Evaluation of natural language processing embeddings in protein function prediction for bacteria

Bianca-Maria Cosma[1]
Supervisors: Aysun Urhan[1,2], Abigail Manson[2], Thomas Abeel[1,2]

June 19, 2022

[1]Delft Bioinformatics Lab, Delft University of Technology Van Mourik, Broekmanweg 6, 2628 XE, Delft, The Netherlands; [2]Infectious Disease and Microbiome Program, Broad Institute of MIT and Harvard, 415 Main Street, Cambridge, MA, 02142, USA

A Dissertation Submitted to EEMCS faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering

Abstract

Motivation: The development of automated protein function prediction models is essential in closing the gap between the large amount of protein sequence data available and the fraction of validly annotated data. Recent approaches to function prediction rely on unsupervised deep learning models, through which protein sequences are represented as real-valued embeddings that can be used as input to a machine-learning model. This study aims to evaluate embedding models in the context of protein function prediction on bacteria, which are organisms less commonly included in these types of benchmarks. To this end, we generated embeddings with four recently developed embedding models, and predicted protein function using a nearest-neighbor search in the embedding space. We evaluated these predictors on two query sets, with proteins from gram-positive *B. subtilis* and gram-negative *E. coli*.

Results: Our nearest neighbor models outperformed BLAST sequence-based protein function annotation, according to the evaluation procedure outlined in the CAFA challenges. The results were also shown to be comparable, and at times better than DeepGOPlus predictions, thus highlighting the potential of embedding-based predictions as state-of-the art models. On the *B. subtilis* dataset, our nearest neighbor model from ESM1b embeddings scored an F_{max} of 0.6 in molecular function predictions, and was able to predict GO terms with a high information content. Hence unsupervised embedding models were shown to encode information about a protein sequence that is useful in the task of function prediction.

Availability: The scripts used in this project are available on GitHub.

1 Introduction

Protein function prediction is the process of associating a protein with its role within an organism (Friedberg, 2006), performed either experimentally or through the use of automated prediction models. Among others, this has relevant applications in areas of medicine such as disease prevention, or the development of targeted drugs in personalized treatments (Kulmanov & Hoehndorf, 2020; Wang et al., 2014; Yan et al., 2020). In recent years, following the onset of high-throughput sequencing, automated prediction methods have become crucial in bridging the gap between the large volume of protein sequence data available and the number of proteins with expert-validated annotations. As of 2018, less than 0.1% of about 180 million proteins in the UniProt Knowledgebase have been manually annotated (The UniProt Consortium, 2018).

Automated protein function prediction is a multi-label, hierarchical classification task, in which labels are generally identified by terms belonging to the Gene Ontology (GO) hierarchy. The terms in this hierarchy are connected by different types of relations, uniquely identified and associated with a human-readable definition. For instance, GO:0032502 stands for "developmental process", which is a type of "biological process" (GO:0008150). GO is further split into three sub-hierarchies, which describe categories of functions: molecular function, biological process, and cellular component (Cruz et al., 2017). As of July 2019, the GO database comprises of more than 45,000 different annotations (Gene Ontology Consortium, 2019). This high number of classes, further connected by hierarchical relationships, is part of what makes automated protein function prediction a challenging task.

Another challenge of function prediction is the representation of protein sequences as real-valued feature vectors that become input to a function prediction model. Some recently developed approaches to this representation learning task rely on principles similar to those used in the generation of Natural Language Processing (NLP) embeddings. To that end, protein sequences are regarded as sentences, made of characters that give them meaning, and set in a specific context, depending on neighboring sequences (Iuchi et al., 2021).

Previous studies have explored the performance and potential of protein embedders as unsupervised deep learning models for function prediction. Littmann et al. (2021) proposed goPredSim, a function prediction model which uses a nearest-neighbor search in an embedding space, with Euclidean distance. Using embeddings generated by SeqVec (Heinzinger et al., 2019), this method was shown to perform among the top 10 contestants in the third edition of the Critical Assessment of Functional Annotation, and significantly outperformed a BLAST sequence-based similarity search. These experiments suggest that embedding models are able to capture features beyond sequence similarity. A study conducted by van den Bent et al. (2021) showed that these features are also relevant in cross-species annotation tasks, through the use of a multi-layer perceptron trained on SeqVec embeddings. This classifier achieved more accurate function prediction than DeepGOPlus (Kulmanov & Hoehndorf, 2020), a supervised model designed as a convolutional neural network. The results additionally suggested that SeqVec embeddings can encode protein length.

While extensive effort goes towards the evaluation of protein function prediction models, notably through the large-scale Critical Assessment of Functional Annotation (CAFA) challenges (Jiang et al., 2016; Radivojac et al., 2013; Zhou et al., 2019), organisms such as bacteria are studied less frequently. This is especially true in the case of gram-positive bacteria. Although the third edition of CAFA included gram-positive bacterium *B. subtilis* in its prediction target set, results were not reported individually for these proteins, due to the lower amount of sequence data and experimental annotations available, in comparison to a species such as *H. sapiens*. Nonetheless, gram-positive bacteria are closely linked to the study of human disease, particularly as proven through research in antibiotic resistance and hospital-acquired infections (Jubeh et al., 2020; Rice, 2006). Protein function prediction for gram-positive bacteria is thus a field of research with far-reaching applications.

We hypothesize that deep-learned embeddings are able to capture information about a bacterial protein sequence that is relevant in the prediction of its functions. Consequently, this study provides an evaluation of four commonly used NLP-based protein embedding models on gram-positive *B. subtilis* and gram-negative *E. coli*. While the former is an organism rarely included in these kinds of evaluations, the latter is more commonly studied, and regarded as a model organism (Blount, 2015). We include both to provide a more complete overview for the performance of the prediction models.

To this end, we apply a prediction model based on a k-nearest neighbor (k-NN) search in the embedding space to predict the function of proteins in three categories: the molecular function ontology (MFO), the biological process ontology (BPO), and the cellular component ontology (CCO). Performance is regarded using two approaches, given the metrics proposed as part of the CAFA challenges (Jiang et al., 2016; Radivojac et al., 2013; Zhou et al., 2019). Firstly, we investigate prediction accuracy using the maximum F-measure, which summarizes the rate of false negatives and false positives among the predictions. Secondly, we compute the minimum semantic distance for the models, to provide insight into the information content of the predictions. Performance of the k-NN predictors is regarded given a standard baseline method based on a BLAST search. We note that our approach to this BLAST search is similar to the one described by Radivojac et al. (2013). While Blast2GO (Conesa et al., 2005) remains the standard for annotation transfer based on sequence identity, it was not suited as a baseline in our evaluation, as its pipeline is more complex than standard sequence-based transfer. Lastly, we also evaluate goPredSim (Littmann et al., 2021), a k-NN model similar to ours, and DeepGOPlus (Kulmanov & Hoehndorf, 2020), a representative of state-of-the-art prediction models, ranking among the top entries in the latest CAFA challenge.

