Embeddings from deep learning transfer GO annotations beyond homology

Maria Littmann [1,2,*,◊], Michael Heinzinger [1,2,◊], Christian Dallago [1,2], Tobias Olenyi[1] & Burkhard Rost [1,3,4]

1 TUM (Technical University of Munich) Department of Informatics, Bioinformatics & Computational Biology - i12, Boltzmannstr. 3, 85748 Garching/Munich, Germany
2 TUM Graduate School, Center of Doctoral Studies in Informatics and its Applications (CeDoSIA), Boltzmannstr. 11, 85748 Garching, Germany
3 Institute for Advanced Study (TUM-IAS), Lichtenbergstr. 2a, 85748 Garching/Munich, Germany & TUM School of Life Sciences Weihenstephan (TUM-WZW), Alte Akademie 8, Freising, Germany
4 Department of Biochemistry and Molecular Biophysics, Columbia University, 701 West, 168th Street, New York, NY 10032, USA

* Corresponding author: littmann@rostlab.org, http://www.rostlab.org/
Tel: +49-289-17-814 (email rost: assistant@rostlab.org)
◊ These authors contributed equally to this work

Abstract

Knowing protein function is crucial to advance molecular and medical biology, yet experimental function annotations through the Gene Ontology (GO) exist for fewer than 0.5% of all known proteins. Computational methods bridge this sequence-annotation gap typically through homology-based annotation transfer by identifying sequence-similar proteins with known function or through prediction methods using evolutionary information. Here, we proposed predicting GO terms through annotation transfer based on proximity of proteins in the SeqVec embedding rather than in sequence space. These embeddings originated from deep learned language models (LMs) for protein sequences (SeqVec) transferring the knowledge gained from predicting the next amino acid in 250 million protein sequences. Replicating the conditions of CAFA3, our method reached an F_{max} of 37±2%, 50±3%, and 57±2% for BPO, MFO, and CCO, respectively. This was numerically close to the top ten methods that had participated in CAFA3. Restricting the annotation transfer to proteins with <20% pairwise sequence identity to the query, performance dropped (F_{max} BPO 33±2%, MFO 43±3%, CCO 53±2%); this still outperformed naïve sequence-based transfer. Preliminary results from CAFA4 appeared to confirm these findings. Overall, this new method may help, in particular, to annotate novel proteins from smaller families or proteins with intrinsically disordered regions.

Key words: function prediction, GO term prediction, homology-based inference, natural language processing, unsupervised learning, word embeddings, transfer learning

Abbreviations used: BERT, Bidirectional Encoder Representations from Transformers (particular deep learning language model); **BP(O)**, biological process (ontology) from GO; **CAFA**, Critical Assessment of protein Function Annotation algorithms; **CC(O)**, cellular component (ontology) from GO; **ELMo**, Embeddings from Language Models**; GO**, gene ontology; **GOA**, Gene Ontology Annotation; **k-NN**, k-nearest neighbor; **LK**, limited-knowledge; **LM**, language model**; LSTMs**, Long-Short-Term-Memory cells**; M**, million; **MF(O)**, molecular function (ontology) from GO; **NK**, no-knowledge; **PIDE**, percentage pairwise sequence identity; **RI**, reliability index

Introduction

GO organizes the function of a cell through hierarchical ontologies. All organisms rely on the correct functioning of their cellular workhorses, namely their proteins, that perform almost all functions needed, ranging from molecular functions (MF) such as chemical catalysis of enzymes to biological processes or pathways (BP) such as realized through signal transduction. On top, only the perfectly orchestrated interplay between proteins allows cells to perform more complex functions, e.g., the aerobic production of energy via the citric acid cycle which requires the interconnection of eight different enzymes with some of them being multi-enzyme complexes[1]. The Gene Ontology (GO)[2,3] thrives to capture this complexity and to standardize the vocabulary used to describe protein function in a human- and machine-readable manner. GO separates three simultaneous aspects of function into three hierarchies: referred to as MFO (Molecular Function Ontology), BPO (biological process ontology), and CCO, i.e. the cellular component(s) or subcellular localization(s) in which the protein acts.

Computational methods bridge the sequence-annotation gap. With the low-hanging fruits having been plugged, the experimental determination of all complete GO numbers for new proteins becomes increasingly challenging despite advances in experimental techniques. Consequently, the gap between the number of proteins with experimental GO numbers and those with known sequence and unknown function (sequence-annotation gap) remain substantial. For instance, UniRef100[4] (UniProt[5] clustered at 100% percentage pairwise sequence identity PIDE) holds roughly 220M (million) protein sequences while maximally 1M have annotations verified by experts (evidence codes EXP, IDA, IPI, IMP, IGI, IEP, TAS, or IC).

Computational biology has been bridging the sequence-annotation gap for decades[6-23]. These methods were based on two different concepts: (1) sequence similarity-based transfer (or *homology-based inference*). This builds upon the observation that proteins with similar sequence have similar function[24,25] (in more formal terms: given a query Q of unknown and a template T of known function (F_t): IF PIDE(Q,T)>threshold θ, transfer annotation F_t to Q). (2) *De-novo* methods predict protein function through machine learning[15]. If applicable, the first approach tends to out-perform the second[26-29] although it largely misses discoveries[30]. The progress of computational methods has been monitored by the CAFA endeavor (*Critical Assessment of protein Function Annotation algorithms*)[21,31-33], an international collaboration to advancing and assessing methods that bridge the sequence-annotation gap. CAFA takes place every two to three years with its fourth instance (CAFA4) currently being evaluated.

Here, we introduced a novel approach transferring annotations without sequence similarity. Instead, the method used the language model (LM) SeqVec[34] to represent protein sequences as fixed-size vectors (=embeddings) which were used to transfer annotations based on proximity in embedding rather than in sequence space. By learning to predict the next amino acid given the entire previous sequence on unlabeled data (only sequences without any type of label), SeqVec learnt to extract features describing proteins that can be input to different tasks, a step referred to as to transfer learning. Instead of transferring annotations from the labeled protein T with the highest percentage pairwise sequence identity (PIDE) to the query Q, we chose T as the protein with the smallest distance in embedding space ($DIST^{emb}$) to Q. This distance is also used as proxy for the reliability of the prediction which was used to define a reasonable cutoff above which hits are considered too distant to infer annotations. Annotations can also be inferred from more than the closest protein, i.e. the *k* closest proteins can be considered where *k* has to be optimized. In addition, we evaluated the influence of the type of LM used (SeqVec[34] vs. ProtBert[35]). Although SeqVec has never explicitly encountered GO terms during training, we

hypothesized SeqVec embeddings to implicitly encode information relevant for the transfer of annotations, i.e. capturing aspects of protein function.

