provides predictions, otherwise, it relies on TEMPROT predictions.

2.4.8 | MAGO and MAGO+

MAGO [14] is an approach based on embeddings of the last layer of the ESM2 T36 architecture with AutoML. It utilizes bitscore filtering for BLASTp [24] to ensemble with MAGO predictions, selecting alignments with at least 250 of bitscore, resulting in the MAGO+ approach. Similarly to other approaches, it employs Equation (2) within bitscore selection when BLASTp provides predictions, and relying on MAGO predictions otherwise.

2.4.9 | PROTGOAT

This approach [20] achieved fourth place in the CAFA5 competition. It utilizes embeddings from the last layer of five different Transformers architectures (ESM2 T30 [30], ESM2 T36 [30], ProtBERT [31], ProtT5 [31], and Ankh [37]), along with information about taxonomy and text data from the papers describing the proteins. All features are passed to a classifier based on a fully-connected architecture to predict the functions. For a fair comparison, only the information from amino acid sequences, that is, the embeddings of Transformers architectures, was utilized in our tests.

2.4.10 | DIAMOND and BLASTp

During the evaluation, we also compared DIAMOND [25] and BLASTp [24] predictions. Both methods utilize the formula of Equation (1).

2.5 | Dataset

In this study, we utilized the dataset from Oliveira, Pedrini, and Dias [14] to compare SUPERMAGO and SUPERMAGO+ with state-of-the-art approaches and to develop SUPERMAGO+Web.

To construct this dataset, the authors utilized proteins from the latest CAFA challenge (https://www.kaggle.com/competitions/cafa-5-protein-function-prediction), referred to as CAFA5, along with data from the 2022_05 release of Swiss-Prot and the GO structure file from January 1, 2023. Based on the available information from CAFA5, the dataset was divided into training, validation, and test sets, comprising 80%, 10%, and 10% of the data, respectively. For the number of terms per ontology, the authors selected the most frequently occurring ones to generate the prediction labels. Table 1 displays the number of proteins and terms in each ontology.

2.6 | Evaluation Metrics

We used six evaluation metrics (F_{max}, AuPRC, IAuPRC, F^*_{max}, wF_{max}, and S_{min}) to compare our approach with state-of-the-art

TABLE 1 | Number of proteins and terms in each ontology.

	BPO	CCO	MFO
Train	73 768	74 328	62 909
Validation	9221	9292	7864
Test	9221	9292	7864
Terms	500	498	499

methods. The first metric, F_{max}, is the main measure employed in the CAFA challenge [38]. It calculates the maximum harmonic mean of precision (pr) and recall (rc) across various thresholds τ, ranging from 0.01 to 1.00. Equation (3) illustrates the calculation of F_{max}.

$$F_{max} = \max_{\tau}\left\{\frac{2 \times \mathrm{pr}(\tau) \times \mathrm{rc}(\tau)}{\mathrm{pr}(\tau) + \mathrm{rc}(\tau)}\right\} \tag{3}$$

The metric $\mathrm{pr}(\tau)$ evaluates the average ratio of correct predictions to total predictions with at least τ confidence for each protein in the evaluation set. Equation (4) outlines the formula for $\mathrm{pr}(\tau)$, where $P_i(\tau)$ represents the terms predicted with at least τ score for a protein i, T_i is the ground-truth of protein i, and $m(\tau)$ is the number of proteins with at least one term predicted with confidence greater than or equal to τ.

$$\mathrm{pr}(\tau) = \frac{1}{m(\tau)} \sum_{i=1}^{m(\tau)} \frac{|P_i(\tau) \cap T_i|}{|P_i(\tau)|} \tag{4}$$

For $\mathrm{rc}(\tau)$, the evaluation considers the average ratio of correct predictions to the ground-truth, as expressed in Equation (5). Here, n denotes the number of proteins in the evaluation set.

$$\mathrm{rc}(\tau) = \frac{1}{n} \sum_{i=1}^{n} \frac{|P_i(\tau) \cap T_i|}{|T_i|} \tag{5}$$

In our study, we also used the Area under the Precision-Recall Curve (AuPRC), which is one of the most commonly employed metrics in the literature for this task. By considering precision and recall values across various thresholds, which are also applied in the calculation of F_{max}, we can determine the area under the curve on the precision-recall axis.

Oliveira, Pedrini, and Dias [12] highlighted issues with AuPRC and introduced the Interpolated Area under the Precision-Recall Curve (IAuPRC). Based on that, we also applied this metric to assess and compare our approach with the literature.

Since the F_{max} metric considers all terms of the evaluated ontology, including the root term, we also calculate F^*_{max}, which is the F_{max} value excluding the root term from the calculation.

The metric wF_{max} is the maximum harmonic mean of weighted precision (wpr) and weighted recall (wrc) across various thresholds τ, ranging from 0.01 to 1.00, as shown in Equation (6).

$$wF_{max} = \max_{\tau}\left\{\frac{2 \times \mathrm{wpr}(\tau) \times \mathrm{wrc}(\tau)}{\mathrm{wpr}(\tau) + \mathrm{wrc}(\tau)}\right\} \tag{6}$$

A crucial step in the computation of wF_{max} involves determining the information content (IC) for each term within the ontology. Equation (7) provides the formula for calculating IC for a specific term t, considering the probability Pb of term t being present, based on its ancestors $\mathrm{Pr}(t)$.

$$\mathrm{IC}(t) = -\log(\mathrm{Pb}(t) | \mathrm{Pr}(t)) \tag{7}$$

With the IC of each term t, it is possible to calculate wpr and wrc for each τ, as described in Equations (8) and (9).

$$\mathrm{wpr}(\tau) = \frac{1}{m(\tau)} \sum_{i=1}^{m(\tau)} \frac{\sum_{t \in P_i(\tau) \cap T_i} \mathrm{IC}(t)}{\sum_{t \in P_i(\tau)} \mathrm{IC}(t)} \tag{8}$$

$$\mathrm{wrc}(\tau) = \frac{1}{n} \sum_{i=1}^{n} \frac{\sum_{t \in P_i(\tau) \cap T_i} \mathrm{IC}(t)}{\sum_{t \in T_i} \mathrm{IC}(t)} \tag{9}$$

The final metric utilized in our investigation is denoted as S_{min} [39]. This particular metric quantifies the minimum semantic distance by taking into account the remaining uncertainty (ru) and misinformation (mi) across various thresholds τ, ranging from 0.01 up to 1.00. Equation (10) illustrates the calculation of S_{min}.

$$S_{min} = \min_{\tau} \sqrt{\mathrm{ru}(\tau)^2 + \mathrm{mi}(\tau)^2} \tag{10}$$

The calculation of ru entails the analysis of false negatives for a specific threshold τ, that is, the correct terms of a protein i that the model failed to predict, as displayed in Equation (11).

$$\mathrm{ru}(\tau) = \frac{1}{n} \sum_{i=1}^{n} \sum_{t \in T_i - P_i(\tau)} \mathrm{IC}(t) \tag{11}$$

The computation of mi assesses the false positive for a specific threshold τ, that is, the predicted terms for a protein that are not present in the ground-truth of a protein i, as demonstrated in Equation (12).

$$\mathrm{mi}(\tau) = \frac{1}{n} \sum_{i=1}^{n} \sum_{t \in P_i(\tau) - T_i} \mathrm{IC}(t) \tag{12}$$

In addition to the evaluations using the mentioned metrics, we also employed statistical analysis to identify methods that have significant differences between them. To do so, we applied the nonparametric Ivan–Davenport [40] and Nemenyi [41] tests. Both tests use the concept of average ranking R_j for each approach j, which is the average position of this method based on some evaluation measure. To obtain the average ranking, we used the F_{max} value for each protein in the evaluation set.