2 Materials and methods

In order to evaluate and compare the performance of different embedding tools in protein function prediction on bacteria, we designed an experiment to resemble annotating new protein sequences for gram-positive *B. subtilis* and gram-negative *E. coli*. A database and query set were created for each bacterium, both containing experimentally annotated protein sequences. We then generated embeddings for all sequences using four different tools, and ran a prediction algorithm based on a nearest neighbor search. The results were evaluated using the main metrics introduced in the CAFA challenges (Jiang et al., 2016; Radivojac et al., 2013; Zhou et al., 2019), and benchmarked against a baseline BLAST search, and two other models, namely goPredSim (Littmann et al., 2021) and DeepGOPlus (Kulmanov & Hoehndorf, 2020). An overview of this pipeline is shown in Figure 1.

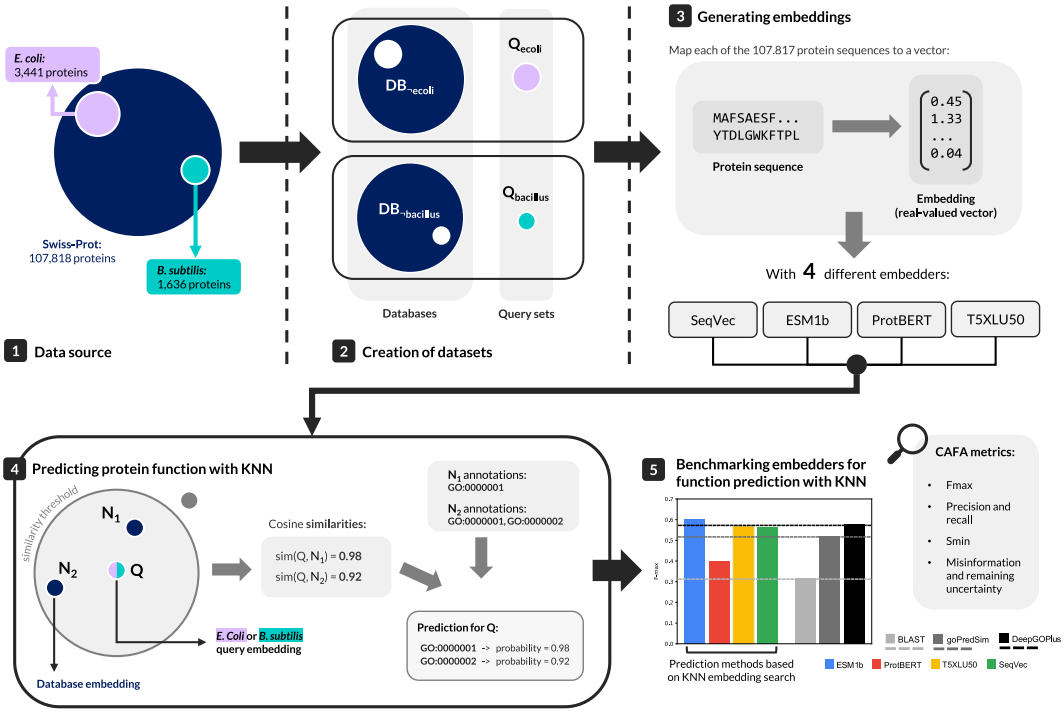

Figure 1: The evaluation pipeline. A set of experimentally annotated protein sequences was retrieved from Swiss-Prot, and two query sets were created, for *E. coli* and *B. subtilis*. Embeddings were generated from all protein sequences, with four different models. These were then benchmarked based on k-NN function predictions in the embedding space.

2.1 The protein sequence dataset

All protein sequences and associated GO terms were retrieved from the Swiss-Prot database, under release 2021_04 of UniProt. The query was performed on November 10, 2021, and restricted to protein sequences that had been experimentally annotated at that time, resulting in a total of 107,818 sequences. The GO annotations for these proteins correspond to those under release 2021-11-16, where one GO term in the dataset annotations was found obsolete: GO:2000775. Lastly, protein sequences with lengths shorter than 40 or longer than 1000 base

pairs were filtered out, as outlined by the methodology proposed by van den Bent et al. (2021).

The focus of this study is on one gram-negative and one gram-positive bacterium, namely *E. coli* strain K-12, substrain MG1655, and *B. subtilis* strain 168, respectively. *E. coli* protein sequences were identified from assembly ASM584v2, with NCBI accession GCF_000005845.2, and *B. subtilis* protein sequences were retrieved from assembly ASM904v1, with NCBI accession GCF_000009045.1.

Two sets of sequences were created for each of the two bacteria: the query set and the database. All proteins from a given bacterium assembly that had been experimentally annotated in Swiss-Prot were added to the query set, while the database contained the remaining protein sequences. We denote these sets as $DB_{\neg bacillus}$, $DB_{\neg ecoli}$, $Q_{bacillus}$ and Q_{ecoli}, with the latter two being the query sets, comprised of 1,636 and 3,441 protein sequences, respectively. A summary of statistics for the protein sets is included in Table 1.

Table 1: Statistics for the databases and query sets created for *B. subtilis* and *E. coli*. To compute the weighted average of GO term depths, we used the number of proteins in the set annotated with a GO term as the weight for the term's depth. The depth of a GO term at the root of the GO hierarchy was set to 0, as were the depths of obsolete GO terms.

	B. subtillis		***E. coli***	
	Query set ($Q_{bacillus}$)	Database ($DB_{\neg bacillus}$)	Query set (Q_{ecoli})	Database ($DB_{\neg ecoli}$)
Number of proteins	**1,636**	**106,182**	**3,441**	**104,377**
Average number of annotations per protein	**2.75**	**6.54**	**5.81**	**6.51**
Molecular function	1.19	1.79	2.27	1.76
Biological process	0.97	2.78	2.03	2.78
Cellular component	0.59	1.98	1.51	1.97
Number of distinct GO terms in the annotations	**1,373**	**25,140**	**3,476**	**24,565**
Molecular function	696	6,303	1,732	6,006
Biological process	602	16,274	1,502	16,087
Cellular component	75	2,563	242	2,472
Weighted average of GO term depths	**5.27**	**5.33**	**5.03**	**5.34**
Molecular function	4.70	4.61	4.68	4.61
Biological process	7.49	6.88	7.07	6.88
Cellular component	2.81	3.80	2.83	3.82

2.2 Protein sequence embedders

We benchmarked four state-of-the-art embedding models commonly used in the literature. This selection was made up of SeqVec (Heinzinger et al., 2019), ESM1b (Rives et al., 2021), ProtBert, and ProtT5XLU50, T5XLU50 for short. The latter two are part of the ProtTrans suite (Elnaggar et al., 2020). All embedders were run with default settings, using version 0.2.2 of the `bio_embeddings` conda package (Dallago et al., 2021).

Each model first produced an embedding for every amino-acid in a given protein sequence, which was then reduced per protein. This was done with the use of the `bio_embeddings` toolkit, by averaging the amino-acid components (Dallago et al., 2021; van den Bent et al., 2021). Following this method, SeqVec, ProtBERT, and T5XLU50 mapped every protein sequence to a vector in a 1024-dimensional space, while ESM1b created embeddings of length 1280. Each of

the four embedders generated feature vectors with a mean approximately equal to 0.

