nature portfolio

https://doi.org/10.1038/s42256-022-00457-9

Supplementary information

Learning functional properties of proteins with language models

Supplementary Information

Learning Functional Properties of Proteins with Language Models

Serbulent Unsal[1,2], Heval Ataş[1], Muammer Albayrak[2], Kemal Turhan[2], Aybar C. Acar[1], Tunca Doğan[1,3,4*]

[1]Cancer Systems Biology Laboratory (KanSiL), Graduate School of Informatics, Middle East Technical University, Ankara, Turkey

[2]Department of Biostatistics and Computer Sciences, Karadeniz Technical University, Trabzon, Turkey

[3]Department of Computer Engineering, Hacettepe University, Ankara, Turkey

[4]Institute of Informatics, Hacettepe University, Ankara, Turkey

*To whom correspondence should be addressed, e-mail: tuncadogan@gmail.com

1. Different Approaches for Representing Proteins

Feature vectors should ideally represent relevant properties (e.g., physical, chemical, or biological properties of proteins). For example, within the classical protein representation approach, a protein can be represented as a two-dimensional numeric vector where the first dimension corresponds to the mean hydrophobicity value, and the second is the mean net charge[1]. Using these vectors as input, a classifier can be trained (Fig. S1a). In another example shown in Fig. S1b, the representation learning approach is employed, where Gene Ontology (GO) based functional annotations of proteins are used as input data. An initial binary matrix that displays the associations between proteins and GO terms is decomposed into latent protein and GO term matrices using matrix factorization. Afterwards, a predicted protein vs. The GO term matrix is calculated with the dot product of the former, and the transpose of the latter matrix. The error between the original and predicted matrices is used to update the parameters of the model during training. When the training is finished, each row of the finalized latent matrices is used as a feature vector that represents the respective protein or GO term and can be used as input to other classification or clustering models for various predictive tasks.

2. Classical Protein Representations

In classical protein representations, fixed sized numerical feature vectors are generated by applying predefined rules (statistical calculations in some cases) as a means of data transformation on previously known quantitative measurements of selected physical, chemical and/or biological properties either at the level of individual amino acids, or sub-sequence fragments. Classical methods are model-driven, as these predefined rules are determined by the expert according to the known properties of biomolecular systems, as opposed to the data-driven approach used in learned representations, where the data at hand is directly utilized for extracting information in an automated fashion.

Physicochemical properties of amino acids are widely utilized in early classical protein representation approaches, as they can be easily obtained using the sequence, and they correlate well with the structural and functional properties of proteins. To the best of our knowledge, one of the first studies in which proteins are represented as vectors was conducted by Klein *et al.*[2]. In this study, proteins are represented using the number of hydrophilic and hydrophobic residues, to detect and classify membrane-spanning proteins. In another study, proteins are represented using an 18-dimensional vector, which is calculated based on hydrophobicity/hydrophilicity and appearance of short characteristic amino acid patterns such as signal peptides[3], in which functions of proteins are predicted with high accuracy based on 26 different groups.

One of the first applications of using high throughput data for representing proteins is by Liao and Noble[4], where the authors attempted to detect the structural and taxonomic relations between proteins using support vector machines (SVM). They represent proteins with fixed vectors of real numbers composed of pairwise sequence similarity scores against a corpus of proteins. The algorithm considers both positive and negative samples in the vectors to represent proteins.

Chen *et al.* proposed iFeature[5], a Python package for constructing structural and physicochemical feature-based protein feature vectors, solely using sequences as input. The tool incorporates 53 different representation methods including amino acid composition, sequence order, secondary structure and protein disorder-based approaches.

Another widely used input to construct protein representations are the evolutionary relationships. It is possible to identify sites/regions with critical importance (e.g., functional regions) using evolutionary conservation between homologs. In their respective study, Wang *et al.* proposed a tool, POSSUM (both as an online service and a stand-alone version), for constructing protein feature vectors from 21 different position specific scoring matrix (PSSM) based classical protein representation methods, where the initial input is the protein sequence. The resulting vectors of 21 PSSM-based methods are evaluated on the prediction of type IV secretion effectors[6]

A variety of biological data types are employed to represent proteins using the classical approach. Here, we grouped the top-performing protein function prediction methods in the CAFA2 challenge[7], in terms of their input type, together with the data transformation approach they utilized for constructing their quantitative vectors (supplementary material Table S1).

3. Protein Representation Learning

Protein representation learning methods collect data from one or more resources (e.g., sequences, interactions, etc.) and employ either supervised or unsupervised learning to train a model, which output the representation vector to be used in other protein informatics related applications. Supervised learning requires labelled data, for example gene/protein entries that are annotated with biomolecular functional definitions such as GO terms. This type of training data can be produced by experiments and manual curation, and since this has a high cost, only a small percentage of biomolecules are labelled. On the other hand, unsupervised learning does not need labelling, making it easily applicable to many types of biomedical data. However, unsupervised methods generally require larger training datasets and additional computational power, especially when deep learning-based methods are used (e.g., GPT-3 which is a state-of-the-art language model trained with 300 billion tokens and costs 3.14 x 10^{23} floating point operations[8]). Unsupervised methods can be further divided into local and global approaches[9]. Methods in the former group construct representations based on the local context (e.g., in a language model, words surrounding the word of interest in a text), whereas in the latter, the sample is evaluated in terms of a larger, global context (e.g., the whole paragraph or document, the word of interest belongs to).

In the domain of natural language processing (NLP), one of the first contemporary word representation learning methods, word2vec, was developed by Mikolov *et al.*[10] . Word2Vec is an unsupervised learning network that calculates a vector representation for each word in a text. In word2vec-like NLP models, the learning process is based on the co-occurrence of words. During the training of a word2vec model with the skip-gram architecture, the vector that represents the current word is optimized based on the correct prediction of surrounding words. In a successful representation model, e.g., the words “protein” and “gene” will be proximally located in the feature hyperspace (i.e., they will be semantically related to each other), since they are frequently observed together. Additionally, the model may also discover relationships between these words transitively. Word2vec is an example of a shallow data representation, in which only one effective data processing layer (i.e., the hidden layer in an artificial neural network) is presented. Word2vec laid the foundation of many widely-used data representation learning methods available today, including protein representations. Later, deeper models, in which there are more than one effective data processing layer[11], were developed and have achieved far better performance on NLP tasks[12].

The first examples of learned protein representations were based on the word2vec algorithm[Citation error][,13] [14,15,16,17,18,19] , most of which are still in use today. Since word2vec depends on word co-occurrence in a limited window, it ignores the larger context which may include critical semantic information. For protein sequence-based representations, this larger context can be the whole protein sequence. Another embedding method, doc2vec[20] includes the whole context to some extent and performs better than word2vec on selected tasks. Several methods use doc2vec to represent proteins[21–27]. Also, deep language models, such as BERProtT5-XL[12] and ELMO[28] were originally developed for NLP, and later employed for protein representations[29,30]. Furthermore, Convolutional Neural networks (CNNs), having the ability to learn to summarize the data with adaptive filters, have been employed to represent proteins[29,31–34]. Additionally, architectures that are capable of inferring patterns from sequential data (e.g., protein sequences) using the attention mechanism[29,35], such as Long Short-Term Memory (LSTM) neural networks[29,30,36–38] and transformer based algorithms [39], are used in representation methods. However, transformer-based methods have shortcomings considering model explainability[40,41]. For this, Restricted Boltzmann Machines (RBM)[42,43] with their self-recursive design are used to construct explainable protein representation models[44]. Finally, hybrid approaches are utilized in the protein representation learning literature[34,45,46]. Furthermore, generative models, which are capable of learning latent space representations, are used in protein representation learning. To provide a few examples, different architectures e.g., variational autoencoders, generative adversarial networks, and etc., are employed to predict the effects of mutations[47], fitness landscape and stability of proteins[48] , and cohort frequency estimation for T-Cell Receptors[49]. In most cases, these models are not designed with the purpose of producing reusable protein representation vectors, so we could not include them in our benchmark analysis.