During our analyses, we began the evaluation with the Ivan–Davenport test, which indicates if there is a statistical difference between the methods. For this calculation, we require the Friedman chi-square value χ^2_F, as described in Equation (13), where N is the number of samples and k is the number of evaluated methods.

$$\chi^2_F = \frac{12N}{k(k+1)} \left[\sum_j R_j^2 - \frac{k(k+1)^2}{4} \right] \tag{13}$$

Taking into account the value of χ^2_F, it is possible to obtain the F_F statistic [42], as presented in Equation (14). If F_F is greater than the critical value, considering the F-distribution with $k-1$ and $(k-1)(N-1)$ degrees of freedom and a certain confidence

interval, the null hypothesis is rejected and we conclude that there is a statistical difference between the methods.

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2} \tag{14}$$

If the Ivan–Davenport test indicates a statistical difference between the approaches, we proceed with the Nemenyi test. This test calculates the critical distance (CD) between the average rankings to determine which methods are statistically equivalent. The CD calculation, described in Equation (15), employs q_α, a value derived from the studentized range divided by $\sqrt{2}$. Here, α represents the confidence level.

$$\mathrm{CD} = q_\alpha \sqrt{\frac{k(k+1)}{6N}} \tag{15}$$

During the evaluation, we utilized the true path rule [43]. For each approach under evaluation, we propagated predictions from the higher levels of the ontology to the root node. This ensured that a super-class node always had a score equal to or higher than its sub-classes.

3 | Results and Discussion

In this section, we conduct experiments to compare SUPERMAGO and SUPERMAGO+ to the literature and to compare SUPERMAGO+ and SUPERMAGO+Web. We also describe the development of the web server with SUPERMAGO+Web.

3.1 | Performance on Test Set

In our initial assessment, we compared SUPERMAGO with other machine learning methods (PFmulDL, DeepGO, DeepGOCNN, TALE, ATGO, TEMPROT, PU-GO, PROTGOAT, and MAGO) for BPO, as shown in the upper section of Table 2. In this context, SUPERMAGO surpassed the state-of-the-art approaches across all evaluations. Considering SUPERMAGO+, in the lower section of Table 2, compared to DIAMOND, BLASTp, DeepGOPlus, TALE+, ATGO+, TEMPROT+, PU-GO+Diamond, and MAGO+, it achieved the highest scores for five of six metrics, being surpassed by PU-GO+Diamond by just 0.001 in $F_{\max}$. However, in the comparison excluding the root term ($F^*_{\max}$) and balanced by IC ($wF_{\max}$), SUPERMAGO+ demonstrated superior performance compared to PU-GO+Diamond.

Then, we assessed SUPERMAGO and SUPERMAGO+ using CCO, as shown in Table 3. In both cases, our methods achieved the best scores for $F_{\max}$, $F^*_{\max}$, $wF_{\max}$, $S_{\min}$, and IAuPRC, though they were slightly surpassed by MAGO and MAGO+ in AuPRC by just 0.003 and 0.002, respectively.

Next, we evaluated our methods for MFO. The results are presented in Table 4, which indicate that SUPERMAGO and

TABLE 2 | Evaluation of the approaches on the BPO test set.

Method	$F_{\max}$	$F^*_{\max}$	$wF_{\max}$	$S_{\min}$	AuPRC	IAuPRC
PFmulDL	0.460	0.416	0.367	22.625	0.489	0.483
DeepGO	0.372	0.328	0.280	25.304	0.293	0.365
DeepGOCNN	0.497	0.448	0.413	21.044	0.521	0.515
TALE	0.427	0.387	0.343	24.023	0.381	0.436
ATGO	0.505	0.467	0.427	22.003	0.497	0.529
TEMPROT	0.514	0.472	0.427	21.355	0.483	0.537
PU-GO	0.463	0.426	0.374	23.490	0.413	0.371
PROTGOAT	0.572	0.538	0.503	19.362	0.567	0.622
MAGO	0.555	0.520	0.483	20.045	0.574	0.593
SUPERMAGO	**0.582**	**0.550**	**0.516**	**19.103**	**0.609**	**0.630**
DIAMOND	0.588	0.566	0.541	18.524	0.455	0.571
BLASTp	0.576	0.550	0.521	19.470	0.495	0.590
DeepGOPlus	0.601	0.568	0.542	18.121	0.636	0.631
TALE+	0.580	0.549	0.524	18.792	0.562	0.608
ATGO+	0.569	0.538	0.509	19.669	0.572	0.605
TEMPROT+	0.588	0.558	0.528	18.987	0.578	0.624
PU-GO+Diamond	**0.620**	0.566	0.542	18.488	0.595	0.622
MAGO+	0.615	0.585	0.559	18.092	0.635	0.660
SUPERMAGO+	0.619	**0.591**	**0.566**	**17.702**	**0.645**	**0.670**

Note: The best results are highlighted in bold and the second-best results are underlined.

TABLE 3 | Evaluation of the approaches on the CCO test set.

Method	F_{max}	F^*_{max}	wF_{max}	S_{min}	AuPRC	IAuPRC
PFmulDL	0.707	0.672	0.529	11.320	0.770	0.766
DeepGO	0.655	0.612	0.416	12.923	0.580	0.690
DeepGOCNN	0.678	0.641	0.475	11.816	0.731	0.726
TALE	0.693	0.656	0.506	11.647	0.657	0.753
ATGO	0.724	0.691	0.562	10.829	0.696	0.790
TEMPROT	0.731	0.699	0.569	10.622	0.731	0.790
PU-GO	0.692	0.655	0.510	11.924	0.637	0.710
PROTGOAT	0.763	0.736	0.630	9.463	0.775	0.841
MAGO	0.755	0.727	0.615	9.661	**0.807**	0.829
SUPERMAGO	**0.769**	**0.743**	**0.641**	**9.211**	0.804	**0.848**
DIAMOND	0.699	0.677	0.586	10.031	0.441	0.673
BLASTp	0.712	0.686	0.582	10.349	0.528	0.729
DeepGOPlus	0.738	0.709	0.594	10.006	0.802	0.799
TALE+	0.753	0.726	0.620	9.644	0.737	0.822
ATGO+	0.743	0.714	0.603	10.082	0.722	0.819
TEMPROT+	0.752	0.724	0.614	9.868	0.752	0.822
PU-GO+Diamond	0.751	0.723	0.622	9.890	0.729	0.801
MAGO+	0.774	0.748	0.649	9.157	**0.815**	0.844
SUPERMAGO+	**0.777**	**0.753**	**0.658**	**8.892**	0.813	**0.857**

Note: The best results are highlighted in bold and the second-best results are underlined.