2.3 Predicting protein function

2.3.1 Nearest neighbor search in the embedding space

We predicted protein function with a k-NN search in the embedding space. We retained the GO annotations for the proteins in $DB_{\neg ecoli}$ and $DB_{\neg bacillus}$, and regarded those in Q_{ecoli} and $Q_{bacillus}$ as proteins with unknown function. For each query protein, this method resulted in a multi-label prediction, where each GO term was assigned with a certainty ranging from 0 to 1, depending on its similarity to the sequences in the database. We determined similarity between two embeddings $\vec{e_1}$ and $\vec{e_2}$ using cosine similarity:

$$sim(\vec{e_1}, \vec{e_2}) = \frac{\vec{e_1} \cdot \vec{e_2}}{||\vec{e_1}|| \cdot ||\vec{e_2}||}, \tag{1}$$

where $\vec{e_1}$ and $\vec{e_2}$ are both real-valued vectors, $\vec{e_1} \cdot \vec{e_2}$ represents the dot product between $\vec{e_1}$ and $\vec{e_2}$, and $||\vec{e_i}||$ is the Euclidean norm of vector $\vec{e_i}$, where $i = 1, 2$.

A query protein with an associated embedding was annotated from multiple neighboring embeddings in the database, which were selected based on a similarity threshold. For a given set of embeddings, after computing the pairwise distances between a query protein and the proteins in the database, we computed this threshold as the x^{th} percentile among all positive similarities, where x is a parameter that can be optimized. Compared to a fixed threshold, this approach was adapted to each embedder, and was a better choice in providing a fair comparison between embedding models. In our experiments, we chose the 99.999 percentile, as lower values resulted in too many predictions, while higher values were too restrictive on the number of neighbors considered. Minor tuning of this parameter did not significantly affect performance. The maximum F-measure, mean precision and recall were recorded for different values of parameter x (see Supplementary Tables S1, S2 and S3).

Normalized pairwise cosine similarities were set as the certainty of each prediction. We mapped all similarities above the threshold to the range [0, 1]. Normalization was performed within each of the three GO classes, taking into account all pairwise similarities between proteins in the query set and those in the database. When predicting a common GO term from multiple neighboring sequences, we retained the maximum similarity as the probability of the prediction.

2.3.2 DeepGOPlus predictions

We trained DeepGOPlus v1.0.1 (Kulmanov & Hoehndorf, 2020) on each of the two databases, namely $DB_{\neg bacillus}$ and $DB_{\neg ecoli}$, and generated predictions for the query sets containing proteins from the two bacteria. DeepGOPlus uses one-hot encoding to represent protein sequences, and makes predictions based on the output of a deep convolutional neural network (CNN), combined with BLAST sequence similarity. As shown in an evaluation performed by the authors on the CAFA3 target set (Zhou et al., 2019), DeepGOPlus would have ranked among the three best predictors in the latest edition of the CAFA challenge. To run the model, we used default parameters for the train/validation split, the batch size, the number of training epochs, and the prediction threshold. We also followed the authors' recommendations for other hyperparameters, such as the learning rate and the number of convolutional filters, which were all kept as the default values set for version 1.0.1 of the model.

2.3.3 Function prediction with goPredSim

The goPredSim model was developed by Littmann et al. (2021), as a k-NN search with a fixed number of neighbors, using Euclidean distance. We ran goPredSim with default parameters: $k = 1$ (the number of neighbors), and Euclidean distance as the distance metric. The query sets and databases constructed for each bacterium were used as the target and look-up proteins, respectively. Although the original goPredSim model used Seqvec embeddings (Heinzinger et al., 2019), we ran the model with T5XLU50 embeddings (Elnaggar et al., 2020), as per the updated recommendation of the authors.

2.3.4 BLAST search based on embedding similarity

Our k-NN models were also compared with a baseline BLAST method. To determine the pairwise similarity between protein sequences, commonly known as the BLAST identity, we ran BLASTP v2.12.0. The database sets were used as the subject sequences, while $Q_{bacillus}$ and Q_{ecoli} were used as the query sequences. Similarly to the cosine similarities in the k-NN predictions, pairwise BLAST identities were then mapped to the range [0, 1], normalized separately for each of the three GO classes, and used as the prediction probabilities. No threshold was set for the sequence similarity, but pairwise identities with an Expect (E) value larger than 10^{-3} were filtered out, as commonly done in practice (Jones & Swindells, 2002). A lower E-value suggests that the "hit" in the database was more significant, with a value of 0 indicating a perfect match with regard to the BLAST identity between the query sequence and the database sequence.

2.4 Evaluation of predictions

One of the main evaluation metrics used in this study was the maximum F-measure, which combines the precision and recall of a prediction model. On the one hand, high precision implies that the prediction model rarely identifies GO annotations that are not part of the ground-truth set of labels. This corresponds to a low rate of false positives. On the other hand, high recall values suggest that the rate of false negatives is low, that is, most of the ground-truth GO annotations for a given protein were also identified as part of the prediction set. Consequently, a high value of the F-measure is meant to convey that a model has both optimal precision and recall, and thus produces reliable predictions.

These metrics were computed as proposed by Radivojac et al. (2013):

$$pr(t) = \frac{1}{m(t)} \sum_{i=1}^{m(t)} \frac{|P_i(t) \cap T_i|}{|P_i(t)|}, \tag{2}$$

$$rc(t) = \frac{1}{n} \sum_{i=1}^{n} \frac{|P_i(t) \cap T_i|}{|T_i|}, \tag{3}$$

$$F_{max} = \max_t \left\{ \frac{2 \cdot pr(t) \cdot rc(t)}{pr(t) + rc(t)} \right\}. \tag{4}$$

In equations 2 - 4, t is defined as a threshold value. Precision, $pr(t)$, and recall, $rc(t)$, were calculated over multiple threshold values, determined by discretizing the interval [0, 1]. The total number of proteins for which at least one prediction was made with probability above threshold t is denoted by $m(t)$, while n represents the total number of query proteins. For a given protein i, T_i is the set of ground-truth, experimentally determined functions, and $P_i(t)$ is the set of

functions (GO terms) predicted with certainties higher than threshold t. Precision and recall were used to computed the maximum F-measure (F_{max}) across all thresholds.

While the F-measure gives insight into the accuracy of a prediction model, it does not account for the hierarchical relationships between GO labels and how often they occur among all annotations in the dataset. GO annotations at the top of the hierarchy are easier to predict, more general, and thus less informative than those further down the hierarchy. Performance indicators should include information about whether a model can predict terms that are less frequent among the dataset annotations, or have a higher depth in the hierarchy.