4. Protein Representation Methods Benchmarked in This Study

In this study, we evaluate the performance of learned representations in comparison to various baseline and state-of-the-art methods, to assess the value added by these novel approaches. From a practical point of view, methods included in our benchmarks can be grouped under 4 main classes:

1. curated rule/association-based systems,
2. classical methods for protein representation and annotation,
3. small-scale models in protein representation learning, and
4. large-scale models in protein representation learning.

The curated rule/association-based systems we tested in our study include UniRule2GO[50] , InterPro2GO[51], and Ensembl-Orthology[52]. These systems provide ready-to-use annotations (e.g., protein - GO term mappings), along with their respective rules, instead of protein representation vectors to be used in machine learning models. Once the curator manually defines an association rule, it is applied to all uncharacterized protein sequences in the database, and the respective annotation is recorded for the protein entries that satisfy the rule condition(s). UniRule2GO[50] mappings are based on the Unified Rule (UniRule) system, in which annotation rules are created by curators. Rules have conditions such as the existence of certain domains, motifs, signatures or annotations for the protein; and upon meeting the listed conditions, the associated GO terms are mapped to the respective protein entry. The InterPro[51] database integrates multiple protein family and domain databases, and merges/organizes their annotations under their own system. InterPro2GO[51] mappings (i.e., associations between domain entries and GO terms) are manually generated by expert InterPro curators. Later, these GO terms associations are transferred to protein entries based on the protein domain annotations provided in InterPro. Ensembl-Orthology[52] annotation set is generated by transferring the existing GO term annotations of a source gene/protein to a target gene/protein using the orthology information obtained from ENSEMBL-COMPARA. The ENSEMBL-COMPARA team builds gene/protein trees using a pipeline that includes HMM and BLAST searches and sequence clustering. This pipeline culminates in the inference of, orthologues and paralogues, and an evolutionary knowledge-based annotation system is built. Since these systems do not have representation vectors, their GO annotations are directly incorporated into our benchmark as predictions. Due to the fact that only GO annotations are provided by these methods, they could only be used in our PFP benchmark.

The classical protein representation methods cover homology-based (i.e., BLAST and HMMER), amino acid composition-based (i.e., AAC and APAAC) and evolution-based (i.e., K-Sep) approaches. These methods are vector-based; however, these vectors are constructed based on deterministic algorithms (e.g., pairwise sequence similarity calculation), and not machine learning.

Homology-based annotation transfer has been widely used for protein annotation. In this approach, a database of characterized/annotated proteins queried for a target protein, and the existing annotations of the source gene(s)/protein(s) are transferred to the target based on homology, evaluated via statistically significant sequence similarities [53]. In our application, we constructed two matrices containing pairwise sequence similarities between query protein sequences in our datasets and 20,422 human protein sequences (all entries in the UniProtKB/Swiss-Prot database) using (i) Blastp from the BLAST software package [54] (using BLOSUM62 and e-value threshold of 0.001, i.e. the default settings) and (ii) jackhmmer from the HMMER software package[55] (BLOSUM62 and an e-value threshold of 10, again, the default values). We used each row of these matrices as the feature vectors of the respective protein, i.e., a protein vector comprises pairwise similarities of that protein against other proteins in the human proteome. The resulting feature vectors (for both BLAST and HMMER) are made up of 20,422 dimensions, where each dimension contains a log transformed e-value that corresponds to the significance of similarity between the query protein and a protein from the human proteome dataset. Within each benchmark, we employed BLAST and HMM vectors for training prediction models, in the same manner as the vectors of representation learning methods. This similarity-based featurization approach has been used in previous studies, especially for protein function prediction[56]. Here, we did not choose to directly transfer annotations between the most similar sequences, as this application would not yield results that were directly comparable to the results of the vector-based protein representation learning methods.

Another homology-based approach we incorporated into our study was the PFAM protein family-based vectors. The difference of PFAM vectors from BLAST and HMMER is that previously identified functional information, based on protein sequence regions, has been incorporated into the PFAM vectors. In PFAM representation vectors, each protein is represented with a binary vector based on the presence and absence of Pfam[57] domain annotations of the corresponding protein. To generate these vectors, we first retrieved Pfam domain annotations of the protein entries in our dataset from UniProt[50]. We then constructed the binary vectors where each dimension corresponds to a different Pfam domain entry. Hence, we created 97 dimensional vectors for the protein-protein binding affinity prediction benchmark and 6,227 dimensional vectors for the other three benchmarks. In the protein-protein binding affinity estimation benchmark, we used the PfamScan web tool[58] to identify the domains of the sequences of PDB models in the SKEMPI dataset[59].

In the amino acid composition (AAC) method, a matrix of amino-acid frequencies was used to describe proteins[60]. In AAC, proteins are represented by vectors of 20 dimensions, each corresponding to a different amino acid. We used the iFeature stand-alone tool[61] to create AAC feature vectors. Amphiphilic Pseudo-Amino Acid Composition (APAAC) utilizes the physicochemical properties of amino acids together with amino acid compositions[62]. A general issue in amino acid composition based classical representation methods is the difficulty of including residue order information. The APAAC[62] model proposes a solution to this problem by using sequence order coupling and hydrophobic correlations together. The model calculates a representation vector with 80-dimensions (by default), in which the first 20 represent the individual amino acid compositions, and the rest represent the hydrophobicity/hydrophilicity correlation factors. The APAAC method was found to be successful in predicting enzyme sub-families using a covariant-discriminant predictor[62].

Evolutionary information is widely used in classical protein representations. In the k-separated-bigrams (K-Sep) method[6], row-type matrix transformations on position specific scoring matrices (PSSM), which are constructed using multiple sequence alignments generated from the query sequence and its homologs, are utilized for calculating the bigram transition probabilities between residues that are *k* positions apart from each other. The final representation vectors have a size of 400x1, each dimension representing a specific transition probability from one amino acid to another (20x20). The method was reported to be successful in predicting type IV secretion effectors[6].

Small-scale protein representation learning methods, which were first proposed in mid 2010's , started a new era in protein featurization and paved the way for the approaches/methods used today in the field of protein informatics. Asgari *et al.[Citation error]* were one of the first to construct a protein representation learning model by applying word2vec[10] to the problem in a method named ProtVec. This method represents proteins as 100-dimensional vectors. The authors treated each protein sequence as a sentence, and each *k*-mer (i.e., *k*-length amino acid sequence) as a word. The authors claimed that the method could be employed for different problems in protein biology including protein function prediction and protein interaction prediction. They evaluated the performance of ProtVec in predicting the mass, volume, polarity, hydrophobicity and charge of proteins, as well as its accuracy in disordered protein classification.

In the study by Yang *et al.*[25], learned protein embeddings (i.e., representation vectors) with sizes ranging from 4 to 128 dimensions were constructed using the doc2vec algorithm[20] on non-overlapping *k*-mers. We call this method "Learned-Vec" throughout our study, as Yang *et al.* did not provide a specific name. The authors measured the performance of Learned-Vec and the effects of hyperparameter optimization on four protein property prediction tasks; namely channelrhodopsin (ChR) localization, cytochrome P450 thermostability, rhodopsin absorption wavelength, and epoxide hydrolase enantioselectivity with blocking design, to compare their model with baseline models (e.g., one-hot encoding and classical feature-based representations). Performance values were calculated using mean absolute error (MAE), a measure of variation between predicted and actual values; the Kendall rank correlation coefficient, which calculates the ordinal accuracy; and log-likelihood. For three out of the four tasks, they report that Learned-Vec provided the best performance in terms of at least one of these metrics. Authors also provided 2-D t-SNE visualizations, which were consistent with the reported results.