SUPERMAGO+ outperformed the state-of-the-art approaches across all evaluation metrics in both scenarios.

Based on these results, we conclude that SUPERMAGO consistently achieved competitive results compared to state-of-the-art approaches, with the best scores in 17 of 18 evaluations. Similarly, SUPERMAGO+ demonstrated excellence with top values in 16 out of 18 evaluations. When comparing SUPERMAGO and SUPERMAGO+, the latter configuration showed improvements of 0.037, 0.008, and 0.007 for F_{max} in BPO, CCO, and MFO, respectively. We particularly highlight the improvement in BPO, which is the most challenging ontology to predict, as evidenced by the performance differences between this ontology and CCO and MFO.

3.2 | Statistical Analysis

In the second part of our analysis, we evaluated the statistical differences between the approaches by employing the Iman–Davenport and Nemenyi tests. To do so, we calculated the F_{max} score for each protein in the test set across all methods, using the threshold τ that produced the outcomes reported in Tables 2–4.

After the calculation of F_{max} score for each protein in the test set, we applied Iman–Davenport test and compared the results of F_F with the critical value of the F-distribution with 95% confidence interval. Table 5 presents the results for comparisons between machine learning methods and ensembles combining machine learning with local alignment tools. In all analyses, F_F exceeded the critical values, indicating that the null hypothesis can be rejected and that statistical differences exist between the approaches.

Based on this, we employed the Nemenyi test to evaluate the CD between the approaches and determine which ones are statistically equivalent. The CD values for each scenario are presented in Table 5. The outcomes for each ontology, comparing machine learning methods and ensembles with local alignment tools, are detailed in the Data S1—Statistical Analysis. These outcomes indicate that both SUPERMAGO and SUPERMAGO+ ranked first among the methods and showed a statistically significant difference from the second-best approaches across all ontologies.

3.3 | Ontology Analysis

Next, we evaluated each approach based on their predictions by analyzing the F_{max} values for each level in the ontologies. We performed a breadth-first search starting from the root node of each ontology to determine the level of each term, taking into account the relationships between terms, such as parent nodes and ancestry. The graphical results for each ontology are presented in the Data S1—Ontology Analysis.

TABLE 4 | Evaluation of the approaches on the MFO test set.

Method	F_{max}	F^*_{max}	wF_{max}	S_{min}	AuPRC	IAuPRC
PFmulDL	0.693	0.628	0.568	7.494	0.622	0.717
DeepGO	0.465	0.387	0.295	10.800	0.149	0.295
DeepGOCNN	0.688	0.622	0.565	7.499	0.706	0.697
TALE	0.685	0.618	0.556	7.492	0.584	0.717
ATGO	0.749	0.700	0.650	6.416	0.701	0.793
TEMPROT	0.762	0.716	0.671	6.117	0.662	0.806
PU-GO	0.703	0.642	0.580	7.368	0.547	0.720
PROTGOAT	0.794	0.756	0.716	5.318	0.732	0.855
MAGO	0.793	0.755	0.716	5.405	0.793	0.851
SUPERMAGO	**0.802**	**0.766**	**0.727**	**5.257**	**0.827**	**0.865**
DIAMOND	0.742	0.712	0.679	5.579	0.454	0.716
BLASTp	0.763	0.728	0.692	5.613	0.553	0.774
DeepGOPlus	0.784	0.744	0.709	5.573	0.832	0.825
TALE+	0.770	0.727	0.685	5.777	0.707	0.820
ATGO+	0.787	0.749	0.710	5.557	0.749	0.846
TEMPROT+	0.794	0.756	0.717	5.437	0.734	0.849
PU-GO+Diamond	0.793	0.753	0.717	5.453	0.689	0.829
MAGO+	0.801	0.765	0.728	5.285	0.806	0.860
SUPERMAGO+	**0.809**	**0.775**	**0.739**	**5.100**	**0.839**	**0.874**

Note: The best results are highlighted in bold and the second-best results are underlined.

TABLE 5 | Statistical analysis for machine learning and ensemble of machine learning with local alignment tools.

	Machine learning			Ensembles		
Statistic	**BPO**	**CCO**	**MFO**	**BPO**	**CCO**	**MFO**
F_F	1176.82	684.30	1786.84	206.57	264.10	133.19
Critical value	1.88	1.88	1.88	1.88	1.88	1.88
Critical distance	0.14	0.14	0.15	0.14	0.14	0.15

For BPO, among machine learning approaches, SUPERMAGO consistently achieved the highest scores for F_{max} across all levels. In the ensemble of machine learning with local alignment tools, SUPERMAGO+ demonstrated competitive results up to level 6, with moderate scores from levels 7 up to 9. In terms of CCO, both SUPERMAGO and SUPERMAGO+ presented competitive results at all levels of the ontology. The same trend occurred for MFO with both methods.

3.4 | Domain Analysis

Another crucial aspect of protein function prediction involves analyzing each approach in relation to different domains. Therefore, we evaluated each method based on the predictions for proteins in the test set, considering the domains Archaea, Bacteria, and Eukaryota, as detailed in the Domain Analysis.

In all ontologies, SUPERMAGO achieved the highest scores across all domains, with a notable emphasis on Archaea in BPO. Meanwhile, SUPERMAGO+ delivered the best results for Archaea and Bacteria across all domains, and showed competitive performance for Eukaryota in BPO, CCO, and MFO.

3.5 | Similarity Analysis

In our subsequent analysis, we explored the performance of each approach based on the similarity between proteins in the test set and those in the training set. To do so, we employed CD-HIT [44] to compute this property and divided it into three intervals, from 0 up to 50% (excluding 50%), from 50% up to 75% (excluding 75%), and from 75% up to 100%.

The results are shown in the Data S1—Similarity Analysis. For SUPERMAGO, it obtained the highest scores in these three

intervals for CCO and MFO. In the case of BPO, it achieved the best results for [0%, 50%) and [50%, 75%), and the second-best F_{max} for [75%, 100%], where it was surpassed by DeepGOCNN. For SUPERMAGO+, it obtained the best values for [0%, 50%], which indicates that the model generalizes better than state-of-the-art approaches for sequences with low similarity, and also showed competitive results in other similarity ranges.

In the ensemble scenario within the [75%, 100%] interval for BPO, MAGO+ (0.85), and DeepGOPlus (0.59) exhibited the highest weights for local alignment prediction. Both methods utilize Equation (2) for the ensemble. Other methods employing this ensemble approach, such as TALE+ (0.50) and ATGO (0.40), showed lower performance in this scenario. In our case, the weights for each term ranged from 0.24 to 0.66. Based on this, DeepGOPlus and MAGO+ achieved the best performance in this interval.