To convey this information in the evaluation, we computed the minimum semantic distance, using remaining uncertainty and misinformation, as defined by Jiang et al. (2016):

$$ru(t) = \frac{1}{n}\sum_{i=1}^{n}\sum_{f} ic(f) \cdot I(f \notin P_i(t) \wedge f \in T_i), \tag{5}$$

$$mi(t) = \frac{1}{n}\sum_{i=1}^{n}\sum_{f} ic(f) \cdot I(f \in P_i(t) \wedge f \notin T_i), \tag{6}$$

$$S_{min} = \min_{t}\left\{\sqrt{ru(t)^2 + mi(t)^2}\right\}. \tag{7}$$

As in the case of precision and recall, we calculated the remaining uncertainty, $ru(t)$, and misinformation, $mi(t)$, for different values of the threshold t, and afterwards determined the minimum semantic distance, S_{min}, across all thresholds (equations 5 - 7). For a protein i, $P_i(t)$ and T_i are defined as in equations 2 and 3. The indicator function I is 1 when the condition inside the parentheses is met, and 0 otherwise, while $ic(f)$ stands for the information content of GO term f. This was calculated as follows:

$$ic(f) = -\log_2\left(\frac{1}{N_c}\sum_{d \in D_f} n_d\right), \tag{8}$$

where N_c is the total number of annotations in the dataset that belong to class c, with c being the GO class of GO term f, that is, c stands for either "molecular function", "biological process", or "cellular component". D_f represents the set of descendants of term f, including f itself, and n_d is the total number of proteins in the dataset annotated with term d. We calculated the information content of a GO term based on the directed acyclic graph (DAG) implementation provided by the GOATOOLS library (Klopfenstein et al., 2018). We considered that two GO terms from the dataset annotations were connected if and only if a relationship existed between them in the DAG generated from the ontology file. The ic of obsolete GO terms was set to 0.

3 Results and discussion

3.1 Predictions from embedding similarity were more accurate than sequence-based annotations, and comparable to results from state-of-the art models

To evaluate the accuracy of protein function prediction from embeddings, we predicted protein function from the output of four different embedders, and calculated the maximum F-measure of the resulting models, F_{max} for short (Figure 2). Embeddings were generated with ESM1b (Rives et al., 2021), SeqVec (Heinzinger et al., 2019), ProtBERT and T5XLU50 (Elnaggar et al.,

2020), and protein function was predicted with a k-NN model. We also included an evaluation of predictions from goPredSim (Littmann et al., 2021), a nearest neighbor model with Euclidean distance, and DeepGOPlus (Kulmanov & Hoehndorf, 2020), a state-of-the-art prediction model that uses a deep convolutional neural network. Overall, the top-performing models across all three GO categories, and for both bacteria query sets, were those that used ESM1b and T5XLU50 embeddings. On the *B. subtilis* query set, ESM1b consistently scored F_{max} values over the 0.5 threshold, reaching as far as 0.6 for MFO. Overall, predictions from goPredSim were similar to the ones generated by our k-NN models. Lastly, in all categories except for CCO, our k-NN predictions from ESM1b embeddings outperformed DeepGOPlus with regard to the F_{max}.

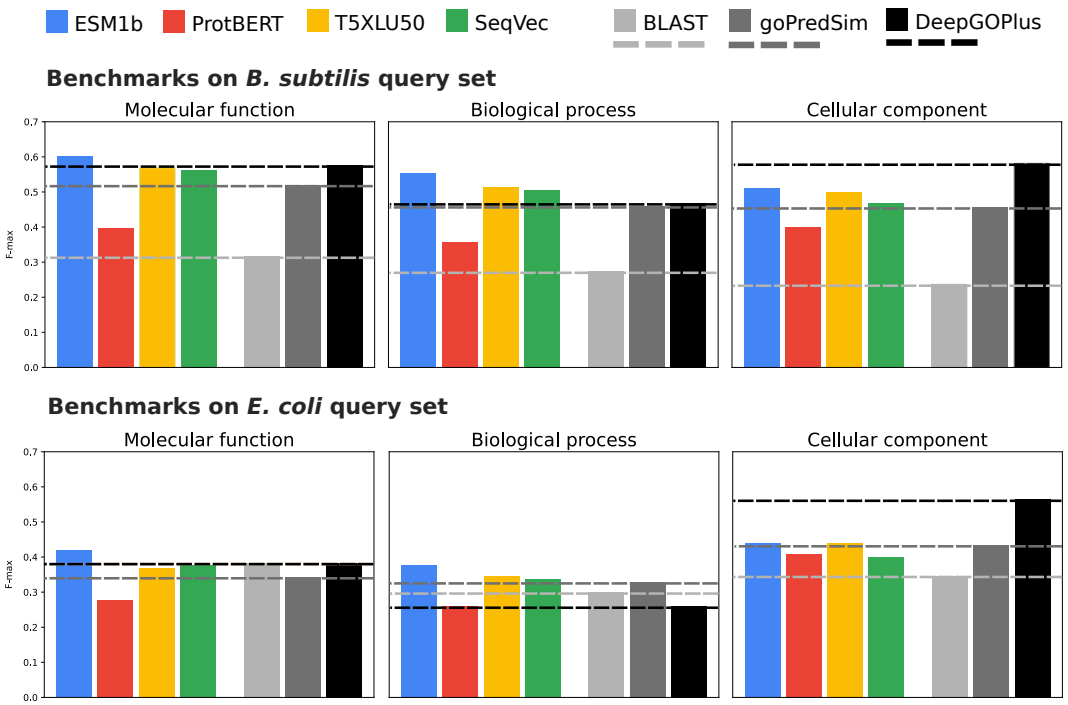

Figure 2: The maximum F-measure of the prediction methods included in the benchmark, on the *B. subtilis* and *E. coli* query proteins. Predictions were made from ESM1b, ProtBERT, SeqVec and T5XLU50 embeddings using a nearest neighbor function transfer, as outlined in subsection 2.3.

To illustrate the performance of these models in a broader context, and support our main hypothesis, we also included results from a baseline BLAST model that uses protein sequence identity. The k-NN prediction models that rely on pairwise embedding similarity were shown to outperform this baseline method in most cases, which implies that unsupervised embeddings capture information additional to raw sequence similarity, suited to the transfer of protein function annotations. This increase in performance was less significant on the gram-negative *E. coli* proteins, but notably overall predictor performance decreased as well on this query set, even for MFO and BPO predictions from DeepGOPlus. We expect that this was partially caused by the difference in the average number of annotations for the two query sets (see Table 1). While, on average, *E. coli* proteins in the query set were annotated with 5.81 ground-truth GO terms, the average for *B. subtilis* protein annotations was less than half of that. We investigated this difference in performance further by looking at the precision and recall of the predictors.

3.2 The performance of all prediction models was brought down by high rates of false negatives

We studied the precision and recall values of the nearest neighbor models by computing an average performance for all thresholds, and comparing it to the precision and recall of the BLAST baseline model and of DeepGOPlus (Figure 3). For each threshold t, defined as in equations 2 and 3, we averaged the precision and recall of goPredSim and our four k-NN models, based on SeqVec, ProtBERT, T5XLU50 and ESM1b embeddings. For precision, these models scored considerably higher than the BLAST baseline, although below DeepGOPlus, which maintained nearly perfect precision in its predictions for the *B. subtilis* query set. Recall was notably lower, making this difference in performance less apparent, particularly for the *E. coli* query set. This further enforced the fact that lower performance in prediction accuracy for this bacterium was in part caused by the higher number of ground-truth annotations available per protein (Table 1).