Results & Discussion

Simple embedding-based transfer almost as good as CAFA3 top-10. First, we predicted GO terms for all 3,328 CAFA3 targets using the Gene Ontology Annotation (GOA) dataset *GOA2017* (Methods), removed all entries identical to CAFA3 targets (PIDE=100%; set: *GOA2017-100*) and transferred the annotations of the closest hit (k=1) in this set to the query. When applying the *NK* evaluation mode (no-knowledge available for query, Methods/CAFA3), the embedding-transfer reached F_{max} scores of 37±2% for BPO (precision=39±2%, recall=36±2%), 50±3% for MFO (precision=54±3%, recall=47±3%), and 57±2% for CCO (precision=61±3%, recall=54±3%; **Table 1**, Fig. 1, Fig. S1 in Supporting Online Material – SOM). Errors were estimated through the 95% confidence intervals (± 1.96 stderr). In the sense that the database with annotations to transfer (GOA2017) had been available before the CAFA3 submission deadline (February 2017), our predictions were directly comparable to CAFA3 [31]. This embedding-based annotation transfer clearly outperformed the two CAFA3 baselines (Fig. 1: the simple *BLAST* for sequence-based annotation transfer, and the *Naïve* method assigning GO terms statistically based on database frequencies, here *GOA2017*); it would have performed close to the top ten CAFA3 competitors (in particular for BPO: Fig. 1) had the method competed at CAFA3.

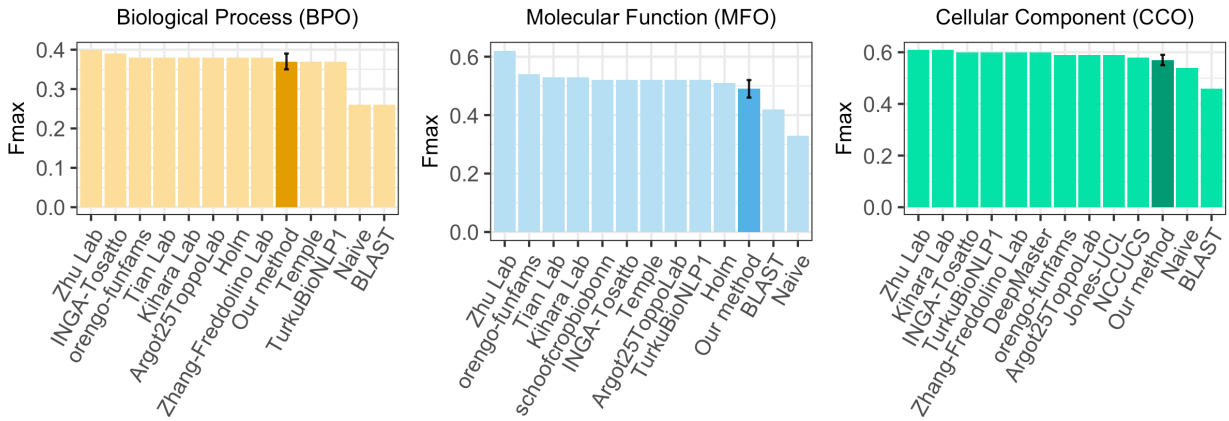

Fig. 1: F_{max} for simplest embedding-based transfer (k=1) and CAFA3 competitors. Using the datasets and conditions from CAFA3, we compared the F_{max} of the simplest implementation of the embedding-based annotation transfer, namely the greedy (k=1) solution in which the transfer comes from exactly one closest database protein (dark bar) for the three ontologies (BPO, MFO, CCO) to the top ten methods that – in contrast to our method - did compete at CAFA3 and to two background approaches "BLAST" (homology-based inference) and "Naïve" (assignment of terms based on term frequency) (lighter bars). The result shown held for the *NK* evaluation mode (no knowledge), i.e. only using proteins that were novel in the sense that they had no prior annotations. If we had developed our method before CAFA3, it would have almost reached the tenth place for MFO and CCO and ranked even slightly better for BPO. Error bars (for our method) marked the 95% confidence intervals.

Table 1: Performance for CAFA3 targets for simple GO annotation transfers.

		F_{max}		
Dataset	*Embeddings*	*BPO*	*MFO*	*CCO*
GOA2017	*SeqVec*	37±2%	50±3%	57±2%
	ProtBert	36±2%	49±3%	59±2%
	BLAST	26%	42%	46%
GOA2017X	*SeqVec*	31±2%	51±3%	56±2%
GOA2020	*SeqVec*	51±2%	61±3%	65±2%
	ProtBert	50±2%	59±2%	65±2%
	BLAST	31%	53%	58%

* Mean F_{max} values for GO term predictions using embeddings from two different language models (*SeqVec* or *ProtBert*) or sequence similarity (*BLAST*) for the datasets *GOA2017-100* (2017), *GOA2017X* (2017), and *GOA2020-100* (2020) used for annotation transfer (note: the notation '-100' implies that any entry in the dataset with PIDE=100% to any CAFA3 protein had been removed). All values were compiled for picking the single top hit (k=1) and using the CAFA3 targets from the NK and full evaluation mode [31]. For all simple annotation transfers (embedding- and sequence-based), performance was higher for the more recent data sets (GOA2020 vs. GOA2017). Error estimates are given as 95% confidence intervals. F_{max} values were computed using the CAFA3 tool [31,33].

Including more neighbors (k>1) only slightly affected F_{max} (Table S2; all average F_{max} values for k=2 to k=10 remained within the 95% confidence interval of the value for k=1). When taking all predictions into account independent of a threshold in prediction strength referred to as the reliability index (RI, Methods; i.e. any copied annotation is considered even if the confidence is low), the number of predicted GO terms increased with higher *k* (Table S3). The average number of terms annotated for each protein in *GOA2017* was already as high as 37 for BPO, 14 for MFO, and 9 for CCO. When including all predictions independent of their strength (RI) our method predicted more terms for CCO and BPO than expected from this distribution even for *k*=1. In contrast for MFO, 11.7 terms were predicted on average slightly lower than the expected average, at least for k=1 (explosion of terms for k>1: Table S3). While precision dropped with adding terms, recall increased (Table S3). To avoid overprediction and given that *k* hardly affected F_{max}, we chose k=1. Although not increasing F_{max}, it might help to consider more than the closest hit (k=1), e.g., if the closest hit only contains unspecific terms.

When applying the <u>*LK* evaluation mode</u> (limited-knowledge, i.e. query already has some annotation about function, Methods/CAFA3), the embedding-based annotation transfer reached F_{max} scores of 49±1%, 52±2%, and 58±3% for BPO, MFO, and CCO, respectively (Fig. S2). Thus, the embedding-based annotation transfer reached higher values for proteins with prior annotations (LK evaluation mode) than for novel proteins without any annotations (NK evaluation mode; **Table 1**); the same was true for the CAFA3 top-10 for which the F_{max} scores increased even more than for our method for BPO and MFO, and less for CCO (Fig. 1, Fig. S2). In the *LK* mode, predictions are evaluated for proteins for which some annotations in one or two of the ontologies existed before and additional annotations in an ontology without prior annotations were added after the CAFA3 deadline[21,31]. Supervised training uses such labels; our approach, in contrast, identified similar proteins through embedding proximity (Eqn. 4) and we had excluded all CAFA3 targets explicitly from the database used for annotation transfer (*GOA2017*). Thus, our method could not benefit from previously known annotations, i.e. for our method the *LK* and *NK* should have been identical. The observed differences were most likely explained by how F_{max} is computed. The higher F_{max} score especially for BPO for our method might be explained by a difference in the data set, e.g. the average number of BPO annotations for the LK assessment

was 19 and that for NK was 26. Other methods might have reached even higher F_{max} scores by using the known annotations for training.