Kim *et al.*[17] trained a representation model named Mut2Vec, producing protein vectors of 300 dimensions. The aim of the proposed model is the classification of mutations according to their disease-causing effects. In Mut2Vec, mutation co-occurrence information, protein-protein interaction (PPI) networks (from BioGRID), and biomedical literature abstracts (from PubMed) were used to construct the representation model. Considering several alternatives, they choose a model that utilizes the skip-gram algorithm[10] on co-occurrence information as the final representation. In the Mut2Vec workflow, mutation co-occurrences and PubMed texts were used first to calculate representation vectors. PPI data was integrated at the post-processing phase, using a retrofitting process similar to WordNet[63]. The authors stated that Mut2Vec could separate passenger and driver mutations successfully, and it produces promising results in the detection of new candidate cancer-related mutations.

The method Gene2Vec was proposed by Du *et al.*[19] in their study where 200-dimensional vectors are calculated to represent genes, using the skip-gram algorithm[10]. Hyperparameter tuning (e.g., vector size and window size optimization) was applied with the objective of maximizing the performance in clustering genes within MSigDB[64] functional pathways. The input data, gene co-expression profiles, were gathered from the GEO database[65]. The main objective of the study was to predict gene-gene interactions (i.e., the genes acting in the same biological process), in which Gene2Vec was reported to be successful. Additionally, it was indicated that the model could summarize latent information about genes by accurately representing functional similarities over tissue specific gene clusters. We used the gene representation vectors of Gene2Vec in our benchmarks by mapping them to canonical forms of their respective proteins.

The study conducted by Choy *et al.*, in which the authors developed the "TCGA_EMBEDDING" method (the name is given by us, as the authors did not provide a specific name), indicates that learned protein representations have potential for explaining molecular biological mechanisms of the cell and disease[66]. In the proposed method, initially a gene expression matrix of cancer samples was prepared using data from the TCGA database. The authors then applied matrix decomposition with a fully connected neural network layer. Next, through matrix multiplication on the decomposed matrices, they created a predicted version of the original matrix. The error between the original and predicted matrices was used for backpropagation. The decomposed matrices are thus the gene-feature and sample-feature matrices. The authors showed that functional relationships between samples and genes are conserved in their model. Even though the gene expression levels were not correlated, functionally related genes were observed in adjacent locations, when the multi-dimensional distance was calculated on the 50-dimensional representation vectors. Additionally, when the representation vectors were inspected, it was seen that similar cancer types were clustered in the representation space to the extent that the authors claim that molecular subtyping of cancer was possible using the representations.

The method CPCProt[67] uses Contrastive Predictive Coding[68] and calculates 512-dimensional representation vectors. CPCProt aims to maximize the mutual information between the input and output, which are defined as two consecutively located 11-mers. The parameter size of CPCProt is one tenth of its nearest competitor (ProtXLNet). Four benchmarking tasks that belong to the TAPE study[29] were used to measure the performance of CPCProt. These tasks are secondary structure prediction, remote homology prediction, fluorescence landscape prediction, and stability landscape prediction. CPCProt showed competitive performance at all tasks, although it has a notably smaller size.

Alley *et al.* developed the method ProtXLNet[36] , a Multiplicative LSTM[69] (mLSTM) backed protein representation. ProtXLNet uses a training sequence dataset with low bias (consisting of 24 million UniRef50[70] protein entries, which are filtered by a 50% similarity threshold from the from UniProtKB, instead of using all available protein sequences in the data source).The authors tested ProtXLNet on different, mostly protein engineering based tasks, including the classification of proteins based on their families and species, and the prediction of physicochemical properties and secondary structural elements. The results indicate that ProtXLNet could create physicochemically meaningful clusters. Moreover, sequentially distant homologous proteins were clustered correctly. Finally, structural information could be extracted from ProtXLNet, shown by the successful clustering of proteins based on SCOP[71]. These results were also verified using functional, evolutionary, and structural similarity labeled datasets such as HOMSTRAD[72] and OXBench[73]. The authors have also shown that ProtXLNet can predict protein stability and variant effects. In our benchmarks, we employed the “ProtXLNet Fusion" model since this version had the highest performance according to the original ProtXLNet study. This model was built with the concatenation of the “final hidden state”, “final cell state”, and “average hidden state” of the LSTM model, each of which has a size of 1x1900, providing a total vector size of 5700.

Heinzinger *et al.*[30] used Embeddings from Language Models (ELMO), a bi-directional LSTM that is popular in the NLP domain[28], to represent proteins using unlabelled protein sequence data, in their method SeqVec. SeqVec generates protein feature vectors with 1024 dimensions. The authors aimed to solve problems generally associated with global representation methods by inferring information from the local context. SeqVec yields a significant advantage in terms of training and inference times over up-to-date language models such as BERProtT5-XL[12]. However, their results show that their model could not surpass the state-of-the-art methods on sequence level tasks such as secondary structure prediction. On the other hand, the model produced results competitive to the state-of-the-art, in protein level tasks such as the prediction of subcellular localization.

Many of the large-scale representation methods use (or are derived from) either BERProtT5-XL[12] or Transformer[74] architectures. ESM1B[75] is one of the BERT-derived models which constructs 1280-dimensional representation vectors. The original BERT-Base and BERT-Large models have 12 and 24 layers, respectively. ESM1B is composed of 33 layers and 650 million parameters. The authors trained ESM1B with 250 million protein sequences acquired from the UniParc database[76]. The model was evaluated on various tasks such as the secondary structure prediction, remote homology prediction, long-range contact prediction, and mutational effect prediction. Except for remote homology prediction, ESM1B produced results that were competitive with or better than other models.

ProtTrans[77] is a study conducted by Elnaggar *et al.* In this study, different Transformer-based models were trained and compared to each other on the prediction of secondary structure, subcellular localization prediction and water-solubility prediction tasks. The models tested in our benchmark are ProtBERT-BFD (1024-D vectors), ProtXLNet (1024-D vectors), ProtALBERT (4096-D vectors), and ProtT5-XL (1024-D vectors). These models have 420 million, 409 million, 224 million, and 3 billion parameters, respectively. As regularization, *masked language modeling*, which hides %15 of the amino acids in the sequences randomly, was used during the training of these models. The source sequence dataset called "BFD" included 2.1 billion metagenomic sequence fragments and was used to train BERT, ProtALBERT, and ProtT5-XL. The ProtXLNet model was trained using the UniRef100 dataset, which includes 216 million protein sequences. The SeqVec model, mentioned above, was also included in this study for comparison. Results indicate that smaller models such as SeqVec and ProtALBERT could easily produce results comparable with larger BERT, ProtXLNet and ProtT5-XL models. All of these models produced as-good or better results in all tasks, compared with other state-of-the-art methods.

Rao *et al.*[29] conducted a study entitled "Tasks Assessing Protein Embeddings" (TAPE). The authors constructed three original sequence representation models based on; *(i)* Bidirectional Encoder Representations from Transformers (BERT)[12] (we call this model, which produces 768-dimensional vectors, TAPE-BERT-PFAM), *(ii)* an unsupervised LSTM[78], and *(iii)* ResNet[79]. These three models were trained on 32 million Pfam[80] domains. Additionally, the authors evaluated two previously developed representations, ProtXLNet[36] and a supervised LSTM[37]. This study employs three groups of tasks which are; structure-based (i.e., secondary structure prediction and contact prediction), evolutionary (i.e., remote homology prediction) and protein engineering (i.e., fluorescence landscape prediction and stability landscape prediction). For structure-based tasks, an alignment-based representation (proposed as part of the baseline models) achieved the best score. In the evolutionary tasks, the pre-trained LSTM model had the top performance. Finally, for protein engineering tasks, TAPE-BERT-PFAM was the best in terms of fluorescence landscape prediction and shared the top position with ResNet in terms of stability landscape prediction. The results indicate that no single method could dominate all of the benchmarking tasks. We incorporated TAPE-BERT-PFAM in our benchmark analyses. We used the version constructed by averaging the final hidden layer of the model. Here, a protein feature vector is calculated by taking the mean of the values in each dimension of the amino acid-based feature vectors of that protein.