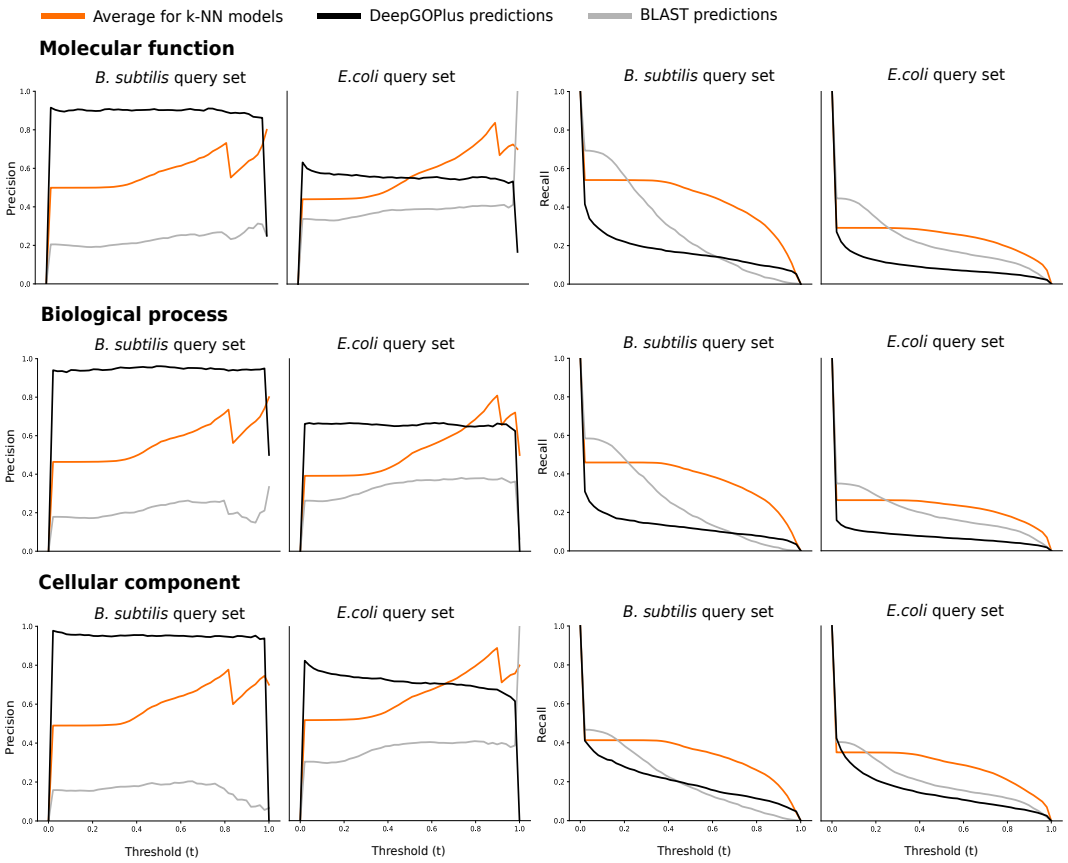

Figure 3: Precision and recall for the k-NN models, the BLAST baseline, and DeepGOPlus, on the *B. subtilis* and *E. coli* query sets. Precision and recall values were averaged for all four of our models that rely on embedding similarity, as well as goPredSim. The threshold t, as introduced in equations 2 and 3, ranges from 0 to 1.

Overall, the k-NN prediction models generally scored high for the maximum F-measure, but further examining their performance showed that the recall for their predictions was not as reliable as the precision. Even though the rate of false positives was low, the models comparatively failed to predict all GO terms associated with a protein. While this could be tackled by lowering

the threshold for the nearest neighbor search, with the desired effect of lowering the rate of false negatives, a possible effect could be lower precision values (see Supplementary Tables S2 and S3). Additionally, as previously shown in Figure 2, the small difference in performance between goPredSim and our prediction models indicated that the choice between cosine similarity and Euclidean distance would not significantly affect recall values, making this parameter a less suitable candidate for optimization. Nonetheless, we note that low recall values were also registered for DeepGOPlus, whose pipeline is more optimized than a standard k-NN search. We attribute part of this to the design of our experiment, which required the prediction of protein function for a given bacterium with no knowledge of annotated proteins from the same organism.

3.3 GO terms with a high information content were transferred through embeddings

To evaluate the effectiveness of using embedding similarity to predict informative protein functions, we compared the minimum semantic distance (S_{min}) of the k-NN models with the ones computed for the baseline BLAST predictor and DeepGOPlus (Figure 4). The rankings of the models for the S_{min} evaluation were consistent with those established by the F_{max} metric, as the models relying on T5XLU50 and ESM1b embeddings remained among the top scoring methods for the S_{min}, and goPredSim still achieved comparable results. Another consistency was kept with regard to the difference in performance between the two query sets, as performance in predicting the function of *E. coli* proteins was comparatively lower for this metric as well.

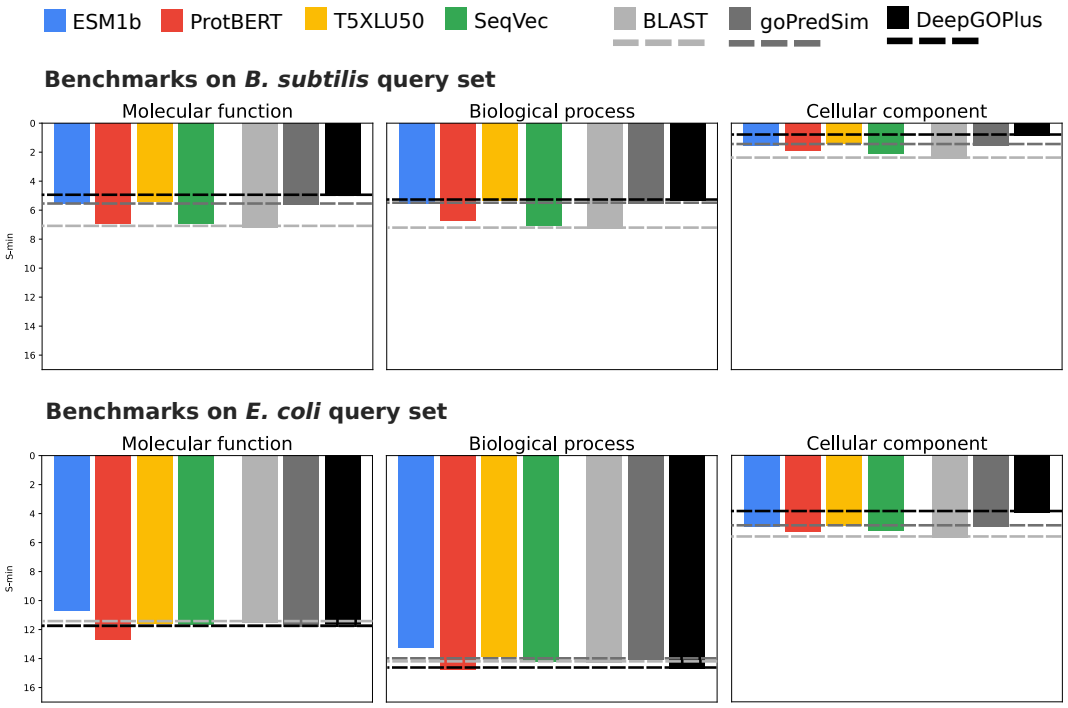

Figure 4: The minimum semantic distance, S_{min}, of the prediction methods included in the benchmark, on the *B. subtilis* and *E. coli* query proteins. Predictions were made from ESM1b, ProtBERT, SeqVec and T5XLU50 embeddings using a k-NN function transfer (see subsection 2.3). A lower S_{min} indicates better performance.