3.6 | Frequency Analysis

An important aspect of protein function classification involves analyzing terms with varying frequencies, that is, evaluating the classifier's performance in identifying both common and rare terms. To address this, we conducted an analysis based on term frequency, dividing the terms into three intervals, from 0% up to 5% frequency (excluding 5%), from 5% up to 10% frequency (excluding 10%), and from 10% up to 100%. Table 6 presents the number of terms in each interval for each ontology.

During the evaluation of each approach, we utilized the F_{max} metric. The results are detailed in the Data S1—Frequency Analysis. Based on the outcomes, both SUPERMAGO and SUPERMAGO+ achieved the best results across all three intervals, with the largest margin in the (0%, 5%) and [5%, 10%) frequency ranges, which include rare terms.

TABLE 6 | The number of terms in each interval in the frequency analysis.

	BPO	CCO	MFO
(0%, 5%)	370	462	479
[5%, 10%)	70	14	12
[10%, 100%]	60	22	8
Total	500	498	499

TABLE 7 | Ablation study of the usage of only ESM2 T36, ProtT5, and both models (SUPERMAGO version) on the validation set.

Last layers	BPO	CCO	MFO
ESM2 T36	0.575	0.769	0.797
ProtT5	0.575	0.767	0.799
SUPERMAGO	**0.584**	**0.774**	**0.804**

Note: The best results are highlighted in bold.

3.7 | Filtered Test Set Analysis

We conducted an additional evaluation using a filtered test set, removing proteins with any alignment of at least 50 amino acids in length and at least 80% sequence identity to the training set, using DIAMOND. This process resulted in the removal of 4323, 4254, and 3280 proteins from the original test set for BPO, CCO,

TABLE 8 | Ablation study of the number of layer for each backbone of SUPERMAGO on the validation set.

Last layers	BPO	CCO	MFO
One	0.571	0.766	0.794
Two	0.577	0.771	0.799
Three	0.579	0.773	0.801
Four	0.580	0.773	0.803
Five	**0.584**	**0.774**	**0.804**
Six	**0.584**	**0.774**	0.802
Seven	0.582	**0.774**	0.800

Note: The best results are highlighted in bold.

TABLE 9 | Ablation study of classifier prediction aggregation for SUPERMAGO on the validation set. The best results are highlighted in bold.

Method	BPO	CCO	MFO
SLC	0.581	0.771	0.802
NN	**0.584**	**0.774**	**0.804**

Note: The best results are highlighted in bold.

TABLE 10 | Ablation study of ensemble of machine learning and local alignment tool classifier for SUPERMAGO+ on the validation set.

Method	BPO	CCO	MFO
BLASTp			
SLC	0.606	0.775	0.804
NN	0.606	0.777	0.805
DIAMOND			
SLC	**0.623**	0.779	0.806
NN	0.621	**0.781**	**0.808**

Note: The best results are highlighted in bold.

TABLE 11 | Comparison between SUPERMAGO+ and SUPERMAGO+Web on the test set.

Method	BPO	CCO	MFO
SUPERMAGO+	**0.619**	**0.777**	**0.809**
SUPERMAGO+Web	0.617	0.775	**0.809**

Note: The best results are highlighted in bold.

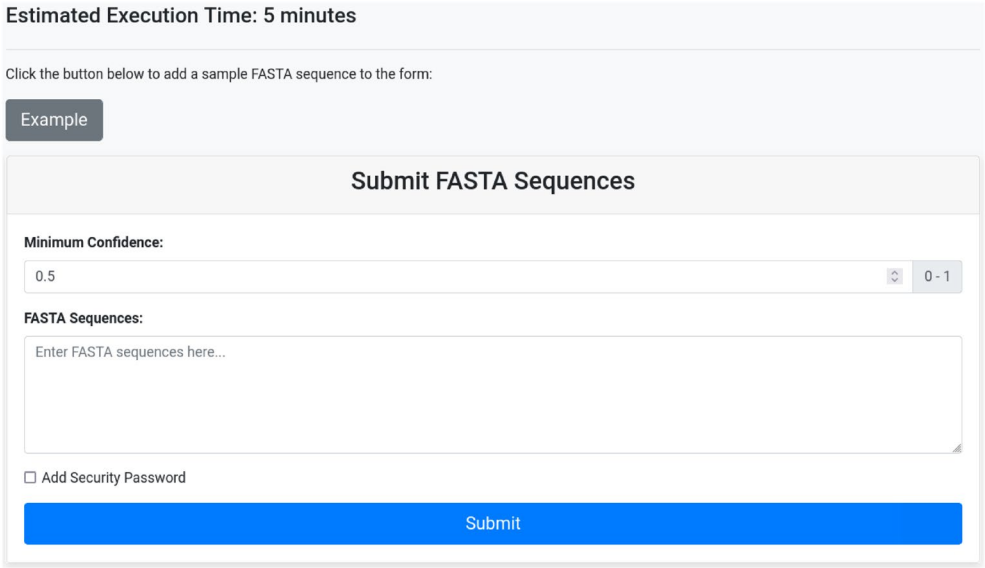

FIGURE 3 | Web server input form. The user must specify the minimum confidence level and provide the protein sequences in FASTA format.