MSA-Transformer is the first Transformer model that uses a multiple sequence alignment as input. Moreover, row and column-wise attention is introduced as an enhancement to previous protein language models that use the attention mechanism. With the utilisation of the attention mechanism the computational cost is reduced from$O(M^2L^2)$ to $O(M^2L) + O(ML^2)$. The model, which has 100 million parameters, 12 layers, 12 attention heads and an embedding vector size of 768, was trained on 26 million MSAs. MSA-Transformer was tested on unsupervised contact prediction, supervised contact prediction and secondary structure prediction tasks and outperformed SOTA competitors.

5. Objective-based Classification of a Comprehensive List of Protein Representations

In this section, we grouped and elaborate protein representation learning methods according to the objectives and applications reported in their respective publications (Fig. S15b). The methods that we included in our benchmark study (i.e., Learned-Vec[25], SeqVec [30], Mut2Vec[17], Gene2Vec[19], TCGA_EMBEDDING[81], ProtVec[82], TAPE-BERT-PFAM[29], CPCProt[67], ProtBERT-BFD[77], UniRep[36], ESM-1b[35], ProtALBERT[77], ProtXLNet[77], ProtT5-XL[77]In addition, classical representation methods, BLAST[54], HMMER[55], AAC[60], APAAC [62], K-Sep [6], PFAM[57] and rule/association-based models, UniRule2GO[50], InterPro2GO[51], and Ensembl-Orthology[52]) are explained above.

5.1. Methods for Physicochemical Feature Prediction

State-of-the-art NLP methods are gaining importance in the protein representation domain with their context-based dynamic inference capabilities. Rives *et al.*[35] uses one of these, Bidirectional Encoder Representations from Transformers (BERT)[35], and adapts it to the protein representation domain by predicting *masked* (hidden) amino acids on sequences where 15% of amino acids are hidden. Using 250M protein sequences from the Uniparc database[76], the model can successfully predict the masked amino acids' features such as polarity, charge, hydrophobicity; and protein secondary structure and activities such as contact and variant effects. The authors also showed that 25M samples would have been sufficient to achieve performance results on par with 250M samples. Moreover, the study results indicated that the model is far better than the baseline representation models, such as random and n-gram models. The model was also tested against an untrained LSTM and shows better performance. When internal mechanics were inspected using orthology based tests, it was observed that the BERT model-based phylogenetic organization has a good correlation with the natural phylogenetic organization even for long homologous proteins. The model was also tested for vectorial stability and protein similarity prediction. For the vectorial stability task, at first, the average distance between species is calculated in the representation space. This vector distance is added to the source protein's representation vector, and ortholog proteins are searched around the new position. The representation's vectorial stability is measured based on the precision of this search in finding true orthologs. Finally, one primary application of this representation is prediction of mutational effects which includes two subtasks: intra-protein variant effect prediction using known mutations of the target protein and generalization of mutational fitness landscape for new proteins without any prior knowledge. The proposed representation model performs on par with three state-of-the-art variant effect prediction methods, without any added knowledge about the protein, such as the structure.

The rest of the methods, where the objective is the prediction of physicochemical features (e.g., ProtVec, LearnedEmbeddingVec and TAPE_BERT models) are summarized in the Methods section of the main text.

5.2. Methods for Sequence-based Feature Prediction

To the best of our knowledge, the first application of learned protein representations was conducted by Melvin *et al.*[83] with the aim of developing a search algorithm for protein sequences. They trained a 2-dimensional protein representation vector, by integrating 3-D structural similarity and protein class label information. The accuracy of the method was reported to be notably higher compared to known alignment-based homology search methods. The method was also trained and tested using SCOP (Structural Classification of Proteins) labels. Moreover, it was shown that a faster protein sequence search is possible with simple protein representation vectors.

Qi *et al.*[84] exploited multi-task learning and developed a deep learning framework for training a protein representation model to predict various local protein properties. Previous studies showed that multi-task learning is advantageous since information extracted from different features creates a synergy, which leads to better performance[85]. The proposed model learned and predicted the secondary structure, signal peptide transmembrane topology, solvent accessibility, and protein/DNA binding residues simultaneously. In the first step of the method, a protein representation model was trained to predict naturally occurring protein sequences. During the second step, sequential feature extraction was applied to capture information at flanking regions of amino acid sequences. As a third step, a neural network composed of feed-forward layers does the classification job. Finally, a post-processing step was added using a Hidden Markov model to utilize information in position-based patterns, such as repeated sequences. The study was critical since the authors used protein representation learning models and multi-task labelling for protein feature prediction, and achieved better performance compared to previous studies.

In the study conducted by Kimothi *et al.*[21], the authors proposed a context-aware protein representation method based on the doc2vec[20] algorithm, and named it “seq2vec”. The authors claimed that a previous method, ProtVec[82], which employs the word2vec algorithm, does not fully capture the order information in the protein sequence. One important shortcoming of word2vec based models is their lack of global context awareness, which means that the order of the words is not considered in terms of the whole document in which the word exists, during the calculation of the representation vector. The authors solved this problem using the doc2vec algorithm, which calculates a vector for each word using neighbouring vectors and a document vector. The document vector represents features of the whole document, such as its topic. According to the results of the study, seq2vec approach is notably better than ProtVec on protein family classification tasks.

In a follow-up study to ProtVec by Asgari *et al.*[86], a variable-length protein sequence segmentation approach is introduced. In the proposed method, ProtVecX, the byte-pair encoding[87] which had also been used in the field of neural machine translation, was adapted to the protein sequence representation domain. Also, as one of the first applications in this domain, a baseline is defined using k-mers occurrences. Ablation studies indicated that this is a successful baseline for protein classification that can be used in future studies as well. It is also important to note that there still is a strong requirement for baseline models in the domain of protein representation learning. ProtVecX, could not display a significant performance advantage against the k-mer-based method on protein classification tasks.

Xu *et al.*[23] exploited learned protein representations in their method called PhosContext2vec, with the aim of recognizing phosphorylation sites on protein sequences. Phosphorylation is one of the fundamental control mechanisms for the cells; thus, the identification of phosphorylation sites is a critical problem in protein science. In PhosContext2vec, both word2vec and doc2vec algorithms were applied together with overlapping n-grams, which produced a better performance compared to non-overlapping n-grams. For the identification of the phosphorylation sites, the support vector machine (SVM) algorithm[88] was utilized. The model used learned representation vectors and six other protein residue-level features such as Shannon entropy, relative entropy, disordered protein regions, secondary structures, Taylor's overlapping properties, and the average cumulative hydrophobicity. The results indicated that word2vec and doc2vec were successful in different cases, thus complementing each other, and neither one was superior in terms of the overall performance. When PhosContext2vec was compared with other phosphorylation site prediction methods on semi-independent tests, it was shown that PhosContext2vec outperforms or competes with them in different cases.

The method D-Space (Deep Semantic Protein Annotation Classification and Exploration[32]) is based on a convolutional neural network (CNN), which produces 256-dimensional representation vectors from protein sequences. In the respective study, these vectors are used to predict labels from 13 different sources including PFAM[80], InterPro[89], EC Number[90], GO[91] and PROSITE[92]. Multi-task modeling is one of the solutions proposed and applied for the scarcity of labelled data[93–95]. When the representation model is tested, it is observed that over 400.000 proteins can be grouped in correlation with their OrthoDB[96] cluster label. OrthoDB is a database of orthologous protein-coding genes, and the correlation is assumed to be a marker of the accuracy of D-Space. The method's speed and sensitivity were tested on 109M protein entries from the UniprotKB, and compared against the results of a standard BLAST search. The results showed that the proposed method is fast (i.e., 5 seconds to find homologous sequences of a query protein, on average, compared to several minutes when BLAST is used). A correlation analysis between sequence identity and representation vector similarity supported the results with R=0.84 with p-value<2.2e-16. The study also included promising results on searching functionally related proteins and variant effect analysis; however, these results were mostly case-based and insufficient to infer generalization.