Embedding-based transfer successfully identified distant relatives. Embeddings constitute learned representations of protein sequences; they are deterministic in that the same sequence produces the same embedding, i.e. if PIDE(Q,T)=100%, then $DIST^{emb}$(Q,T)=0 (Eqn. 4). Initially, we had assumed that the more similar two sequences, the more similar their embeddings, i.e. that sequence and embedding similarity correlate. However, according to the CAFA3 assessment, sequence-based annotation transfer using *BLAST* reached significantly lower F_{max} scores than embedding-based annotation transfer (**Table 1**, Fig. 1). This suggested embedding similarity to capture information different from that of sequence similarity. Explicitly calculating the correlation between sequence and embedding similarity for 431,224 sequence pairs from CAFA3/*GOA2017*-100, we observed a correlation of ρ=0.29 (Spearman's correlation coefficient, p-value < 2.2e-16; **Table 2**). Thus, sequence and embedding similarity did correlate, albeit at an unexpectedly low level. We could not conceive a convincing argument for expecting that this low correlation implies embedding-based transfer to be better or worse than sequence-based transfer. After the observation, we have now established that embedding-based transfer is, on average, better than sequence-based transfer. Given the lack of prior expectations, it appears meaningless to explain this empirical fact. Clearly, the result demonstrated that embedding similarity identified more distant relatives than sequence similarity (e.g. BLAST E-value).

Table 2: Embedding and sequence similarity correlated.

	PIDE	d_{SeqVec}	$d_{ProtBert}$
PIDE	1	0.293	0.248
d_{SeqVec}		1	0.576
$d_{ProtBert}$			1

* **PIDE**: percentage pairwise sequence identity; $\mathbf{d_{SeqVec}}$: similarity in SeqVec [34] embeddings (Eqn. 4); $\mathbf{d_{ProtBert}}$: similarity in ProtBert [35] embeddings (Eqn. 4). All values represent Spearman's correlation coefficients calculated for 434,001 sequence pairs. For all pairs, the significance was p-value<2.2e-16, i.e. significant at the level of the precision of the software R [36]. The similarity between sequence and embeddings correlated less than the two different types of embeddings, namely SeqVec and ProtBert with each other. In order to highlight the trivial symmetry of the matrix, only the upper diagonal was given.

In order to quantify how different two proteins Q and T can be in terms of their embeddings and still share GO terms, we redundancy reduced the *GOA2017* database used for annotation transfers at distance thresholds of decreasing PIDE with respect to our queries (in nine steps of 10 from 90% to 20%, data set sizes in Table S1). By construction, all proteins in the *GOA2017-100* set had PIDE<100% to all CAFA3 queries (Q). If the pair (Q,T) with the most similar embedding was also similar in sequence, embedding-based would equal sequence-based transfer. At lower PIDE thresholds, e.g. PIDE<20%, reliable annotation transfer through simple pairwise sequence alignment is no longer possible[28,37-40]. Although embeddings-based transfer tended to be slightly less successful between protein pairs with lower PIDE (Fig. 2: bars decrease toward right: more for BPO than for CCO), the drop appeared rather low and almost at all PIDE values, embedding-transfer remained above BLAST, i.e. the sequence-based transfer (Fig. 2: most bars higher than reddish line – error bars show 95% confidence intervals). The exception was for MFO at PIDE<30% and PIDE<20% for which the F_{max} scores from sequence-based transfer (BLAST) were within the 95% confidence interval (Fig. 2). This clearly showed that our approach benefited from information available through embeddings but not available from sequence, and that at least some protein pairs close in embedding and distant in sequence space might function alike.

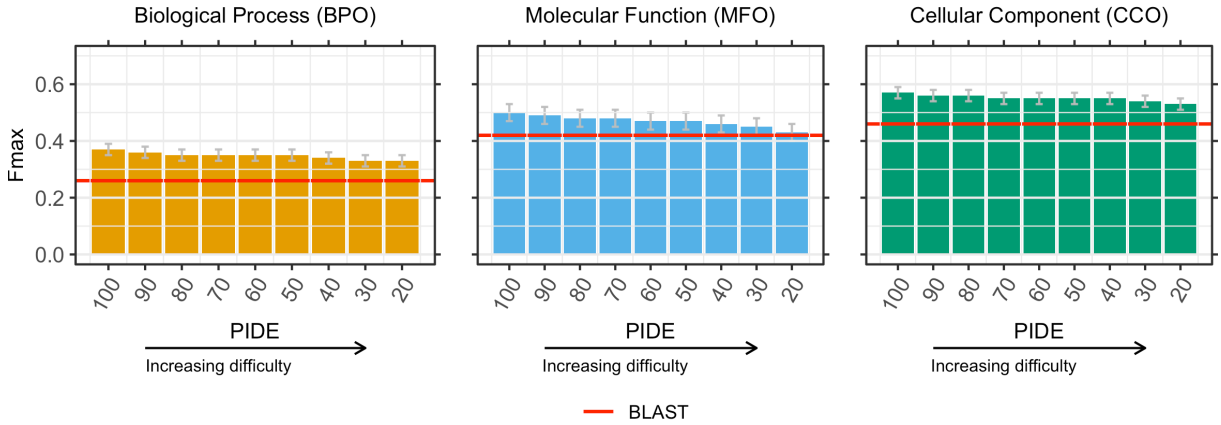

Fig. 2: Embedding-based transfer succeeded for rather diverged proteins. After establishing the low correlation between embedding- and sequence-similarity, we tested how the level of percentage pairwise sequence identity (PIDE, x-axes) between the query (protein without known GO terms) and the transfer database (proteins with known GO terms, here subsets of *GOA2017*) affected the performance of the embedding-based transfer. Technically, we achieved this by removing proteins above a certain threshold in PIDE (decreasing toward right) from the transfer database. The y-axes showed the F_{max} score as compiled by CAFA3 [31]. If embedding similarity and sequence identity correlated, our method would fall to the level of the reddish lines marked by BLAST. On the other hand, if the two were completely uncorrelated, the bars describing embedding-transfer would all have the same height (at least within the standard errors marked by the gray vertical lines at the upper end of each bar), i.e. that embedding-based transfer would be completely independent of the sequence similarity between query and template (protein of known function). The observation that all bars tended to fall toward the right implied that embedding and sequence similarity correlated (although for CCO, F_{max} remained within the 95% confidence interval of F_{max} for *GOA2017-100*). The observation that our method remained mostly above the baseline predictions demonstrated that embeddings captured important orthogonal information. Error bars indicate 95% confidence intervals.

Embedding-based transfer benefited from non-experimental annotations. Unlike the dataset made available by CAFA3, annotations in our *GOA2017* dataset were not limited to experimentally verified annotations. Instead, they included annotations inferred by computational biology, homology-based inference, or by "author statement evidence", i.e. through information from publications. Using *GOA2017X,* the subset of *GOA2017-100* containing only experimentally verified GO term annotations as transfer database, our method reached F_{max} scores of 31±2% (precision=28±2%, recall=34±2%), 51±3% (precision=53±3%, recall=49±3%), and 56±2% (precision=55±3%, recall=57±3%) for BPO, MFO, and CCO, respectively. Compared to using *GOA2017-100*, the performance dropped significantly for BPO (F_{max}=37±2% for *GOA2017-100,* **Table 1**); it decreased slightly (within 95% confidence interval) for CCO (F_{max}=57±2% for *GOA2017-100,* **Table 1**); and it increased slightly (within 95% confidence interval) for MFO (F_{max}=50±3% for *GOA2017-100*, **Table 1**). This indicated that less reliable annotations might still help annotation transfer, in particular for BPO for which many non-experimental annotations appeared helpful. Annotations for BPO may rely more on information available from publications that is not as easily quantifiable experimentally as annotations for MFO or CCO.