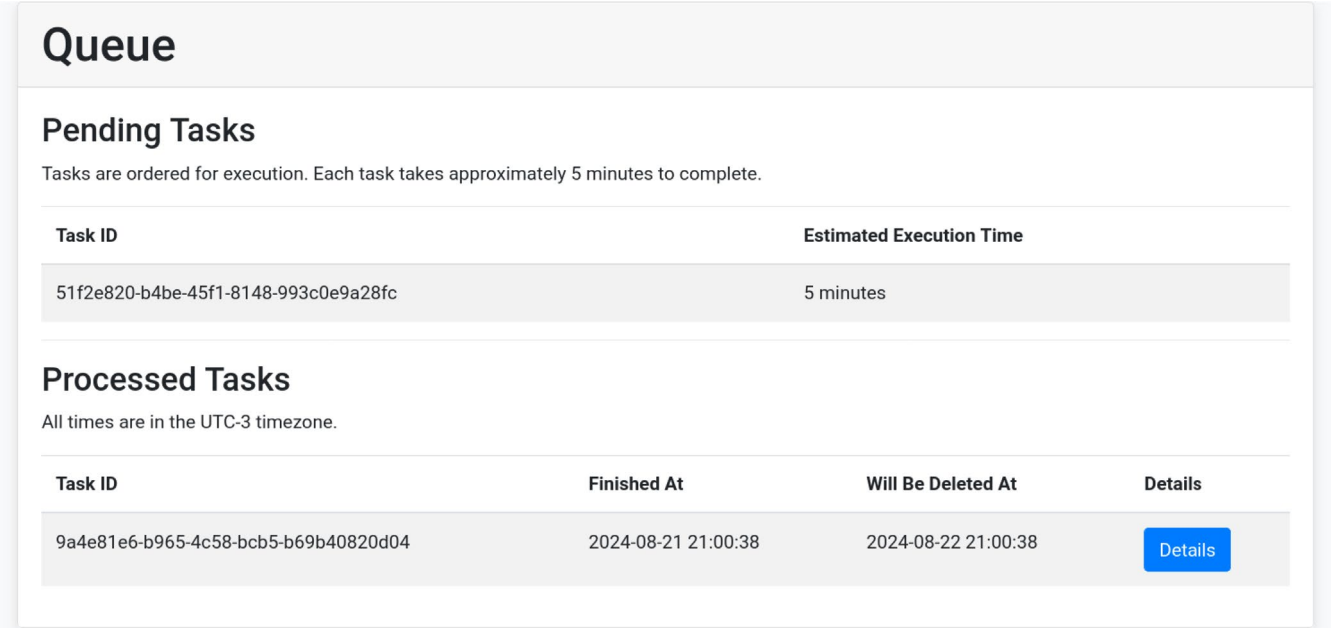

FIGURE 4 | Web server tasks queue. Pending Tasks indicates the tasks currently in the processing queue, while Processed Tasks displays the tasks that have already been completed.

and MFO, respectively. Consequently, the filtered test set comprises 4898, 5038, and 4584 proteins for BPO, CCO, and MFO, respectively.

Next, we evaluated the methods using the metrics F_{max}, F^*_{max}, wF_{max}, S_{min}, AuPRC, and IAuPRC. The results are detailed in the Data S1—Filtered Test Set Analysis. Based on the results for each scenario, our approach achieved all the best results for BPO and MFO, as well as F_{max}, F^*_{max}, wF_{max}, S_{min}, and IAuPRC for CCO. For AuPRC in CCO, our method obtained the second-best score, both in the machine learning approach and the ensemble with local alignment. Thus, we achieved 34 out of 36 best results in the evaluations.

Then, we assessed the statistical differences between classifiers using the Iman–Davenport and Nemenyi tests. For the Iman–Davenport test, we compared the F_F value with the critical value following an F-distribution with a 95% confidence interval, as detailed in the Data S1—Filtered Test Set Analysis. The results indicated that, in all cases, F_F exceeded the critical value, rejecting the null hypothesis and confirming a statistical difference between the models.

Finally, we applied the Nemenyi test to compute the CD and generated visualizations for each scenario, available in the Data S1—Filtered Test Set Analysis. The results showed that both SUPERMAGO and SUPERMAGO+ achieved the

Protein Name: Q5A3Z5

Protein Sequence:

MLLNQYTFSAERCTTPTSISHPPLACPTSPPPTNSLNRYQHKPVTLAPTVDHGCEINGTTIDCIPKDSKFYQLIMDLKQLLAGKGLSNEDIDVEKVKQLMADYDSNEID
WQHLALHDPSRNYSRNGIINLNGNANLLILVWSPGKSSAIHDHADAHCCVKMLAGELIEHLYDFPDHEGEELQCRQETSMKRNDVGYINDSIGLHKMSNPLQNRVSV
SLHLYTPPYASMYGCSMYEASSGRKHHVDMSKYYSWQGQLVNEKESSTC

Predicted Functions:

Biological Process

Term	Description	Confidence
GO:0009987	cellular process	0.77
GO:0008152	metabolic process	0.74
GO:0071704	organic substance metabolic process	0.70
GO:0006807	nitrogen compound metabolic process	0.61
GO:1901564	organonitrogen compound metabolic process	0.61
GO:0044237	cellular metabolic process	0.56
GO:0050896	response to stimulus	0.54
GO:0044238	primary metabolic process	0.41

FIGURE 5 | Web server output. This table represents the results of BPO. Similar tables are generated for CCO and MFO.

top-ranking positions and demonstrated a statistical difference compared with the second-ranked approach in all evaluations.

3.8 | Ablation Study

To evaluate the impact of different configurations of our method, we conducted an ablation study on the validation set for SUPERMAGO and SUPERMAGO+.

In the first analysis, we evaluated the impact of using the ESM2 T36 and ProtT5 classifiers individually in the stacking model compared with SUPERMAGO. The results, shown in Table 7 considering F_{max}, demonstrate that using only the five classifiers based on ESM2 T36 or ProtT5 embeddings in the stacking model produced worse outcomes compared to using both in the SUPERMAGO configuration.

Next, we assessed the results for the use of one up to seven last layers of ESM2 T36 and ProtT5 for SUPERMAGO. The outcomes of this study are presented in Table 8, considering the F_{max} score. The ensemble of the five predictions for ESM2 T36 and ProtT5 yielded the best results across all domains compared to configurations with different numbers of layers, even achieving ties for BPO and CCO.

In our third ablation study, we evaluated the use of a stacking-based neural network classifier for SUPERMAGO and the modified stacking approach for SUPERMAGO+, comparing these with the ensemble approach commonly applied in the literature. In the literature, ensemble methods for machine learning with protein local alignment typically use a single α value to combine predictions, as described in the Equation (2).

Since the literature methods do not use ensembles of base models for the machine learning classifier, we applied a similar concept to aggregate the base classifiers of SUPERMAGO by performing a simple linear combination. Table 9 presents the results of the experiment considering the F_{max} score, where SLC denotes the simple linear combination and NN refers to the stacking applied through neural network (the approach utilized in SUPERMAGO, presented in Section 2), which indicates that the stacking method applied in our methodology achieves better results across all ontologies compared to a simple linear combination.

Subsequently, we conducted the same experiment to generate SUPERMAGO+, evaluating both SLC and NN configurations. For the SLC version, the base method used is SUPERMAGO with SLC aggregation, as presented in the first row of Table 9. In addition to comparing aggregation methods, we also compared the local alignment tools BLASTp and DIAMOND, as shown in Table 10. The results indicate that in both scenarios (BLASTp or DIAMOND), NN outperformed SLC in the number of ontologies where it achieved superior performance. Regarding the local alignment tools, DIAMOND achieved better F_{max} results compared to BLASTp.