Cohen *et al.*[97] used vector symbolic architectures[98] with a set of quantum-related compositional operators to generate protein representation vectors. These orthographic vectors are able to represent words or k-mers and employed to overcome the dependency of protein representations on the exact locations of amino acids. Different physicochemical properties of amino acids are encoded into these vectors to represent the proteins. The representation and similarity measures are tested using immunoglobulin (Ig) sequences gathered from patients infected with the West Nile Virus (WNV). The task was the identification of WNV-specific clonal lineages between thousands of different Ig sequences. The study showed that the best overall results were obtained by the proposed method, where the alternatives were models based on bag-of-amino-acids and bag-of-properties. This approach may be promising not only in terms of the provided results but also considering the potential applications on quantum computers.

It is known that the smallest fully independent functional building block of a protein is a domain. In the study conducted by Viehweger *et al.*[26] authors use protein domains where a word vector model is trained via the doc2vec[20] algorithm, using protein domain sequences as words. The proposed method is called nanotext. The study was the first to train a protein representation using metagenomes (from 32 thousand genome assemblies). Training models for metagenomes is a critical issue since metagenomes have billions of records and most of their functions are unknown. The accuracy of the representation model was tested with the semantic odd man out (SOMO) approach, which can be defined as identifying the odd word in a context, where the method achieved more than 99% accuracy. When the representations were visualized with t-SNE, it was observed that clusters were correlated with the enzymatic functions of proteins. When genome vectors were used instead of domains, it was noted that nanotext could infer the taxonomic information, even for highly incomplete genomes. Finally, using nanotext, various features of bacteria, such as the culture medium and water temperatures, in which the bacteria were sampled, could successfully be predicted. The results indicated that there might be many other potential applications of protein representations.

You and Zhu[99] utilized text data for automated protein function prediction in terms of gene ontology-based annotations, in their method called DeepText2GO. Biomedical texts are one of the least exploited types of data in protein representation learning since the inference of relevant information is difficult. The main novelty of DeepText2GO was that the authors used the text data (trained on abstracts of MEDLINE[100]) and the homology information together, since each was reported to be successful in different tasks (i.e., sequence-based homology gave a better performance on molecular function prediction, and text-based information was more successful in predicting biological processes and cellular components). In DeepText2GO, TF-IDF and doc2vec[20] methods were used for the text-based classification, and BLAST-KNN and logistic regression were used for the sequence-based classification on InterPro[89] where domains, families, and motifs were used. The algorithm produced comparable results to the state-of-the-art methods on the CAFA2[7] benchmark dataset. One interesting finding of the study was the better performance of TF-IDF over doc2vec, which was also consistent with the latest study of Asgari *et al.*[34]. This finding indicates the shortcomings of static word vectors and the requirement for more sophisticated representation models.

Jaeger *et al.*[18] developed a cheminformatics-based application of word2vec. The method, named Mol2Vec, was trained as an unsupervised representation model. Here, the aim is to create a representation for molecular substructures that could be used for various tasks, including drug discovery and repositioning. SMILES[101] representations and Morgan fingerprints[102] were used to create the input to the method. Continuous Bag-of-Words (CBOW) and skip-gram algorithms of the word2vec[10] were tested on different tasks (e.g., regression-based prediction of solubilities, classification of mutagenic and non-mutagenic compounds, and the prediction of compound toxicities), and the most successful one was selected for each task to represent molecules. Also, a combination of ProtVec[82] and Mol2Vec was created and tested (named as PCM2Vec). It was shown that PCM2Vec could successfully predict kinase bioactivities. Also, both Mol2Vec and PCM2Vec could attain success comparable to the state-of-the-art methods at every task. Finally, Mol2Vec was used to visually illustrate the molecular substructures.

There are also additional studies aiming for the prediction of sequence-based features that are worth mentioning. In the study conducted by Faisal *et al.*[103], the authors divided the protein sequence into segments and calculated position-free descriptors, such as amino acid composition, dipeptide composition, and normalized Moreau-Broto, on each segment, together with position-based numerical features. When implemented with the SVM algorithm for feature selection and classification, the method scored a slightly higher accuracy over ProtVec in terms of protein family prediction. Kane *et al.*[27] aimed to predict the functions of proteins by augmenting protein sequence representation vectors with protein-protein interactions. Their most successful representation vectors were compared to random representation vectors, and there was a slight increase in terms of the area under the ROC curve (from 0.5 to 0.6). Strodthoff *et al.*[38] showed that a pre-trained protein representation could transfer knowledge from large unlabelled datasets to labelled small data sets. This approach was tested for enzymatic function prediction using EC numbers. Moreover, their AWD-LSTM[104] based model performed well compared to other state-of-the-art methods for protein function prediction and homology detection.

Bileschi et al. developed the ProtCNN model[105] for protein family annotation (using Pfam) which is a significant problem, especially in metagenomics. BLAST and hidden Markov models are widely used for this problem. However, the performance of these models decreases at low sequence similarities between the training and test datasets (i.e., cases of distant homology). The authors designed a ResNet[106] based CNN model (ProtCNN) to address this problem. At the input level, each amino acid is represented with a one-hot embedding vector. These vectors were appended to a tensor which at the end represents the protein. The tensor was used as input to the ResNet layer, and then mean/max-pooling was applied for calculating an 1100-dimensioned protein representation vector. This vector was used to predict Pfam-based protein families. The model was trained with supervised learning using protein family classification tasks. Also, an ensemble-based model was developed based on the ProtCNN model (ProtENN). Results of the study showed that ProtCNN and ProtENN produced better results, especially when the sequence identity between the train set and the test set is low.

DeepSequence[47] is a model that could predict mutational effects. The model used MSAs as input. A variational autoencoder (VAE) was trained using MSAs and a prior distribution was calculated. The same VAE model was also used as a generator to predict mutational effects. The sequence except the mutant residues were given to the model as input. Then wild-type and mutant generation probabilities were calculated and proportioned to find the mutation probability. The mutational effects were predicted using these probabilities. The model was compared with SOTA mutation effect prediction tool EVMutation[107] and outperformed it.

5.3. Methods for Interaction Prediction

Wan and Zeng[108] exploited vectorial representations of proteins for the prediction of interactions between compounds and target proteins. The authors employed a 3-D structure-free drug-target interaction prediction method using latent semantic analysis and word2vec[10] algorithm to learn both the compound and protein representation vectors. In this work, the substructures of compounds (which are created using Morgan fingerprints) and k-mers of protein sequences were considered as words, and the whole compounds and protein sequences were considered as sentences. The method uses skip-gram with negative sampling. After constructing the independent compound and protein representations, authors concatenated the vectors and input to a deep neural network, to predict the interaction between the compound and protein of interest. The performance of the model was evaluated using known interactions of drugs and drug-candidate compounds against proteins. Well known databases such as ChEMBL[109] and DrugBank[110] were used as data sources. According to the authors, results are promising and better than conventional approaches.

In their method called DeepDTA[111], Ozturk *et al.* created learned representation vectors for proteins and compounds to predict the drug-target binding affinities. The authors used SMILES notations of compounds and protein sequences to create the representation vectors. Using character-based representation vectors for ligand representation, instead of words, is novel since most of the representation learning methods utilized the latter for similar tasks. DeepDTA utilizes CNN to train the representation vectors. Afterwards, the representation vectors are aggregated and given to a fully connected deep neural network for prediction. Concordance Index (CI) and mean square error (MSE) measures were used as evaluation metrics on two different benchmark datasets. The authors also conducted a performance analysis, which indicated that DeepDTA had a lower MSE and a higher CI value than the compared methods. In a follow-up study from the same group, authors utilized protein domains, motifs and the maximum common substructure information within a similar algorithmic framework and developed the method WideDTA[33]. The results indicated that this data did not contribute to the model performance, but interestingly, using only domain and motif data produced competing results to using the full protein sequence, which indicates that a significant amount of ligand binding information is located in domain/motif regions.