Many of the non-experimental annotations constituted sequence-based annotation transfers. Thus, non-experimental annotations might have helped because they constituted an implicit merger of sequence and embedding transfer. Adding homologous proteins might "bridge" sequence and embedding space by populating embedding space using annotations transferred from sequence space. The weak correlation between both spaces supported this speculation because protein pairs with very similar sequences may differ in their embeddings and *vice versa*.

Improving annotations from 2017 to 2020 increased performance significantly. For the comparison to CAFA3, we only used data available at the CAFA3 submission deadline. When applying this method to new queries, annotations will be transferred from the latest GOA. We used *GOA2020-100* (from 01/2020 removing the CAFA3 targets) to assess how the improvement of annotations from 2017 to 2020 influenced annotation transfer (**Table 1**). On *GOA2020-100,* SeqVec embedding-based transfer achieved F_{max} scores of 50±2% (precision=50±3%, recall=50±3%), 60±3% (precision=52±3%, recall=71±3%), and 65±2% (precision=57±3%, recall=75±3%) for BPO, MFO, and CCO, respectively, for the *NK* evaluation mode (**Table 1**). This constituted a substantial increase over *GOA2017-100* (Table 1). We submitted predictions for MFO for the CAFA4 targets. In the first preliminary evaluation published during ISMB2020[41], our method achieved 9th place in line with the *post facto* CAFA3 results presented here.

The large performance boost between *GOA2017* and *GOA2020* indicated the addition of many new and relevant GO annotations in three years. However, for increasingly diverged pairs (Q,T), we observed a much larger drop in F_{max} than for *GOA2017* (Fig. 2, Fig. S3). In the extreme, *GOA2020-20* (PIDE(Q,T)<20%) with F_{max} values of 33±2% (BPO), 44±2% (MFO), and 54±2% (CCO) fell to the same level as *GOA2017-20* (Fig. 2, Fig. S3). These results suggested that many of the relevant GO annotations added since 2017 were added for proteins very sequence-similar (here PIDE) to proteins for which annotations already existed in 2017. Put differently, many helpful new experiments simply refined previous computational predictions.

Running BLAST against *GOA2020-100* for sequence-based transfer (choosing the hit with the highest PIDE) showed that sequence-transfer also profited from improved annotations (difference in F_{max} values for BLAST in **Table 1**). However, while F_{max} scores for embedding-based transfer increased the most for BPO, those for sequence-based transfer increased most for MFO. Embedding-transfer still outperformed BLAST for the *GOA2020-100* set (Fig. S3C).

Even when constraining annotation transfers to more sequence-distant pairs (lower PIDE), our method outperformed BLAST against *GOA2020-100* in terms of F_{max} at least for BPO and for higher levels of PIDE in MFO/CCO (Fig. S3C). However, comparing the results for BLAST on the GOA2020-100 set with the performance of our method for subsets of very diverged sequences (e.g. PIDE<40% for GOA2020-40) under-estimated the differences between sequence- and embedding-based transfer, because the two approaches transferred annotations from different data sets. For a more realistic comparison, we re-ran BLAST only considering hits below certain PIDE thresholds (we avoided to do this for CAFA3 in order to remain directly comparable to published results). As expected, performance for BLAST dropped with decreasing sequence identity (Fig. S3 lighter bars). For instance, at PIDE<20%, F_{max} fell to 8% for BPO, 10% for MFO, and 11% for CCO (Fig. S3C lighter bars) largely due to low coverage, i.e. most queries had no hit to transfer annotations from. At this level (and for the same set), the embedding-based transfer proposed here, still achieved values of 33±2% (BPO), 44±2% (MFO), and 54±2% (CCO). Thus, our method made reasonable predictions at levels of sequence identity form which homology-based inference (BLAST) failed completely.

Embedding similarity influenced performance. Homology-based inference works best for pairs with high PIDE. Analogously, we assumed embedding-transfer to be best for pairs with high embedding similarity, i.e. low Euclidean distance (Eqn. 4). We used this to define a reliability index (RI, Eqn. 5). For the *GOA2020-100* set, the minimal RI was 0.24. The CAFA evaluation determined 0.35 for BPO, 0.28 for MFO, and 0.29 for CCO as thresholds leading to optimal performance as measured by F_{max} (Fig. dashed grey lines marked these thresholds). For all ontologies, precision and recall were almost constant for lower RIs (up to ~ 0.3). For higher RIs, precision increased, and recall decreased as expected (Fig.). While precision increased up to 82% for BPO, 91% for MFO, and 70% for CCO, it also fluctuated for high RIs (Fig.). This trend was probably caused by the low number of terms predicted at these RIs. For CCO, the RI essentially did not correlate with precision. This might point to a problem in assessing annotations

for which the non-sense Naïve method reached values of F_{max}~55% outperforming most methods. Possibly, some prediction of the type "related to cytoplasm" is all that is needed to achieve a high F_{max} in this ontology.

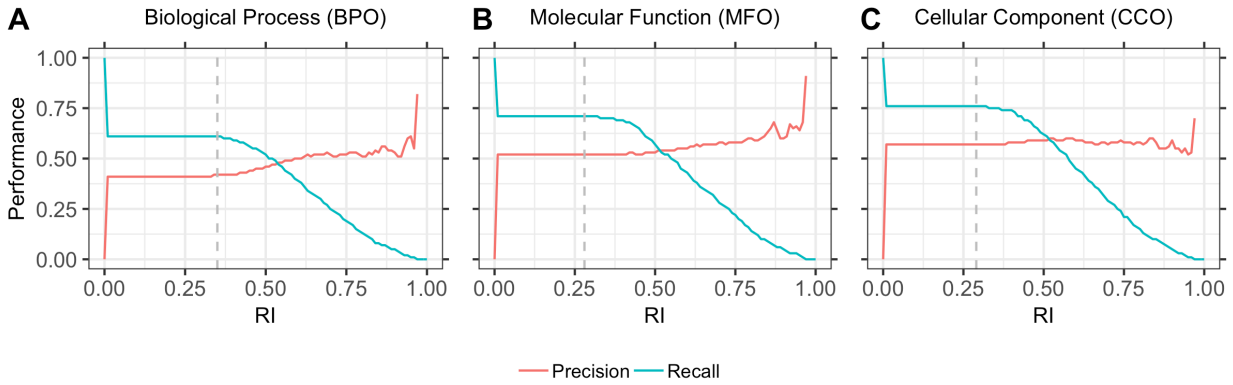

Fig. 3: Precision and recall for different reliability indices (RIs). We defined a reliability index (RI) measuring the strength of a prediction (Eqn. 5), i.e. for the embedding proximity. Precision (Eqn. 1) and recall (Eqn. 2) were almost constant for lower RIs (up to ~ 0.3) for all three ontologies (BPO, MFO, CCO). For higher RIs, precision increased while recall dropped. However, due to the low number of terms predicted at very high RIs (>0.8), precision fluctuated and was not fully correlated with RI. Dashed vertical lines marked the thresholds used by the CAFA3 tool to compute F_{max}: 0.35 for BPO, 0.28 for MFO, and 0.29 for CCO. At least for BPO and MFO higher RI values correlated with higher precision, i.e. users could use the RI to estimate how good the predictions are likely to be for their query, or to simply scan only those predictions more likely to be correct (e.g. RI>0.8).