Most notably for gram-positive bacterium *B. subtillis*, the models that used pairwise embedding similarity scored lower minimum semantic distances than the BLAST sequence-based search. Consequently, they were able to predict GO terms with a higher information content. Such GO terms were either located further down the GO hierarchy, meaning that they provided more precise definitions for the functions of a protein, or they occurred more rarely among the database and query set annotations, making them more difficult to predict.

4 Conclusion

The main conclusion of this study confirms our initial hypothesis, namely that unsupervised protein embeddings encode information that aids in the prediction of function for bacteria. We evaluated four embedding models by generating predictions based on a k-NN search in the embedding space, using cosine similarity as a metric and a varying threshold. With regard to the maximum F-measure and minimum semantic distance, the two main metrics used in the latest CAFA challenge (Zhou et al., 2019), these models consistently outperformed a BLAST database search, which relies only on the pairwise identity between raw protein sequences. We also showed that predictions from embedding similarity have an accuracy comparable to state-of-the-art models such as DeepGOPlus (Kulmanov & Hoehndorf, 2020). This suggests that unsupervised embeddings encode features of a protein beyond sequence, and such features are suited for the task of automated protein function prediction.

This conclusion supports the findings of a similar study conducted by Littmann et al. (2021), who developed goPredSim, a nearest neighbor model that uses Euclidean distance. The authors showed that even without optimizing the number of neighbors considered, goPredSim would have ranked among the top ten models of the third CAFA challenge (Zhou et al., 2019). We also included this model in our evaluation, and found that its performance was comparable to that of our k-NN models.

We note that a limitation of the nearest neighbor model itself was reflected by the higher rate of false negatives among the predictions. However, the fact that DeepGOPlus predictions also had low recall suggests that the function prediction task in itself was also challenging. This was likely caused, in part, by the setup of the experiment, which entailed the prediction of function for *B. subtilis* and *E. coli* proteins without any knowledge regarding other proteins belonging to the same organisms. Consequently, this highlights potential directions for future work in the design of automated protein function prediction models, particularly regarding the annotation of novel protein sequences.

5 Responsible research

This study was conducted and documented in a way that allows for the results to be reproducible, by following the procedures outlined in the section on "Materials and methods". We described that our dataset was retrieved from Swiss-Prot, with restrictions on protein length and the validity of the provided GO annotations. The time the query was performed is also specified, alongside the assembly accession numbers for the identification of *B. subtilis* and *E. coli* proteins. The libraries and models that were used in the experiments were referenced with their version, and the approach to the nearest neighbor prediction models, including all relevant parameters, was explained as well. Lastly, we provided the formulas for all metrics that

were used in the evaluation. The implementation of the prediction models and evaluation metrics are made publicly available.

References

Blount, Z. D. (2015). The natural history of model organisms: The unexhausted potential of *E. coli*. *eLife*, *4*, e05826. https://doi.org/10.7554/eLife.05826

Conesa, A., Götz, S., García-Gómez, J. M., Terol, J., Talón, M., & Robles, M. (2005). Blast2GO: a universal tool for annotation, visualization and analysis in functional genomics research. *Bioinformatics*, *21*(18), 3674–3676. https://doi.org/10.1093/bioinformatics/bti610

Cruz, L. M., Trefflich, S., Weiss, V. A., & Castro, M. A. (2017). Protein function prediction. *Methods in Molecular Biology*, *1654*, 55–75. https://doi.org/10.1007/978-1-4939-7231-9_5

Dallago, C., Schütze, K., Heinzinger, M., Olenyi, T., Littmann, M., Lu, A. X., Yang, K. K., Min, S., Yoon, S., Morton, J. T., & Rost, B. (2021). Learned embeddings from deep learning to visualize and predict protein sets. *Current Protocols*, *1*(5), e113. https://doi.org/https://doi.org/10.1002/cpz1.113

Elnaggar, A., Heinzinger, M., Dallago, C., Rihawi, G., Wang, Y., Jones, L., Gibbs, T., Feher, T., Angerer, C., Steinegger, M., Bhowmik, D., & Rost, B. (2020). ProtTrans: Towards Cracking the Language of Life's Code Through Self-Supervised Deep Learning and High Performance Computing. https://doi.org/10.48550/ARXIV.2007.06225

Friedberg, I. (2006). Automated protein function prediction - the genomic challenge. *Briefings in Bioinformatics*, *7*(3), 225–242. https://doi.org/10.1093/bib/bbl004

Gene Ontology Consortium. (2019). The gene ontology resource: 20 years and still GOing strong. *Nucleic Acids Research*, *47*(D1), D330–D338. https://doi.org/10.1093/nar/gky1055

Heinzinger, M., Elnaggar, A., Wang, Y., Dallago, C., Nechaev, D., Matthes, F., & Rost, B. (2019). Modeling aspects of the language of life through transfer-learning protein sequences. *BMC Bioinformatics*, *20*(1), 723. https://doi.org/10.1186/s12859-019-3220-8

Iuchi, H., Matsutani, T., Yamada, K., Iwano, N., Sumi, S., Hosoda, S., Zhao, S., Fukunaga, T., & Hamada, M. (2021). Representation learning applications in biological sequence analysis. *Computational and Structural Biotechnology Journal*, *19*, 3198–3208. https://doi.org/https://doi.org/10.1016/j.csbj.2021.05.039

Jiang, Y., Oron, T. R., Clark, W. T., Bankapur, A. R., D'Andrea, D., Lepore, R., Funk, C. S., Kahanda, I., Verspoor, K. M., Ben-Hur, A., Koo, D. C. E., Penfold-Brown, D., Shasha, D., Youngs, N., Bonneau, R., Lin, A., Sahraeian, S. M. E., Martelli, P. L., Profiti, G., ... Radivojac, P. (2016). An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome Biology*, *17*(1), 184. https://doi.org/10.1186/s13059-016-1037-6

Jones, D. T., & Swindells, M. B. (2002). Getting the most from PSI–BLAST. *Trends in biochemical sciences*, *27*(3), 161–164. https://doi.org/10.1016/S0968-0004(01)02039-4

Jubeh, B., Breijyeh, Z., & Karaman, R. (2020). Resistance of gram-positive bacteria to current antibacterial agents and overcoming approaches. *Molecules*, *25*(12), 2888. https://doi.org/10.3390/molecules25122888