Yao *et al.*[112] developed a deep learning pipeline named DeepFE-PPI to predict protein-protein interactions (PPI). In this method, proteins are represented using a word2vec[10] based algorithm (skip-gram) named Res2vec, which calculates a vector for each interaction between residues. Then, multiple dense layers were formed to predict PPIs. PPI networks belong to S. Cerevisiae, and Homo sapiens were used to test the proposed method. According to the authors, DeepFE-PPI achieved the best performance for most of the test cases.

Zhang and Kabuka[113] claimed that they developed the first deep multi-modal PPI prediction system. The authors stated that traditional PPI prediction techniques mostly rely on the protein sequence. In the proposed method, features based on the whole protein context (amino acid composition), protein sequence (using a stacked autoencoder), and PPI graph (using continuous bag-of-words algorithm) are utilized together to predict PPIs. Also, protein family prediction capabilities of the output representation were tested. The results indicated that the performance of the proposed approach was reported to be better compared to the state-of-the-art methods.

5.4. Methods for Structural Feature Prediction

Nguyen *et al.*[114] developed DeepCon-QA, a deep convolutional neural network model that uses continuously distributed protein representation (ProtVec) and protein profiles as input to predict a protein distance matrix that represents protein structure for quality assessment of predicted structures. Quality assessment is part of the protein structure prediction process which can be defined as measuring the similarity between the true and the predicted 3-D protein structures. The study results indicated that the utilization of protein representation vectors based on word2vec models created a significant performance improvement compared to using the protein profiles alone. It was also shown in the paper that DeepCon-QA competes with the state-of-the-art quality assessment methods. Hence, it is possible to comment that protein representation models have the potential for various types of *in silico* protein analysis tasks.

Mirabello and Wallner[46] trained protein representation models using a deep learning architecture and the multiple sequence alignment (MSA) data. The model predicted the secondary structural elements, relative solvent accessibility, and protein contact maps. In the first step, the input MSA was represented with an embedding layer. After a 2-D CNN with pooling was applied, an LSTM was employed to generate the predictions for secondary structures and relative-solvent accessibility. Besides, another CNN was employed for the contact-map prediction. Bias due to homology contamination (i.e., precise separation of training and test sets according to protein sequence similarity and protein family classification) were also taken into account and filtered during the model development phase. The results indicated that the proposed approach was successful for all of the above-mentioned prediction tasks. The method could compete with the state-of-the-art approaches in protein contact map prediction, based on the benchmarks.

Asgari *et al.*[34] conducted a comprehensive study on the secondary structure prediction task. The authors evaluated one-hot vector representations, biophysical scores of amino acids, amino acid protein vectors, contextualized embeddings, and a Position-Specific Scoring Matrix (PSSM) as input. On the model side, various combinations of CNNs and LSTMs were applied to predict secondary structures in the CB513 Q8[115], which is a challenging dataset with eight different classes. The results showed that models that combine biophysical features, one-hot encoding, and PSSM achieved the best results. Still, it was shown that most of this accuracy (more than 99%) originated from the PSSM. On the model side, the ensemble of 100 different neural networks achieves the highest performance, but similarly, most of this score (more than 99%) originated from CNN-BiLSTMs. Finally, an inspection of confusion matrices showed that prediction accuracy decreased dramatically in the border regions of protein secondary structures.

There have been various efforts[30,36,82] to predict the structural features of proteins with unsupervised representation learning, but Bepler and Berger[37] took these efforts one step further.

The authors developed a Bidirectional LSTM model that was trained on labelled structure data. A critical problem in sequence-based protein representation learning model development is the loss of positional correspondence during the representation vector calculation. Authors proposed a new solution to this problem with a new similarity measure they call “soft symmetric alignment” (SSA), a symmetrisation of the directional alignment commonly used in attention mechanisms. Using SSA, the model achieved the best structural similarity classification performance based on SCOP[71]. Moreover, the paper states that the developed model also produced improved contact and transmembrane prediction results.

The study conducted by Tubiana *et al.*[44] has addressed two critical problems in the protein representation learning domain. The first one was the interpretability of representation models. Since most of the models in this domain are based on black-box deep learning methods, it is hard to interpret and understand the internals. Moreover, finding the essential features of a protein representation vector is also an intricate effort. The second issue is related to the trend towards huge models[77]. Although there are ablation studies made for preventing overfitting, these models might tend to memorize patterns. Ramajuan *et al.* [116] showed that even with random weights, deep neural network models include optimal subnetworks, representing any mathematical function. The authors, Tubiana *et al.* used Restricted Boltzmann Machine (RBM) to model protein sequences using multiple sequence alignments of protein families. The network learned stochastic functions, which defined a two-way mapping between the protein sequence and the representation space. Since RBM depends on the Boltzmann distribution, the interpretability of the network is not a complex problem, as it is in black-box methods. Also, RBM’s low number of parameters reduced the training cost. The study results were also remarkable, in that, using the statistical base of the RBM, that it was possible to design proteins. Moreover, they demonstrate that it is possible to activate or deactivate the desirable features of a protein by conditioning the model. In the application phase, an accurate contact prediction could be made using the proposed model. In this study, a new type of activation function, a double Rectified Linear Unit (dReLU), was introduced, which contributed to the success of the model.

5.5. Methods for Genetic Feature Prediction

Oubounyt *et al.*[117] exploited word2vec[10] and doc2vec[20] based sequence representation models to predict the percentage of splicing inclusion (PSI) in the context of alternative splicing. Using datasets from different tissues, the authors indicated that sequence representation learning models may be effective in predicting PSI. In this method, learned sequence representation vectors are given to the Inception Network[118] for PSI classification (as low, medium or high), and after that, PSI calculation with regression. The results of the study indicated that word2vec and doc2vec models could capture similar features, and the deep learning model trained using these features outperformed the state-of-the-art methods for PSI calculation.

In their study, Dutta *et al.* [119] utilized word embeddings to calculate representation vectors using word2vec[10] and doc2vec[20]. The study aimed to solve the RNA-Seq data-based intron boundary recognition problem, since RNA-Seq suffers from misalignment of short reads. During the model development, 3-mers of the genetic sequences are considered as words, sequences with 2000 nucleotides are used as sentences, and splice junction sequences are used as the context. After the representation model is trained, a simple multilayer perceptron is used to detect the splice junctions and to annotate intron boundaries. The accuracy of the doc2vec based model was found to be better than word2vec. The results of the study also indicated that classification after representation training was invariant to class imbalance problem.

Mejía-Guerra and Buckler[14] developed a model to represent k-mers and classified different regions of the genetic sequences either as regulatory or random by learning complex patterns in regulatory regions. The authors employed word2vec and bag-of-words methods to define k-mer representations. A logistic regression model was used to create the bag-of-words representation ("bag-of-k-mers"), and a recurrent neural network was used to train the "vector-k-mers" representation. Although the accuracies of both models are satisfactory (over 90%), bag-of-k-mers outperformed vector-k-mers model.

Ng[15] developed a DNA sequence representation model and demonstrated the stability of their model empirically. The model is named dna2vec. The author used word2vec with the skip-gram algorithm to train the model, using a three-step procedure. The first step is dividing sequences into long non-overlapping fragments using gap characters (such as "X", "-" and etc.). This approach was also used in phylogenetic analysis[120]. The second step is converting long fragments to overlapping variable-length k-mers. In the third step, skip-gram and negative sampling is applied to train the model. Two different tasks were used to prove the stability of the trained representation model. The first one examined whether the summation of two k-mer sequence vectors was equal to the vector of the concatenated sequence. The second one was searching for a correlation between similar k-mers and the global alignment. For both tasks, the developed method was deemed successful.