Similar performance for different embeddings. We compared embeddings derived from two different language models (LMs). So far, we used embeddings from SeqVec[34]. Recently, ProtBert, a transformer-based approach using auto-encoding instead of auto-regression (Bert[42]) and more protein sequences (BFD[43,44]), was shown to improve secondary structure prediction[35]. Replacing SeqVec by ProtBert embeddings to transfer annotations, our approach achieved similar F_{max} scores (**Table 1**). In fact, the ProtBert F_{max} scores remained within the 95% confidence intervals of those for SeqVec (**Table 1**). Similar results were observed when using *GOA2017-100* (**Table 1**).

On the one hand, the similar performance for both embeddings might indicate that both LMs extracted equally beneficial aspects of function, irrespective of the underlying architecture (LSTMs in SeqVec, transformer encoders in ProtBert) or training set (SeqVec used UniRef50 with ~30M proteins, ProtBert used BFD with ~2.1B proteins). On the other hand, the similar F_{max} scores might also highlight that important information was lost when averaging over the entire protein to render fixed-size vectors. The similarity in F_{max} scores was less surprising given the high correlation between SeqVec and ProtBert embedding distances (ρ=0.58, p-value < 2.2e-16; **Table 2**). The two LMs correlated more with each other than either with PIDE (**Table 2**).

No gain from simple combination of embedding- and sequence-based transfer. All three approaches toward annotation transfer (embeddings from SeqVec or ProtBert, and sequence) had strengths, i.e. for some proteins sequence-transfer performed better, although on average performing worse. In fact, analyzing the pairs for which embedding-based transfer or sequence-based transfer outperformed the other method by at least four percentage points ($|F_{max}(BLAST)-F_{max}(embeddings)|\geq 4$) illustrated the expected cases for which PIDE was high and embedding similarity low, and vice versa, along with more surprising cases for which low PIDE still yielded better predictions than relatively high embedding RIs (Fig. S4). We tried to benefit from these

observations through simple combinations. Firstly, we considered all terms predicted by embeddings from either SeqVec or ProtBert. Secondly, reliability scores were combined leading to higher reliability for terms predicted in both approaches than for terms only predicted by one. None of those two approaches improved performance (Table S4, method SeqVec/ProtBert). Further simple combinations of embedding and sequence also did not improve performance (Table S4, method SeqVec/ProtBert/BLAST). Finding out whether or not more advanced combinations were beyond the scope of this work.

Case study: embedding-based annotation transfer for SARS-CoV-2 proteome. Due to its simplicity and speed, embedding-based annotation transfer can easily be applied to novel proteins to shed light on their potential functionality. Given the relevance of SARS-CoV-2, the virus causing COVID-19, we applied our method to predict GO terms (BPO, MFO, and CCO) for all 14 SARS-CoV-2 proteins (taken from UniProt[5]; all raw predictions were made available as additional files named predictions_$emb_$ont.txt replacing the variables $emb and $ont as follows: $emb=seqvec|protbert, and $ont=bpo|mfo|cco).

Step 1: confirmation of known annotations. Out of the 42 predictions (14 proteins in 3 ontologies), 12 were based on annotation transfers using proteins from the human SARS coronavirus (SARS-CoV), and 13 on proteins from the bat coronavirus HKU3 (BtCoV). CCO predictions appeared reasonable with predicted locations mainly associated with the virus (e.g. viral envelope, virion membrane) or the host (e.g. host cell Golgi apparatus, host cell membrane). Similarly, MFO predictions often matched well-known annotations, e.g. the replicase polyproteins 1a and 1ab were predicted to be involved in RNA-binding as confirmed by UniProt. In fact, annotations in BPO were known for 7 proteins (in total 40 GO terms), in MFO for 6 proteins (30 GO terms), and in CCO for 12 proteins (68 GO terms). Only three of these annotations were experimentally verified. With our method, we predicted 25 out of the 30 GO terms for BPO (83%), 14/30 for MFO (47%), and 59/68 for CCO (87%). Even more annotations were similar to the known GO annotations but were more or less specific (Table S4 summarized all predicted and annotated GO leaf terms, the corresponding names can be found in the additional files predictions_$emb_$ont.txt).

Step 2: new predictions. Since the GO term predictions matched well-characterized proteins, predictions might provide insights into the function of proteins without or with fewer annotations. For example, function and structure of the non-structural protein 7b (Uniprot identifier P0DTD8) are not known except for a transmembrane region of which the existence was supported by the predicted CCO annotation "*integral component of the membrane*" and "*host cell membrane*". This CCO annotation was also correctly predicted by the embedding-based transfer from an *Ashbya gossypii* protein. Additionally, we predicted "*transport of virus in host*, *cell to cell*" for BPO and for MFO "*proton transmembrane transporter activity*" for MFO. This suggested non-structural protein 7b to play a role in transporting the virion through the membrane into the host cell. Visualizing the leaf term predictions in the GO hierarchy could help to better understand very specific annotations. For the BPO annotation of the non-structural protein 7b, the tree revealed that this functionality constituted two major aspects: The interaction with the host and the actual transport to the host (Fig. S5). To visualize the predicted terms in the GO hierarchy, for example the tool NaviGO[45] can be used which can help to interpret the GO predictions given for the SARS-CoV-2 proteins here.

Comparing annotation transfers based on embeddings from SeqVec and from ProtBert showed that 16 of the 42 predictions agreed for the two different language models (LMs). For five predictions, one of the two LMs yielded more specific annotations, e.g., for the nucleoprotein (Uniprot identifier P0DTC9) which is involved in viral genome packaging and regulation of viral transcription and replication. For this protein SeqVec embeddings found no meaningful result, while ProtBert embeddings predicted terms such as "*RNA polymerase II preinitiation complex*" and "*positive regulation of transcription by RNA polymerase II*" fitting to the known function of the

nucleoprotein. This example demonstrated how the combination of results from predictions using different LMs may refine GO term predictions.

Conclusions

We introduced a new concept for the prediction of GO terms, namely the annotation transfer based on similarity of embeddings obtained from deep learning language models (LMs). This approach essentially replaced sequence information by complex embeddings that capture some non-local information; it resulted in a very simple novel prediction method complementing homology-based inference. Despite its simplicity, this new method would have reached the top ten, had it participated at CAFA3 (Fig. 1); it outperformed by several margins of statistically significance homology-based inference ("BLAST") with F_{max} values of BPO **+11**±2% (F_{max}(embedding)-F_{max}(sequence)), MFO **+8**±3%, and CCO **+11**±2% (**Table 1**, Fig. 1). Embedding-based transfer remained above the average for sequence-based transfer even for protein pairs with PIDE<20% (Fig. 2). In other words, embedding similarity still worked for proteins that diverged beyond the recognition in pairwise alignments (Figs. S3 & S4). Embedding-based transfer not only outperformed homology-based inference, it is also blazingly fast to compute, i.e. around 0.1s per protein. The only time-consuming step was computing embeddings for all proteins in the lookup database which needs to be done only once; it took about 30min for the entire human proteome. Better GO annotations added from 2017 to 2020 improved both sequence- and embedding-based annotation transfer significantly (**Table 1**). Another aspect of the simplicity was that, at least in the context of the CAFA3 evaluation, the choice of none of the two free parameters really mattered: embeddings from both LMs tested performed, on average, equally, and the number of best hits (k-nearest neighbors) did not matter much (Table S2). The power of this new concept was generated by the degree to which embeddings implicitly capture important information relevant for protein structure and function prediction. Ultimately, the reason for this was in that the correlation between embeddings and sequence remained limited (**Table 2**).