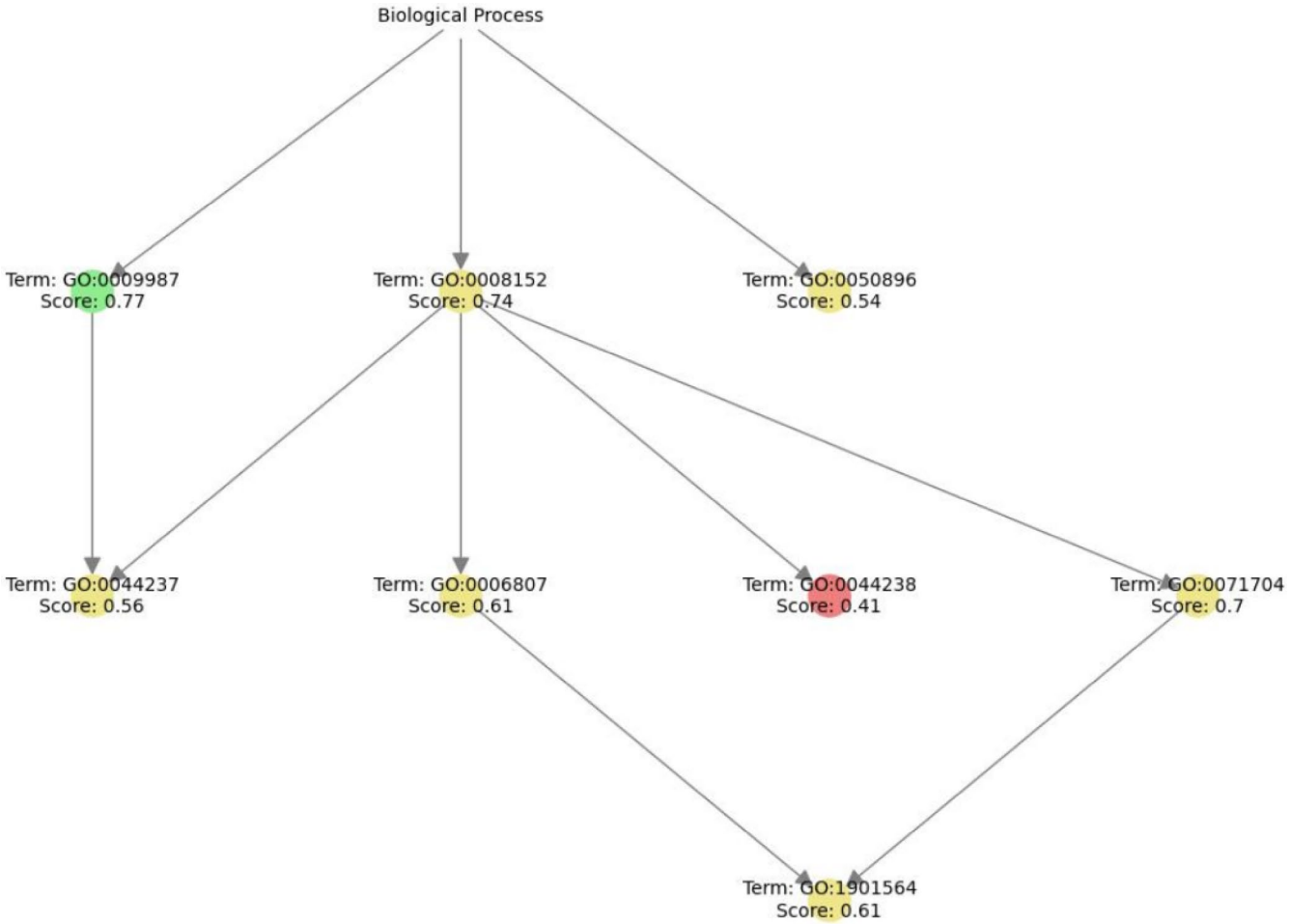

FIGURE 6 | Web server output visualization. This graph represents the results of BPO. Similar graphs are generated for CCO and MFO.

TABLE 12 | Comparison of SUPERMAGO+Web and NetGO 3.0 web server on the test set.

Method	F_{max}	F^*_{max}	wF_{max}	S_{min}	AuPRC	IAuPRC
BPO						
SUPERMAGO+Web	0.617	0.589	0.563	17.785	0.636	**0.664**
NetGO 3.0 web server	**0.650**	**0.622**	**0.601**	**16.467**	**0.674**	0.658
CCO						
SUPERMAGO+Web	**0.775**	**0.750**	**0.653**	9.015	**0.813**	**0.853**
NetGO 3.0 web server	0.768	0.741	**0.653**	**8.475**	0.723	0.847
MFO						
SUPERMAGO+Web	**0.809**	**0.775**	0.739	5.128	**0.838**	0.871
NetGO 3.0 web server	0.807	0.770	**0.742**	**4.662**	0.799	**0.872**

Note: The best results are highlighted in bold.

3.9 | Comparison Between SUPERMAGO+ and SUPERMAGO+Web

Considering SUPERMAGO+ and SUPERMAGO+Web, we compared these two approaches with respect to F_{max}, as shown in Table 11. The results indicate that SUPERMAGO+Web achieves results comparable to SUPERMAGO+, with a tie in MFO.

Next, we assessed the statistical differences between the methods, as detailed in the Data S1—Statistical Analysis. According to the Nemenyi test, there was no statistically significant

difference between the two methods. Therefore, we consider both methods to be statistically equivalent.

In terms of computational cost, SUPERMAGO+ requires a graphics card with at least 16 GB of memory, whereas SUPERMAGO+Web can be executed on a GPU with as little as 4 GB of memory, representing only a quarter of the memory required by SUPERMAGO+.

3.10 | Web Server

For the web server (https://supermago.ic.unicamp.br), we utilized SUPERMAGO+Web implemented on a NVIDIA GeForce GTX 980 GPU with 4GB of memory. Thanks to the use of quantization in `int4`, LoRA, and `bfloat16` computation type, we were able to deploy this model on a GPU with limited computational resources.

In the following subsections, we describe the process of submitting protein sequences and visualizing the results.

3.10.1 | Web Server Input

The sequence submission form is shown in Figure 3. Users can submit up to 50 protein sequences or total of 100 000 characters in FASTA format. In addition to the sequences, users must set a minimum confidence for the model to consider a term as predicted. There is also an option to add a security password to restrict access to the results. After completing the submission process, users receive an identifier to track the progress of the task.

3.10.2 | Web Server Queue

Once the task is submitted, users can check its position in the queue under the Pending Tasks section, as illustrated in Figure 4. Each submission takes approximately 5 min to complete. After execution, the task moves to the Processed Tasks and remains available for 24 h.

3.10.3 | Web Server Output

When accessing the task results, users can view the predicted terms for each protein, along with the minimum confidence value indicated for BPO, CCO, and MFO, as shown in Figure 5.

In addition to the table with term and confidence information by ontology, a graphical result will be displayed, following the organization of each ontology in the Gene Ontology. Terms with confidence levels of 75% or higher are shown in green, those with confidence levels between 50% and 75% in yellow, and those below 50% in red, as depicted in Figure 6.

Finally, users can download the results in CSV and JSON formats, as well as figures representing the model's predictions.

3.11 | Comparison Between SUPERMAGO+Web and NetGO 3.0 Web Server

In this comparison, we assessed the performance of SUPERMAGO+Web compared with NetGO 3.0 web server, a tool for protein function prediction based on the NetGO 3.0 [23] classifier. This model employs machine learning, as well as local alignment through BLASTp [24], and incorporates additional data such as family, textual information, and protein–protein interaction network features. To evaluate the NetGO 3.0 web server, we reached out to the authors and ran our test set of protein sequences on their server.