Choi *et al.* trained a model named G2Vec[16] using random paths of functional interaction networks as sentences and genes as words. G2Vec, which is based on the continuous bag-of-words (CBOW) algorithm, was utilized for finding prognostic genes. The study showed that genes that imply good and bad prognosis could be separated in the representation space, and it can be used to discover new biomarkers.

6. Traits of Successful Protein Representations

Traits of good data representations were first defined in a study by Bengio *et al.*[121] and it is still widely accepted in the representation learning literature. In this section, we evaluated these properties in the context of protein representations.

Smoothness: If small changes on the input sample cause small changes in the output representation vectors, then the representation can be considered as smooth. Evaluating this information in terms of protein representations; small perturbations in the protein sequence (e.g., single residue variations between the amino acids with very similar physicochemical properties, on a non-critical/non-conserved region of the protein sequence) generally do not cause significant changes in the structure and function of the protein, hence, a successful protein representation should be smooth. However, if there is a single amino acid variation at a critically important region of the protein sequence and structure, such as a binding site, the variation should be observed on the representation vector as well, to reflect the change in the functional traits of the protein.

Explanatory factors: Explanatory properties are important for representation vectors. When the biological, chemical or the physical feature that corresponds to a specific dimension on the representation vector is known (e.g., the nth dimension of a protein representation vector corresponds to the hydrophobicity value), feature importance, which is just a quantitative measure of how much that specific feature contributes to the task at hand (e.g., prediction the function of a protein), can be associated with a real-world biological/chemical/physical property. Another advantage is that the vector can be reused partially with only its relevant selected features. Explanatory properties can also be utilized to construct more compact representation vectors. Explainability is usually not a problem for classic representations since the calculated values that correspond to different biological/chemical/physical features are just concatenated to generate the feature vectors; however, trained representation vectors are not inherently explanatory. Nevertheless, there are methods to uncover the explanatory features of a high dimensional representation[122–124]. When the explanatory features are discovered, it is also possible to build a task-specific low dimensional representation from the original high dimensional vector. Moreover, these features may be used towards completely different tasks such as the design of novel proteins with desired properties.

A hierarchical organization of explanatory factors: Proteins can be classified using hierarchical organizations. Hierarchical classification systems for proteins such as SCOP[125] and CATH[126] are widely used in the protein informatics domain. In these systems, hierarchical organizations are created based on the experimental knowledge on the proteins. For example, in SCOP classes are groups created from fundamental secondary structures such as alpha-helices or beta-sheets, moreover, folds are groups placed under the classes and consisted of combinations of secondary structures (e.g., “7-bladed beta-propeller” - consists of seven 4-stranded beta-sheet motifs, meander). Similarly, the hierarchical organization of explanatory factors contributes to the development of modular representation models for learned representation vectors. The modularity might be important for the reusability of representation vectors. For example, some features of the protein representation vectors can be used to predict high-level entities such as the SCOP classes (e.g., “All alpha proteins”) and other features might be used to predict more specific entities such as SCOP families (e.g., “Eukaryotic proteases”).

Shared properties across different tasks: A protein representation model, which was trained based on a specific task (e.g., contact map prediction), should be used towards other tasks (e.g., protein function prediction) as long as there is a biological, chemical or physical relationship between these tasks.

Natural clustering: Manifolds can be defined as lower-dimensional representations of information and this dimensionality reduction can be useful for data processing such as noise reduction. When the probability distribution of the training data has high-density regions, a manifold can be discovered along these regions which lower the dimension of the data representation. Naturally, low-density regions are observed between these manifolds which are similar to valleys between mountains. These low-density regions can be used to cluster samples. Manifolds are also important for defining simple/linear dependencies between features of the representation vector (also known as the simplicity of factor dependencies). The output of a successful representation model should be easily separable using a simple linear classifier/regressor, even though the input samples have complex and non-linear relations between each other. In the benchmarks of this study, we use linear classifiers on protein representation vectors to classify them. Hence, a good protein representation can be naturally clustered.

Sparsity: In general, only a few of the dimensions in a representation vector include information relevant to a task of interest. From a statistical point of view, most of the factors are not sensitive to small changes. This property manifests itself as sparsity in the representation vectors. For example, active sites of a protein cover only a small fraction of the protein sequence, but an amino acid substitution in this region might cause significant changes in protein function, and this information may be captured by a sparse feature on the representation vector. This property can be utilized for interpreting/explaining the trained models by associating the output features with certain biological/physical/chemical properties of input proteins.

7. Performance Evaluation Metrics

In our semantic similarity inference benchmark, we used Spearman rank correlation[127]. For a sample with size *n* and the ranks of variables $rg_{x_i}$and rg_{Y_i}, Spearman rank correlation (r_S) can be defined as:

$$r_s = 1 - \frac{6\sum_{i=1}^{n} {d_i}^2}{n(n^2-1)} \tag{1}$$

where difference between ranks for observations is defined by:

$$d_i = rg(X_i) - rg(Y_i) \tag{2}$$

For ontology-based protein function prediction and drug-target protein family classification benchmarks, we mainly used recall, precision, F1-score, accuracy, Matthews correlation coefficient[128] (MCC) and Hamming distance[129] metrics, to evaluate the predictive performance of protein representation learning methods. The formulae of these evaluation metrics are given below:

$$accuracy = \frac{tp+tn}{tp+tn+fp+fn} \tag{3}$$

$$recall = \frac{tp}{tp+fn} \tag{4}$$

$$precision = \frac{tp}{tp+fp} \tag{5}$$

$$F1\ Score = 2.\frac{precision.recall}{precision+recall} \tag{6}$$

$$MCC = \frac{tp.tn - fp.fn}{\sqrt{(tp+fp)(tp+fn)(tn+fp)(tn+fn)}} \tag{7}$$

where *tp* denotes number of true positive predictions, *fp* denotes number of false positive predictions, *fn* denotes number of false negative predictions, *tn* denotes number of true negative predictions. Finally, the Hamming distance (D_H) is defined by:

$$D_H(u,v) = \frac{1}{k}\sum_{i=1}^{k}(1-\delta_{v_i u_i}) \tag{8}$$

where *u* and *v* are 1-dimensional arrays of real and predicted class labels, respectively, δ is the Kronecker delta function, and k is the vector dimension.

In ontology-based protein function prediction benchmark, the F1-score, and its components precision and recall, are weighted proportionally to the inverse of the class sizes. This weighting operation was necessary for unbiased analysis, as the classes were highly imbalanced. In both ontology-based protein function prediction and drug target protein family classification benchmarks, models were designed as multi-task (i.e., five GO terms are predicted by one function prediction model, and five protein families are predicted by one family classification model). In ontology-based protein function prediction benchmark, the models were also designed as multi-label, where more than one GO term can be predicted for a test protein (since a protein can have more than one function). In this setting, a random predictor would produce a correct prediction in one out of 32 cases (i.e., 2^5 different combinations exist for a label vector of size 5x1, one of which is the true label vector). However, models are designed as single-label in the drug target protein family prediction benchmark (since each protein can only belong one of the main families), meaning that a random predictor would produce a correct prediction once out of five cases (i.e., only five different combinations exist for a label vector of size 5x1, one of which is the true label vector).

We calculated mean squared error (MSE) and mean absolute error (MAE) in the protein-protein binding affinity estimation task, these being the same metrics used in the PIPR study. The formulae of these metrics are given below:

$$MSE(y, y') \; = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} (y_i - y_i')^2 \tag{9}$$

$$MAE(y, y') \; = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} | \, y_i - y_i'| \tag{10}$$

where *y* denotes the ground-truth value, y’ denotes the estimated value, and $n_{samples}$denotes the number of samples.