Methods

Generating embedding space. The embedding-based annotation transfer introduced here required each protein to be represented by a fixed-length vector, i.e. a vector with the same dimension for a protein of 30 and another of 40000 residues (maximal sequence length for ProtBert). To this end, we used SeqVec[34] to represent each protein in our dataset by a fixed size embedding. SeqVec is based on ELMo[46] using a stack of LSTMs[47] for auto-regressive pre-training, i.e. predicting the next token (originally a word in a sentence, here an amino acid in a protein sequence), given all previous tokens. Two independent stacks of LSTMs process the sequence from both directions to capture bi-directional context. For SeqVec, three layers, i.e. one uncontextualized CharCNN[48] and two bi-directional LSTMs, were trained on each protein in UniRef50 (UniProt[5] clustered at 50% PIDE). As no labeled data was used (self-supervised training), the embeddings could not capture any explicit information such as GO numbers. Thus, SeqVec does not need to be retrained for subsequent prediction tasks using the embeddings as input. After pre-training, the hidden states of the model were used to extract features. For SeqVec, each layer and each direction outputs a 512-dimensional vector; forward and backward passes for the first LSTM layer were extracted and concatenated into a matrix of size L * 1024

for a protein with L residues. A fixed-size representation was then derived by averaging over the length dimension, resulting in a vector of size 1024 for each protein (Fig. S6).

To evaluate the effect of using different LMs to generate the embeddings, we also used a transformer-based LM trained on protein sequences (ProtBert-BFD[35], here simply referred to as *ProtBert*). ProtBert is based on the LM BERT[49] (Bidirectional Encoder Representations from Transformers[42]) which processes sequential data through the self-attention mechanism [50]. Self-attention compares all tokens in a sequence to all others in parallel, thereby capturing long-range dependencies better than LSTMs. BERT also replaced ELMo's auto-regressive by an auto-encoding pre-training, i.e. reconstructing corrupted tokens from the input, which enables to capture bi-directional context. ProtBert was trained on 2.1B protein sequences (BFD)[43,44] which is 70 times larger than UniRef50. The output of the last attention layer of ProtBert was used to derive a 1024-dimensional embedding for each residue. As for SeqVec, the resulting L * 1024 matrix was pooled by averaging over protein length providing a fixed-size vector of dimension 1024 for each protein.

Generating the embeddings for all human proteins using both SeqVec and ProtBert allowed estimating the time required for the generation of the input to our new method. Using a single Nvidia P100 with 16GB vRAM and dynamic batching (depending on the sequence length), this took, on average, about 0.1s per protein[35].

Data set. To create a database for annotation transfer, we extracted protein sequences with annotated GO terms from the Gene Ontology Annotation (GOA) database[51-53] (containing 29,904,266 proteins from UniProtKB[5] in February 2020). In order to focus on proteins known to exist, we only extracted records from Swiss-Prot[54]. Proteins annotated only at the ontology roots, i.e. proteins limited to “GO:0003674” (molecular_function), “GO:0008150” (biological_process), or “GO:0005575” (cellular_component) were considered meaningless and were excluded. The final dataset *GOA2020* contained 295,558 proteins (with unique identifiers, IDs) described by 31,485 different GO terms. The GO term annotation for each protein includes all annotated terms and all their parent terms. Thereby, proteins are, on average, annotated by 37 terms in BPO, 14 in MFO, and 9 in CCO. Counting only leaves brought the averages to 3 in BPO, 2 in MFO, and 3 in CCO.

For comparison to methods that contributed to CAFA3[31], we added another dataset *GOA2017* using the GOA version available at the submission deadline of CAFA3 (Jan 17, 2017). After processing (as for *GOA2020*), *GOA2017* contained 307,287 proteins (unique IDs) described by 30,124 different GO terms. While we could not find a definite explanation for having fewer proteins in the newer database (*GOA2020* 295K proteins vs. *GOA2017* with 307K), we assumed that it originated from major changes in GO including the removal of obsolete and inaccurate annotations and the refactoring of MFO[3].

The above filters neither excluded GOA annotations inferred from phylogenetic evidence and author statements nor those based on computational analysis. We constructed an additional data set, *GOA2017X* exclusively containing proteins annotated in Swiss-Prot as experimental (evidence codes EXP, IDA, IPI, IMP, IGI, IEP, TAS, or IC) following the CAFA3 definition[31]. We further excluded all entries with PIDE=100% to any CAFA3 target bringing *GOA2017X* to 303,984 proteins with 28,677 different GO terms.

Performance evaluation. The targets from the CAFA3 challenge[31] were used to evaluate the performance of our new method. This set consisted of 3,328 proteins with the following subsets with experimental annotations in each sub-hierarchy of GO: BPO 2,145, MFO 1,101, and CCO 1,097 (more details about the dataset are given in the original CAFA3 publication[31]).

In order to expand the comparison of the transfer based on sequence- and embedding similarity, we also reduced the redundancy through applying CD-HIT and PSI-CD-HIT[55,56] to the *GOA2020* and *GOA2017* sets against the evaluation set at thresholds θ of PIDE=100, 90, 80, 70, 60, 50, 40, 30 and 20% (Table S1 in the Supporting Online Material (SOM) for more details about these nine subsets).

We evaluated our method against two baseline methods used at CAFA3, namely *Naïve* and *BLAST*, as well as, against CAFA3's top ten[31]. We computed standard performance measures. True positives (TP) were GO terms predicted above a certain reliability (RI) threshold (Method below), false positives (FP) were GO terms predicted but not annotated, and false negatives (FN) were GO terms annotated but not predicted. Based on these three numbers, we calculated precision (Eqn. 1), recall (Eqn. 2), and F1 score (Eqn. 3) as follows.

$$Precision = \frac{TP}{TP+FP} \quad \text{(Eqn. 1)}$$

$$Recall = \frac{TP}{TP+FN} \quad \text{(Eqn. 2)}$$

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision+Recall} \quad \text{(Eqn. 3)}$$

The F_{max} value denoted the maximum F1 score achievable for any threshold in reliability (RI, Eqn. 5). This implies that the assessment fixes the optimal value rather than the method providing this value. Although this arguably over-estimates performance, it has evolved to a quasi-standard of CAFA; the publicly available CAFA Assessment Tool[31,33] calculated F_{max} for the CAFA3 targets in the same manner as for the official CAFA3 evaluation. If not stated otherwise, we reported precision and recall values for the threshold leading to F_{max}.