Klopfenstein, D., Zhang, L., Pedersen, B. S., Ramírez, F., Warwick Vesztrocy, A., Naldi, A., Mungall, C. J., Yunes, J. M., Botvinnik, O., Weigel, M., et al. (2018). GOATOOLS:

A Python library for Gene Ontology analyses. *Scientific reports*, *8*(1), 1–17. https://doi.org/10.1038/s41598-018-28948-z

Kulmanov, M., & Hoehndorf, R. (2020). DeepGOPlus: Improved protein function prediction from sequence. *Bioinformatics (Oxford, England)*, *36*(2), 422–429. https://doi.org/10.1093/bioinformatics/btz595

Littmann, M., Heinzinger, M., Dallago, C., Olenyi, T., & Rost, B. (2021). Embeddings from deep learning transfer go annotations beyond homology. *Scientific reports*, *11*(1), 1–14. https://doi.org/10.1038/s41598-020-80786-0

Radivojac, P., Clark, W. T., Oron, T. R., Schnoes, A. M., Wittkop, T., Sokolov, A., Graim, K., Funk, C., Verspoor, K., Ben-Hur, A., Pandey, G., Yunes, J. M., Talwalkar, A. S., Repo, S., Souza, M. L., Piovesan, D., Casadio, R., Wang, Z., Cheng, J., ... Friedberg, I. (2013). A large-scale evaluation of computational protein function prediction. *Nature Methods*, *10*(3), 221–227. https://doi.org/10.1038/nmeth.2340

Rice, L. B. (2006). Antimicrobial resistance in gram-positive bacteria. *American Journal of Infection Control*, *34*(5), S11–S19. https://doi.org/10.1016/j.amjmed.2006.03.012

Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., & Fergus, R. (2021). Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, *118*(15), e2016239118. https://doi.org/10.1073/pnas.2016239118

The UniProt Consortium. (2018). UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Research*, *47*(D1), D506–D515. https://doi.org/10.1093/nar/gky1049

van den Bent, I., Makrodimitris, S., & Reinders, M. (2021). The power of universal contextualized protein embeddings in cross-species protein function prediction. *Evolutionary Bioinformatics*, *17*. https://doi.org/10.1177/11769343211062608

Wang, Z., Liu, P., Inuzuka, H., & Wei, W. (2014). Roles of F-box proteins in cancer. *Nature Reviews Cancer*, *14*(4), 233–247. https://doi.org/10.1038/nrc3700

Yan, L., Lin, M., Pan, S., Assaraf, Y. G., Wang, Z.-W., & Zhu, X. (2020). Emerging roles of F-box proteins in cancer drug resistance. *Drug Resistance Updates: Reviews and Commentaries in Antimicrobial and Anticancer Chemotherapy*, *49*, 100673. https://doi.org/10.1016/j.drup.2019.100673

Zhou, N., Jiang, Y., Bergquist, T. R., Lee, A. J., Kacsoh, B. Z., Crocker, A. W., Lewis, K. A., Georghiou, G., Nguyen, H. N., Hamid, M. N., Davis, L., Dogan, T., Atalay, V., Rifaioglu, A. S., Dalkıran, A., Cetin Atalay, R., Zhang, C., Hurto, R. L., Freddolino, P. L., ... Friedberg, I. (2019). The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. *Genome Biology*, *20*(1), 244. https://doi.org/10.1186/s13059-019-1835-8

Supplementary notes

Table S1: F_{max} values for 10 values of the parameter x, on the *B. subtilis* and *E. coli* query sets. The values $x_1, x_2, ..., x_{10}$ were chosen as equidistant in the interval $[99.99, 100]$, with $x_1 = 99.99$ and $x_{10} = 100$. F_{max} values are reported separately for the molecular function, biological process and cellular component. Recall that we use the x^{th} percentile among the pairwise similarities between a query protein and the database proteins to determine the similarity threshold used in the nearest neighbor search (as described in section 2.3). Note that $x = 99.999 \in (x_9, x_{10})$ is the value used in our evaluation.

	Value of parameter x									
	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
B. subtilis										
Molecular function										
ESM1b	0.405	0.424	0.441	0.459	0.500	0.525	0.548	0.576	0.602	0.610
ProtBERT	0.222	0.233	0.245	0.261	0.295	0.319	0.349	0.375	0.398	0.415
T5XLU50	0.376	0.392	0.407	0.431	0.471	0.493	0.520	0.544	0.568	0.565
SeqVec	0.363	0.379	0.394	0.417	0.458	0.485	0.507	0.538	0.562	0.580
Biological process										
ESM1b	0.376	0.390	0.407	0.422	0.457	0.480	0.502	0.526	0.552	0.565
ProtBERT	0.204	0.211	0.223	0.239	0.272	0.293	0.316	0.337	0.358	0.375
T5XLU50	0.353	0.366	0.380	0.397	0.432	0.451	0.472	0.488	0.514	0.526
SeqVec	0.322	0.338	0.357	0.376	0.410	0.430	0.448	0.476	0.506	0.532
Cellular component										
ESM1b	0.387	0.396	0.405	0.415	0.440	0.455	0.472	0.492	0.510	0.518
ProtBERT	0.294	0.306	0.313	0.323	0.344	0.356	0.374	0.389	0.400	0.397
T5XLU50	0.385	0.396	0.407	0.419	0.443	0.457	0.471	0.486	0.499	0.492
SeqVec	0.340	0.352	0.365	0.378	0.405	0.421	0.438	0.454	0.469	0.484
E. coli										
Molecular function										
ESM1b	0.340	0.349	0.357	0.378	0.389	0.401	0.410	0.413	0.419	0.409
ProtBERT	0.180	0.187	0.197	0.215	0.226	0.239	0.251	0.263	0.278	0.285
T5XLU50	0.264	0.273	0.282	0.307	0.320	0.334	0.345	0.360	0.369	0.367
SeqVec	0.305	0.313	0.320	0.339	0.347	0.358	0.367	0.374	0.378	0.367
Biological process										
ESM1b	0.304	0.313	0.320	0.335	0.344	0.352	0.358	0.366	0.375	0.376
ProtBERT	0.165	0.172	0.180	0.199	0.210	0.222	0.235	0.247	0.261	0.275
T5XLU50	0.243	0.252	0.263	0.285	0.296	0.311	0.323	0.333	0.345	0.349
SeqVec	0.251	0.259	0.266	0.285	0.294	0.305	0.314	0.326	0.337	0.344
Cellular component										
ESM1b	0.381	0.391	0.401	0.418	0.424	0.432	0.437	0.438	0.438	0.431
ProtBERT	0.367	0.374	0.383	0.401	0.410	0.417	0.421	0.417	0.410	0.385
T5XLU50	0.391	0.400	0.408	0.427	0.432	0.440	0.440	0.444	0.440	0.428
SeqVec	0.335	0.344	0.353	0.373	0.384	0.389	0.395	0.400	0.401	0.394

Table S2: Mean precision values for 10 values of the parameter x, on the *B. subtilis* and *E. coli* query sets. The values $x_1, x_2, ..., x_{10}$ were chosen as equidistant in the interval $[99.99, 100]$, with $x_1 = 99.99$ and $x_{10} = 100$. Precision values are reported separately for the molecular function, biological process and cellular component. Mean precision was calculated over different values of threshold t, as defined in equation 2. Recall that we use the x^{th} percentile among the pairwise similarities between a query protein and the database proteins to determine the similarity threshold used in the nearest neighbor search (as described in section 2.3). Note that $x = 99.999 \in (x_9, x_{10})$ is the value used in our evaluation.