The results from both web servers are presented in Table 12. For CCO and MFO, our approach achieved the best results in 8 of 12 metrics. However, in the BPO, NetGO 3.0 achieved best performance in 5 out of 6 metrics. These findings suggest that the additional features utilized by NetGO 3.0 have contributed to its improved efficacy in predicting this terms. This improvement is likely due to the ontology's focus on biological processes analysis, which are largely influenced by protein interactions [45]. In contrast, for the other two ontologies, our approach remains competitive, relying solely on the amino acid sequence, which is the most available information about proteins.

4 | Conclusions

In this work, we introduce SUPERMAGO, a novel method for predicting protein functions. This method leverages embeddings from the final five layers of the ESM2 T36 and ProtT5 encoder, applies a multilayer perceptron classification model for each layer, and utilizes a neural network stacking classifier to aggregate the predictions. We also present SUPERMAGO+, an ensemble approach combining SUPERMAGO with DIAMOND, which incorporates term-specific weights differently from existing methods in the literature. Finally, we introduce SUPERMAGO+Web, a web-compatible version of SUPERMAGO+ that requires fewer computational resources.

The significance of these methods lies in their potential to reveal previously unknown functions using only the amino acid sequence information, the most abundant data about proteins. By improving the accuracy and specificity of protein function predictions, SUPERMAGO and its variants can offer insights into protein roles, potentially contributing to advancements in this field of study.

During the evaluation phase, SUPERMAGO outperformed state-of-the-art approaches in 17 out of 18 comparisons. It also demonstrated competitive results in analyses of ontology, domain, similarity, term frequency, and filtered test set. Our statistical analysis revealed significant differences compared to existing methods across all evaluations. In terms of using stacking-based neural networks instead of linear combination, the former approach yielded better results. This indicates that assigning specific weights to each term based on the prediction method is more effective than using a single weight for all terms within a specific classifier. In the stacking process, by employing multiple individual models, the error of each model can be

mitigated by the others, resulting in a final prediction that can be more accurate than a single robust model.

SUPERMAGO+ similarly exhibited competitive performance, surpassing state-of-the-art methods in 16 out of 18 comparisons, showing improvements with statistically significant differences compared with the second-best approach and competitive results on ontology, domain, similarity, term frequency, and filtered test set analysis. Regarding local alignment tools, DIAMOND yielded superior outcomes compared with BLASTp when used in conjunction with SUPERMAGO. Furthermore, when comparing stacking-based neural networks to linear combination, the neural network approach again delivered the best results. This indicates that assigning specific weights to each model prediction (SUPERMAGO or DIAMOND) leads to better outcomes than using a single weight value during the ensemble process.

Considering our SUPERMAGO+Web method, it demonstrated competitive results compared with SUPERMAGO+, and presented no statistical difference between them. When compared to NetGO 3.0, SUPERMAGO+Web obtained competitive results on CCO and MFO, even using only the amino acid sequence information. In the case of BPO, NetGO 3.0 consistently achieved the best results, indicating that additional features can help to improve the results on this ontology.

Due to the reliance on data from CAFA5 for developing SUPERMAGO, SUPERMAGO+, and SUPERMAGO+Web, our models may be impacted by underrepresented species. Therefore, future work will focus on exploring data from additional species beyond those represented in the current dataset, including a new version of SUPERMAGO+Web. Another topic of investigation will be the integration of other types of data, such as protein–protein interactions, structural information, and taxonomic data, which could enhance the performance of each model.

Author Contributions

Gabriel Bianchin de Oliveira: conceptualization, data curation, formal analysis, investigation, methodology, project administration, resources, software, visualization, writing – original draft, writing – review and editing. **Helio Pedrini:** conceptualization, funding acquisition, supervision, validation, writing – review and editing. **Zanoni Dias:** conceptualization, funding acquisition, validation, writing – review and editing, supervision.

Acknowledgments

This work was supported by the São Paulo Research Foundation (grant number 2017/12646-3), the National Council for Scientific and Technological Development (grant numbers 302530/2022-3, 304836/2022-2, 161015/2021-2), the Coordination for the Improvement of Higher Education Personnel, and Santander Bank—Brazil. The authors would like to thank LNCC/MCTI for providing HPC resources of the SDumont supercomputer and Centro Nacional de Processamento de Alto Desempenho em São Paulo (CENAPAD-SP) for providing computational resources.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The data that support the findings of this study are openly available in SUPERMAGO at https://github.com/gabrielbianchin/SUPERMAGO.

Peer Review

The peer review history for this article is available at https://www.webofscience.com/api/gateway/wos/peer-review/10.1002/prot.26782.

References

1. T. Hu, N. Chitnis, D. Monos, and A. Dinh, "Next-Generation Sequencing Technologies: An Overview," *Human Immunology* 82, no. 11 (2021): 801–811.

2. P. Radivojac, *A (Not So) Quick Introduction to Protein Function Prediction* (USA: Indiana University, 2013).

3. A. G. McDonald and K. F. Tipton, "Enzyme Nomenclature and Classification: The State of the Art," *FEBS Journal* 290, no. 9 (2023): 2214–2231.

4. A. Ruepp, A. Zollner, D. Maier, et al., "The FunCat, a Functional Annotation Scheme for Systematic Classification of Proteins From Whole Genomes," *Nucleic Acids Research* 32, no. 18 (2004): 5539–5545.

5. M. Ashburner, C. A. Ball, J. A. Blake, et al., "Gene Ontology: Tool for the Unification of Biology," *Nature Genetics* 25, no. 1 (2000): 25–29.

6. Q. Gong, W. Ning, and W. Tian, "GoFDR: A Sequence Alignment Based Method for Predicting Protein Functions," *Methods* 93 (2016): 3–14.

7. P. Törönen, A. Medlar, and L. Holm, "PANNZER2: A Rapid Functional Annotation Web Server," *Nucleic Acids Research* 46, no. W1 (2018): W84–W88.

8. G. Zehetner, "OntoBlast Function: From Sequence Similarities Directly to Potential Functional Annotations by Ontology Terms," *Nucleic Acids Research* 31, no. 13 (2003): 3799–3803.

9. M. Kulmanov and R. Hoehndorf, "DeepGOPlus: Improved Protein Function Prediction From Sequence," *Bioinformatics* 36, no. 2 (2019): 422–429.

10. Y. Cao and Y. Shen, "TALE: Transformer-Based Protein Function Annotation With Joint Sequence–Label Embedding," *Bioinformatics* 37, no. 18 (2021): 2825–2833.

11. F. Zhapa-Camacho, Z. Tang, M. Kulmanov, and R. Hoehndorf, "Predicting Protein Functions Using Positive-Unlabeled Ranking With Ontology-Based Priors," *bioRxiv* (2024): 1–9.

12. G. B. Oliveira, H. Pedrini, and Z. Dias, "TEMPROT: Protein Function Annotation Using Transformers Embeddings and Homology Search," *BMC Bioinformatics* 24, no. 1 (2023): 1–16.

13. Y. H. Zhu, C. Zhang, D. J. Yu, and Y. Zhang, "Integrating Unsupervised Language Model With Triplet Neural Networks for Protein Gene Ontology Prediction," *PLoS Computational Biology* 18, no. 12 (2022): e1010793.