CAFA3 assessed performance separately for two sets of proteins for all three ontologies: (i) proteins for which no experimental annotations were known beforehand (no-knowledge, NK evaluation mode) and (ii) proteins with some experimental annotations in one or two of the other ontologies (limited-knowledge, LK evaluation mode)[21,31]. We also considered these sets separately in our assessment. CAFA3 further distinguished between *full* and *partial evaluation* with *full evaluation* penalizing if no prediction was made for a certain protein, and *partial evaluation* restricting the assessment to the subset of proteins with predictions[31]. Our method predicted for every protein; thus, we considered only the *full evaluation*. Also following CAFA3, symmetric 95% confidence intervals were calculated as error estimates assuming a normal distribution and 10,000 bootstrap samples estimated mean and standard deviation.

Method: annotation transfer through embedding similarity. For a given query protein Q, GO terms were transferred from proteins with known GO terms (sets *GOA2020* and *GOA2017*) through an approach similar to the k-nearest neighbor algorithm (k-NN)[57]. For the query Q and for all proteins in, e.g. *GOA2020*, the SeqVec[34] embeddings were computed. Based on the Euclidean distance between two embeddings *n* and *m* (Eqn. 4), we extracted the *k* closest hits to the query from the database where *k* constituted a free parameter to optimize.

$$d(n,m) = \sqrt{\sum_{i=1}^{1024}(n_i - m_i)^2} \quad \text{(Eqn. 4)}$$

In contrast to standard k-NN algorithms, all annotations from all hits were transferred to the query instead of only the most frequent one[57]. When multiple pairs reached the same distance, all were considered, i.e. for a given *k*, more than *k* proteins might be considered for the GO term prediction. The calculation of the pairwise Euclidean distances between queries and all database proteins and the subsequent nearest neighbor extraction was accomplished very efficiently. For instance, the nearest-neighbor search of 1,000 query proteins against GOA20* with about

300,000 proteins took on average only about 0.005s per query on a single i7-6700 CPU, i.e. less than two minutes for all human proteins.

Converting the Euclidean distance enabled to introduce a reliability index (RI) ranging from 0 (weak prediction) to 1 (confident prediction) for each predicted GO term *p* as follows:

$$RI(p) = \frac{1}{k}\sum_{i=1}^{l}\frac{0.5}{0.5+d(q,n_i)} \quad \text{(Eqn. 5)}$$

with *k* as the overall number of hits/neighbors, *l* as the number of hits annotated with the GO term *p* and the distance $d(q,n_i)$ between query and hit being calculated according to Eqn. 4.

Proteins represented by an embedding identical to the query protein (d=0) led to RI=1. Since the RI also takes into account, how many proteins *l* in a list of *k* hits are annotated with a certain term *p* (Eqn. 5), predicted terms annotated to more proteins (larger *l*) have a higher RI than terms annotated to fewer proteins (smaller *l*). As this approach accounts for the agreement of the annotations between the k hits, it requires the RI to be normalized by the number of considered neighbors *k*, making it not directly comparable for predictions based on different values for *k*.

Availability. GO term predictions using embedding similarity for a certain protein sequence can be performed through our publicly available webserver: https://embed.protein.properties/. The source code along with all embeddings for *GOA2020* and *GOA2017*, and the CAFA3 targets are also available on GitHub: https://github.com/Rostlab/goPredSim (more details in the repository).

Acknowledgements

Thanks to Tim Karl and Inga Weise (both TUM) for invaluable help with technical and administrative aspects of this work. Thanks to the organizers and participants of the CAFA challenge for their crucial contributions, in particular to Predrag Radivojac (Northeastern Boston), Iddo Friedberg (Iowa State Ames), Sean Mooney (U of Washington Seattle), Casey Green (U Penn Philadelphia, USA), Mark Wass (Kent, England), and Kimberly Reynolds (U Texas Dallas). Thanks to the maintainers of the Gene Ontology (in particular to Michael Ashburner - Cambridge and Suzanna Lewis – Berkeley), as well as, the Gene Ontology Annotation database (maintained by the team of Claire O'Donovan at the EBI) for their standardized vocabulary and curated data sets. Last, but not least, thanks to all maintainers of public databases and to all experimentalists who enabled this analysis by making their data publicly available. This work was supported by the Bavarian Ministry of Education through funding to the TUM, by a grant from the Alexander von Humboldt foundation through the German Ministry for Research and Education (BMBF: Bundesministerium für Bildung und Forschung) as well as by a grant from Deutsche Forschungsgemeinschaft (DFG–GZ: RO1320/4–1). We gratefully acknowledge the support of NVIDIA Corporation with the donation of one Titan GPU used for this research.

References

1 Krebs, H. A. & Johnson, W. A. Metabolism of ketonic acids in animal tissues. *Biochem J* **31**, 645-660, doi:10.1042/bj0310645 (1937).

2 Ashburner, M. *et al.* Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet* **25**, 25-29, doi:10.1038/75556 (2000).

3 The Gene Ontology, C. The Gene Ontology Resource: 20 years and still GOing strong. *Nucleic Acids Res* **47**, D330-D338, doi:10.1093/nar/gky1055 (2019).

4 Suzek, B. E., Huang, H., McGarvey, P., Mazumder, R. & Wu, C. H. UniRef: comprehensive and non-redundant UniProt reference clusters. *Bioinformatics* **23**, 1282-1288, doi:10.1093/bioinformatics/btm098 (2007).

5 UniProt Consortium. UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Res* **47**, D506-D515, doi:10.1093/nar/gky1049 (2019).

6 Koonin, E. V., Bork, P. & Sander, C. Yeast chromosome III: new gene functions. *EMBO Journal* **13**, 493-503 (1994).

7 Nishikawa, K. & Ooi, T. Correlation of the amino acid composition of a protein to its structural and biological characteristics. *Journal of Biochemistry* **91**, 1821-1824 (1982).

8 Klein, P., Kanehisa, M. & DeLisi, C. Prediction of protein function from sequence properties: discriminant analysis of a data base. *Biochim. Biophys. Acta* **787**, 221-226 (1984).

9 Hirst, J. D. & Sternberg, M. J. E. Prediction of structural and functional features of protein and nucleic acid sequences by artificial neural networks. *Biochemistry* **31**, 615-623 (1992).

10 Gaasterland, T. & Ragan, M. A. Microbial genescapes: phyletic and functional patterns of ORF distribution among prokaryotes. *Microb Comp Genomics* **3**, 199-217 (1998).

11 Mandel-Gutfreund, Y. & Margalit, H. Quantitative parameters for amino acid-base interaction: implications for prediction of protein-DNA binding sites. *Nucleic Acids Res* **26**, 2306-2312, doi:gkb396 [pii] (1998).

12 Andrade, M. A., O'Donoghue, S. I. & Rost, B. Adaptation of protein surfaces to subcellular location. *Journal of Molecular Biology* **276**, 517-525 (1998).

13 Cokol, M., Nair, R. & Rost, B. Finding nuclear localisation signals. *EMBO Reports* **1**, 411-415 (2000).

14 Nair, R. & Rost, B. Inferring sub-cellular localisation through automated lexical analysis. *Bioinformatics* **18**, S78-S86 (2002).

15 Rost, B., Liu, J., Nair, R., Wrzeszczynski, K. O. & Ofran, Y. Automatic prediction of protein function. *Cellular and Molecular Life Sciences* **60**, 2637-2650 (2003).