	Value of parameter x									
	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
B. subtilis										
Molecular function										
ESM1b	0.294	0.315	0.336	0.358	0.415	0.452	0.489	0.544	0.607	0.690
ProtBERT	0.138	0.151	0.167	0.184	0.237	0.274	0.329	0.395	0.465	0.563
T5XLU50	0.290	0.312	0.331	0.363	0.436	0.471	0.520	0.574	0.638	0.725
SeqVec	0.244	0.261	0.279	0.305	0.357	0.403	0.439	0.495	0.556	0.647
Biological process										
ESM1b	0.280	0.301	0.325	0.348	0.402	0.436	0.475	0.528	0.604	0.700
ProtBERT	0.125	0.138	0.151	0.168	0.221	0.261	0.311	0.386	0.457	0.563
T5XLU50	0.293	0.311	0.333	0.362	0.430	0.471	0.513	0.559	0.628	0.741
SeqVec	0.213	0.230	0.250	0.285	0.335	0.369	0.404	0.456	0.528	0.667
Cellular component										
ESM1b	0.317	0.335	0.355	0.372	0.422	0.457	0.498	0.558	0.638	0.737
ProtBERT	0.240	0.255	0.269	0.288	0.328	0.358	0.403	0.462	0.523	0.626
T5XLU50	0.364	0.384	0.405	0.432	0.500	0.544	0.581	0.629	0.691	0.768
SeqVec	0.234	0.248	0.269	0.292	0.338	0.373	0.409	0.455	0.520	0.641
E. coli										
Molecular function										
ESM1b	0.307	0.324	0.342	0.393	0.423	0.459	0.514	0.560	0.621	0.700
ProtBERT	0.120	0.131	0.146	0.180	0.215	0.246	0.289	0.340	0.410	0.516
T5XLU50	0.226	0.242	0.259	0.307	0.342	0.382	0.422	0.495	0.581	0.668
SeqVec	0.267	0.284	0.301	0.346	0.373	0.406	0.449	0.492	0.552	0.636
Biological process										
ESM1b	0.262	0.281	0.299	0.343	0.372	0.405	0.449	0.492	0.559	0.661
ProtBERT	0.095	0.104	0.116	0.153	0.189	0.222	0.266	0.319	0.393	0.498
T5XLU50	0.193	0.210	0.231	0.281	0.312	0.355	0.396	0.455	0.534	0.647
SeqVec	0.209	0.223	0.242	0.289	0.317	0.348	0.385	0.432	0.498	0.603
Cellular component										
ESM1b	0.369	0.389	0.413	0.460	0.487	0.518	0.554	0.594	0.664	0.751
ProtBERT	0.311	0.333	0.349	0.389	0.416	0.446	0.481	0.516	0.567	0.671
T5XLU50	0.369	0.388	0.408	0.466	0.493	0.530	0.577	0.631	0.687	0.759
SeqVec	0.286	0.302	0.324	0.371	0.402	0.432	0.485	0.533	0.589	0.680

Table S3: Mean recall values for 10 values of the parameter x, on the *B. subtilis* and *E. coli* query sets. The values $x_1, x_2, ..., x_{10}$ were chosen as equidistant in the interval $[99.99, 100]$, with $x_1 = 99.99$ and $x_{10} = 100$. Recall values are reported separately for the molecular function, biological process and cellular component. Mean recall was calculated over different values of threshold t, as defined in equation 3. Recall that we use the x^{th} percentile among the pairwise similarities between a query protein and the database proteins to determine the similarity threshold used in the nearest neighbor search (as described in section 2.3). Note that $x = 99.999 \in (x_9, x_{10})$ is the value used in our evaluation.

	Value of parameter x									
	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
B. subtilis										
Molecular function										
ESM1b	0.646	0.641	0.637	0.634	0.623	0.616	0.604	0.586	0.563	0.512
ProtBERT	0.440	0.437	0.430	0.426	0.413	0.405	0.394	0.374	0.360	0.326
T5XLU50	0.545	0.543	0.539	0.535	0.521	0.515	0.504	0.489	0.472	0.415
SeqVec	0.617	0.613	0.606	0.604	0.590	0.582	0.566	0.554	0.532	0.486
Biological process										
ESM1b	0.530	0.529	0.525	0.524	0.519	0.517	0.512	0.499	0.482	0.444
ProtBERT	0.368	0.364	0.363	0.359	0.347	0.343	0.335	0.320	0.311	0.283
T5XLU50	0.454	0.453	0.452	0.449	0.436	0.429	0.424	0.413	0.404	0.363
SeqVec	0.506	0.504	0.501	0.497	0.489	0.484	0.474	0.468	0.455	0.396
Cellular component										
ESM1b	0.442	0.441	0.441	0.440	0.435	0.433	0.430	0.424	0.409	0.381
ProtBERT	0.387	0.385	0.382	0.379	0.366	0.360	0.353	0.337	0.321	0.281
T5XLU50	0.391	0.390	0.389	0.387	0.383	0.376	0.373	0.369	0.356	0.329
SeqVec	0.434	0.433	0.431	0.428	0.426	0.422	0.416	0.408	0.393	0.365
E. coli										
Molecular function										
ESM1b	0.383	0.378	0.372	0.362	0.354	0.349	0.340	0.325	0.311	0.284
ProtBERT	0.288	0.283	0.278	0.266	0.258	0.251	0.243	0.235	0.226	0.208
T5XLU50	0.337	0.333	0.328	0.317	0.309	0.304	0.297	0.281	0.270	0.249
SeqVec	0.363	0.355	0.348	0.336	0.327	0.318	0.308	0.297	0.280	0.254
Biological process										
ESM1b	0.314	0.310	0.308	0.304	0.300	0.298	0.294	0.288	0.279	0.262
ProtBERT	0.247	0.244	0.241	0.236	0.232	0.229	0.224	0.220	0.213	0.202
T5XLU50	0.289	0.288	0.285	0.280	0.277	0.273	0.269	0.262	0.255	0.237
SeqVec	0.288	0.287	0.281	0.276	0.272	0.268	0.262	0.258	0.249	0.235
Cellular component										
ESM1b	0.402	0.399	0.393	0.381	0.371	0.363	0.352	0.338	0.317	0.292
ProtBERT	0.441	0.436	0.431	0.416	0.405	0.391	0.371	0.345	0.314	0.260
T5XLU50	0.402	0.398	0.392	0.379	0.370	0.359	0.335	0.321	0.303	0.279
SeqVec	0.399	0.395	0.383	0.366	0.357	0.344	0.330	0.314	0.294	0.267