14. G. B. Oliveira, H. Pedrini, and Z. Dias, "Integrating Transformers and AutoML for Protein Function Prediction," in *46th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (Orlando: IEEE, 2024), 1–5.

15. Z. Wu, M. Guo, X. Jin, J. Chen, and B. Liu, "CFAGO: Cross-Fusion of Network and Attributes Based on Attention Mechanism for Protein Function Prediction," *Bioinformatics* 39, no. 3 (2023): btad123.

16. S. Wang, H. Cho, C. Zhai, B. Berger, and J. Peng, "Exploiting Ontology Graph for Predicting Sparsely Annotated Gene Function," *Bioinformatics* 31, no. 12 (2015): i357–i364.

17. B. Sarker, N. Khare, M. D. Devignes, and S. Aridhi, "Improving Automatic GO Annotation With Semantic Similarity," *BMC Bioinformatics* 23, no. 433 (2022): 1–19.

18. F. V. Song, J. Su, S. Huang, et al., "DeepSS2GO: Protein Function Prediction From Secondary Structure," *Briefings in Bioinformatics* 25, no. 3 (2024): bbae196.

19. M. Kulmanov, M. A. Khan, and R. Hoehndorf, "DeepGO: Predicting Protein Functions From Sequence and Interactions Using a Deep Ontology-Aware Classifier," *Bioinformatics* 34, no. 4 (2018): 660–668.

20. Z. M. Chua, A. Rajesh, S. Sinha, and P. D. Adams, "PROTGOAT: Improved Automated Protein Function Predictions Using Protein Language Models," *bioRxiv* (2024): 1–15, https://doi.org/10.1101/2024.04.01.587572.

21. Z. Huang, R. Zheng, and L. Deng, "DeepFusionGO: Protein Function Prediction by Fusing Heterogeneous Features Through Deep Learning," in *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (Las Vegas: IEEE, 2022), 12–17.

22. K. Wu, L. Wang, B. Liu, Y. Liu, Y. Wang, and J. Li, "PSPGO: Cross-Species Heterogeneous Network Propagation for Protein Function Prediction," *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 20, no. 3 (2022): 1713–1724.

23. S. Wang, R. You, Y. Liu, Y. Xiong, and S. Zhu, "NetGO 3.0: Protein Language Model Improves Large-Scale Functional Annotations," *Genomics, Proteomics & Bioinformatics* 21, no. 2 (2023): 349–358.

24. S. F. Altschul, T. L. Madden, A. A. Schäffer, et al., "Gapped BLAST and PSI-BLAST: A New Generation of Protein Database Search Programs," *Nucleic Acids Research* 25, no. 17 (1997): 3389–3402.

25. B. Buchfink, K. Reuter, and H. G. Drost, "Sensitive Protein Alignments at Tree-of-Life Scale Using DIAMOND," *Nature Methods* 18, no. 4 (2021): 366–368.

26. A. Kabir and A. Shehu, "Graph Neural Networks in Predicting Protein Function and Interactions," *Graph Neural Networks: Foundations, Frontiers, and Applications* (2022): 541–556, https://doi.org/10.1007/978-981-16-6054-2_25.

27. W. Xia, L. Zheng, J. Fang, et al., "PFmulDL: A Novel Strategy Enabling Multi-Class and Multi-Label Protein Function Annotation by Integrating Diverse Deep Learning Methods," *Computers in Biology and Medicine* 145 (2022): 105465.

28. L. Breiman, "Random Forests," *Machine Learning* 45 (2001): 5–32.

29. T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *22nd International Conference on Knowledge Discovery and Data Mining (KDD)* (San Francisco: ACM, 2016), 785–794.

30. Z. Lin, H. Akin, R. Rao, et al., "Evolutionary-Scale Prediction of Atomic-Level Protein Structure With a Language Nodel," *Science* 379, no. 6637 (2023): 1123–1130.

31. A. Elnaggar, M. Heinzinger, C. Dallago, et al., "ProtTrans: Toward Understanding the Language of Life Through Self-Supervised Learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, no. 10 (2021): 7112–7127.

32. D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv* (2017): 1–15, https://doi.org/10.48550/arXiv.1412.6980.

33. T. Dettmers and L. Zettlemoyer, "The Case for 4-Bit Precision: K-Bit Inference Scaling Laws," in *40th International Conference on Machine Learning (ICML)* (Honolulu: PMLR, 2023), 7750–7774.

34. Y. Yu, C. H. H. Yang, J. Kolehmainen, et al., "Low-Rank Adaptation of Large Language Model Rescoring for Parameter-Efficient Speech Recognition," in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)* (Taipei, Taiwan: IEEE, 2023), 1–8.

35. A. Vaswani, N. Shazeer, N. Parmar, et al., "Attention Is All You Need," in *Advances in Neural Information Processing Systems (NIPS)* (Long Beach: NIPS, 2017), 5998–6008.

36. A. Rives, J. Meier, T. Sercu, et al., "Biological Structure and Function Emerge From Scaling Unsupervised Learning to 250 Million Protein Sequences," *National Academy of Sciences of the United States of America* 118, no. 15 (2021): e2016239118.

37. A. Elnaggar, H. Essam, W. Salah-Eldin, et al., "Ankh: Optimized Protein Language Model Unlocks General-Purpose Modelling," *arXiv* (2023): 1–29, https://doi.org/10.48550/arXiv.2301.06568.

38. N. Zhou, Y. Jiang, T. R. Bergquist, et al., "The CAFA Challenge Reports Improved Protein Function Prediction and New Functional Annotations for Hundreds of Genes Through Experimental Screens," *Genome Biology* 20, no. 1 (2019): 244.

39. W. T. Clark and P. Radivojac, "Information-Theoretic Evaluation of Predicted Ontological Annotations," *Bioinformatics* 29, no. 13 (2013): 53–61.

40. R. L. Iman and J. M. Davenport, "Approximations of the Critical Region of the Friedman Statistic," *Communications in Statistics—Theory and Methods* 9, no. 6 (1980): 571–595.

41. P. B. Nemenyi, *Distribution-Free Multiple Comparisons* (Princeton: Princeton University, 1963).

42. J. Demšar, "Statistical Comparisons of Classifiers Over Multiple Data Sets," *Journal of Machine Learning Research* 7 (2006): 1–30.

43. G. Valentini, "True Path Rule Hierarchical Ensembles for Genome-Wide Gene Function Prediction," *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 8, no. 3 (2010): 832–847.

44. L. Fu, B. Niu, Z. Zhu, S. Wu, and W. Li, "CD-HIT: Accelerated for Clustering the Next-Generation Sequencing Data," *Bioinformatics* 28, no. 23 (2012): 3150–3152.

45. R. Bonetta and G. Valentino, "Machine Learning Techniques for Protein Function Prediction," *PROTEINS: Structure, Function, and Bioinformatics* 88, no. 3 (2020): 397–413.

Supporting Information

Additional supporting information can be found online in the Supporting Information section.