16 Leslie, C., Eskin, E., Weston, J. & Noble, W. S. Mismatch string kernels for SVM protein classification. *Bioinformatics*, in press (2003).

17 Ofran, Y., Punta, M., Schneider, R. & Rost, B. Beyond annotation transfer by homology: novel protein-function prediction methods to assist drug discovery. *Drug Discovery Today* **10**, 1475-1482 (2005).

18 Barutcuoglu, Z., Schapire, R. E. & Troyanskaya, O. G. Hierarchical multi-label prediction of gene function. *Bioinformatics* **22**, 830-836, doi:10.1093/bioinformatics/btk048 (2006).

19 Mostafavi, S., Ray, D., Warde-Farley, D., Grouios, C. & Morris, Q. GeneMANIA: a real-time multiple association network integration algorithm for predicting gene function. *Genome Biol* **9 Suppl 1**, S4, doi:10.1186/gb-2008-9-s1-s4 (2008).

20 Hamp, T. *et al.* Homology-based inference sets the bar high for protein function prediction. *BMC Bioinformatics* **14 Suppl 3**, S7, doi:10.1186/1471-2105-14-S3-S7 (2013).

21 Radivojac, P. *et al.* A large-scale evaluation of computational protein function prediction. *Nat Methods* **10**, 221-227, doi:10.1038/nmeth.2340 (2013).

22 Cozzetto, D., Minneci, F., Currant, H. & Jones, D. T. FFPred 3: feature-based function prediction for all Gene Ontology domains. *Sci Rep* **6**, 31865, doi:10.1038/srep31865 (2016).

23 Kulmanov, M. & Hoehndorf, R. DeepGOPlus: improved protein function prediction from sequence. *Bioinformatics* **36**, 422-429, doi:10.1093/bioinformatics/btz595 (2020).

24 Zuckerkandl, E. Evolutionary processes and evolutionary noise at the molecular level. *J. Mol. Evol.* **7**, 269-311 (1976).

25 Dayhoff, M. O. *Atlas of protein sequence and structure*. Vol. 4, Suppl. 3 (National Biomedical Research Foundation, 1978).

26 Nakai, K. & Horton, P. PSORT: a program for detecting sorting signals in proteins and predicting their subcellular localization. *Trends Biochem Sci* **24**, 34-36 (1999).

27 Goldberg, T. *et al.* LocTree3 prediction of localization. *Nucleic Acids Res* **42**, W350-355, doi:10.1093/nar/gku396 (2014).

28 Nair, R. & Rost, B. Sequence conserved for sub-cellular localization. *Protein Science* **11**, 2836-2847 (2002).

29 Qiu, J. *et al.* ProNA2020 predicts protein-DNA, protein-RNA, and protein-protein binding proteins and residues from sequence. *J Mol Biol* **432**, 2428-2443, doi:10.1016/j.jmb.2020.02.026 (2020).

30 Goldberg, T., Rost, B. & Bromberg, Y. Computational prediction shines light on type III secretion origins. *Sci Rep* **6**, 34516, doi:10.1038/srep34516 (2016).

31 Zhou, N. *et al.* The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. *Genome Biol* **20**, 244, doi:10.1186/s13059-019-1835-8 (2019).

32 Friedberg, I. & Radivojac, P. Community-Wide Evaluation of Computational Function Prediction. *Methods Mol Biol* **1446**, 133-146, doi:10.1007/978-1-4939-3743-1_10 (2017).

33 Jiang, Y. *et al.* An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome Biol* **17**, 184, doi:10.1186/s13059-016-1037-6 (2016).

34 Heinzinger, M. *et al.* Modeling aspects of the language of life through transfer-learning protein sequences. *BMC Bioinformatics* **20**, 723, doi:10.1186/s12859-019-3220-8 (2019).

35 Elnaggar, A. *et al.* ProtTrans: towards cracking the language of life's code through self-supervised deep learning and high performance computing. *bioRxiv* (2020).

36 R Core Team. (R Foundation for Statistical Computing, 2017).

37 Rost, B. Twilight zone of protein sequence alignments. *Protein Engineering* **12**, 85-94 (1999).

38 Rost, B. Enzyme function less conserved than anticipated. *Journal of Molecular Biology* **318**, 595-608 (2002).

39 Mika, S. & Rost, B. Protein–protein interactions more conserved within species than across species. *PLoS Computational Biology* **2**, e79, doi:10.1371/journal.pcbi.0020079 (2006).

40 Clark, W. T. & Radivojac, P. Analysis of protein function and its prediction from amino acid sequence. *Proteins* **79**, 2086-2096, doi:10.1002/prot.23029 (2011).

41 El-Mabrouk, N. & Slonim, D. K. ISMB 2020 proceedings. *Bioinformatics* **36**, i1-i2, doi:10.1093/bioinformatics/btaa537 (2020).

42 Vaswani, A. *et al.* in *Neural Information Processing Systems Conference.* (eds I Guyon *et al.*) 5998-6008 (Curran Associates, Inc.).

43 Steinegger, M., Mirdita, M. & Soding, J. Protein-level assembly increases protein sequence recovery from metagenomic samples manyfold. *Nat Methods* **16**, 603-606, doi:10.1038/s41592-019-0437-4 (2019).

44 Steinegger, M. & Soding, J. Clustering huge protein sequence sets in linear time. *Nat Commun* **9**, 2542, doi:10.1038/s41467-018-04964-5 (2018).

45 Ding, Z., Wei, Q. & Kihara, D. *NaviGO - An analytic tool for Gene Ontology Visualization and Similarity*, <http://kiharalab.org/web/navigo/views/goparent.php> (2020).
46 Peters, M. E. *et al.* Deep contextualized word representations. *arXiv preprint arXiv:180205365* (2018).
47 Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural Comput* **9**, 1735-1780, doi:10.1162/neco.1997.9.8.1735 (1997).
48 Kim, Y., Jernite, Y., Sontag, D. & Rush, A. M. Character-Aware Neural Language Models. *arXiv preprint arXiv:150806615* (2015).
49 Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* (2018).
50 Bahdanau, D., Cho, K. & Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv* (2014).
51 Camon, E. *et al.* The Gene Ontology Annotation (GOA) Database: sharing knowledge in Uniprot with Gene Ontology. *Nucleic Acids Res* **32**, D262-266, doi:10.1093/nar/gkh021 (2004).
52 Huntley, R. P. *et al.* The GOA database: gene Ontology annotation updates for 2015. *Nucleic Acids Res* **43**, D1057-1063, doi:10.1093/nar/gku1113 (2015).
53 *GOA*, <http://www.ebi.ac.uk/GOA> (2020).
54 Boutet, E., Lieberherr, D., Tognolli, M., Schneider, M. & Bairoch, A. UniProtKB/Swiss-Prot. *Methods Mol Biol* **406**, 89-112, doi:10.1007/978-1-59745-535-0_4 (2007).
55 Li, W. & Godzik, A. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics* **22**, 1658-1659, doi:10.1093/bioinformatics/btl158 (2006).
56 Fu, L., Niu, B., Zhu, Z., Wu, S. & Li, W. CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics* **28**, 3150-3152, doi:10.1093/bioinformatics/bts565 (2012).
57 Cover, T. & Hart, P. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory* **13**, 21-27 (1967).