8. Extended Results

8.1. Semantic Similarity Inference Benchmark

This is the extension of the results of the semantic similarity inference benchmark provided in the main text.

ProtT5-XL model[130] , which uses the transformers with encoder-decoder architecture in two different models with 3B and 11B parameters, is one of the top NLP models as of May 2021. In this study, the authors designed a large-scale ablation study where different fine-tuning strategies are applied to find the best combination. Attention mask patterns, transformer architecture variants, and corruption methods are notable strategies combined in ProtT5-XL, which uses 800 GB of text data for unsupervised training. In the ProtTrans study[77] , the authors used the ProtT5-XL model with 3B and 11B parameters to train a protein representation model (we used the 3B parameter model in our study) using 2.1 billion metagenomic sequence fragments. This model outperformed others in the ProtTrans study. Similarly, the ProtT5-XL model achieved the top results in our tests. We believe that carefully chosen architectural combinations, the rich pre-training dataset, and the large scale of the model could explain the high performance of the ProtT5-XL model in our benchmarks.

Mut2Vec[17] was initially developed to predict the effects of mutations; however, it performed very well in our analysis considering the BP based semantic similarities. The model was developed using patient mutation profiles, biomedical literature and protein-protein interactions. The last two datasets may include information considering the role of the proteins in BPs. For example, if two proteins interact, observing them as a part of the same BP is highly probable. Similarly, supposing two proteins had a role in the same BP, they may frequently be observed together in the same text (e.g., article). As a result, those proteins are more likely to be embedded proximally in the functional vector space. We suggest that the top performance of Mut2Vec probably depends on these factors. The PFAM[57] representation is created based on domain annotation information. If a domain exists in a protein, the corresponding cell is filled with 1, otherwise with 0. Although PFAM representations are generated with a simple architecture, it produces top results for the CC prediction. Since members of a protein family are often located in the same subcellular location(s) (e.g., GPCRs in the membrane, transcription factors in the nucleus), the high performance of PFAM in CC prediction is expected. It should be noted that the performances of both Mut2Vec and PFAM may be artificially inflated due to a possible data leak. We discuss this topic, in detail, in the discussion section.

TCGA_EMBEDDING[66] exploits gene expression data and a simple learning system inspired by non-negative matrix factorization. With this approach, the authors constructed a representation with a vector size of 50, which is one of the smallest representations in our benchmark. TCGA_EMBEDDING achieved notable performance in the prediction of CC and BP-based semantic similarities. Similar to TCGA_EMBEDDING, the Gene2Vec[19] model utilizes gene co-expression data with the skip-gram algorithm, and performed well in both BP and CC based semantic similarity inference tasks. Gene (co)expression profiles are one of the least studied data types for developing protein representations as only a few studies exist; however, it is informative to infer similarities between proteins in terms of the BPs they take part in and the CCs that they localize to. Previous literature reports that there is a correlation between the expression profiles and the subcellular locations of genes/proteins, as evaluated in the context of machine learning-based prediction of protein localizations[131,132]. The results of the CAFA Pi challenge (over the bacterial motility and biofilm formation biological processes) also support this, noting that gene expression was a critical input for predicting the biological roles of proteins[53].

TAPE-BERT-PFAM[29] had the penultimate place in the MF-based similarity inference task. TAPE-BERT-PFAM (a bi-directional transformer) is the only method in this comparison that uses the self-attention mechanism. The technical details of BERT and self-attention are discussed elsewhere[133,134]. Similar to the success of the BERT model in natural language[135], the TAPE-BERT-PFAM model can represent protein sequences with high accuracy. It should also be noted that the implementation we used here was directly obtained from the TAPE benchmark study[29] without any fine-tuning. TAPE-BERT-PFAM may indeed perform better with further optimization.

Finally, Learned-Vec[25], which processes protein sequences with doc2vec[20], scored as well as the TAPE-BERT-PFAM[29] models in our semantic similarity based analysis. The size of the TAPE-BERT-PFAM models were notably larger compared to Learned-Vec[25] (i.e., 12 hidden layers with 768 neurons for each layer, as opposed to 1 hidden layer with 64 neurons). We argue that some of the shallow models still preserve their significance, especially in MF-based semantic similarity inference.

8.2. Ontology-based Protein Function Prediction

This is the extension of the results of the ontology-based protein function prediction benchmark provided in the main text.

Here, we also discuss a critical topic that was mostly overlooked in previous PFP studies, the assessment of performance in terms of annotated GO term specificity. This is important since there is a relationship between the specificity of a GO term (i.e., its location of the graph of GO) and its informativeness. For example, an annotation with the GO term “negative regulation of molecular function” is too general. These GO terms are generally located near the root of the GO graph and called “shallow terms”. If the same protein were annotated with the term “negative regulation of double-stranded telomeric DNA binding”, which is a descendent of the former, the annotation would have been more informative. In order to take this into account, we grouped GO terms under three categories as shallow, normal, and specific; according to their depth on the GO graph (see Methods).

One key problem in applying deep learning to any field is the requirement for a high amount of training data[136]. To handle this issue in our benchmark, we grouped GO terms based on the number of proteins that they annotate. This is expected to uncover the performance of representation learning models when learning with only a few training examples, which is the case for a considerable number of informative GO terms. Furthermore, some protein functions are well studied and others are under-studied, and this creates a discrepancy in terms of the number of annotated proteins for each. We expect that our approach will be useful in assessing the representations' ability to learn under-studied functional properties. To this end, we created three categories that point out the number of proteins annotated to a GO term as; low, middle, and high (see Methods).

We observed that the methods are clustered together in all three heat maps, based on their performances on different datasets (Fig. 3). ProtT5-XL, ProtALBERT, SeqVec and ProtBERT-BFD share common characteristics, which may explain their performance similarities in the PFP benchmark. First of all, they are all based on large state-of-the-art sequence modeling algorithms: LSTM (e.g., SeqVec with 93M parameters) and transformers (e.g., ProtT5-XL, ProtALBERT, ProtBERT-BFD with 3B, 224M, 420M parameters, respectively). Additionally, they share similar model training objectives. Finally, they were all trained with large datasets (i.e., 2.1B sequences for ProtT5-XL, ProtALBERT, ProtBERT-BFD, and 33M for SeqVec). The performance of SeqVec is notable here since it has a lower computational cost but was able to compete with larger models that were trained with extensive datasets. SeqVec uses the ELMO model[28], a bi-directional LSTM with 93M parameters capable of learning long sequential patterns, which is stated to be highly efficient for language modeling[135]. The most evident difference of SeqVec from the other successful state of the art models in our study is that SeqVec contains a CNN layer, before the LSTM layers, to embed the amino acids in the sequence onto a latent space. In the original ELMO model, the same approach, charCNN[137], was used to obtain word vectors of fixed size. It is also important to mention that SeqVec displayed a moderate performance on the semantic similarity inference benchmark. This indicates that, although protein function prediction and semantic similarity inference can be seen as correlated tasks, specialized solutions are required for each one. The moderate performance of SeqVec on semantic similarity inference might be explained by the noise on the original representation vectors that SeqVec produced. This noise may have been filtered out due to simple feature weighting done by the linear classifier during training in the PFP and drug-target protein family classification benchmarks. As a result, SeqVec was successful. However, there was no classifier in the semantic similarity inference benchmark. This phenomenon was also observed in the original SeqVec study (see Fig. 2 of the SeqVec paper[30]).

We should note that the rule-based methods (UniRule2GO[50], InterPro2GO[51], Ensembl-Orthology[52]) produced low F1-scores. One reason for this is the prediction philosophy applied by curation teams. In most of the rule-based methods, the main objective of optimization is avoiding false positives (i.e., maximizing precision) as much as possible, even though this can result in dramatic decreases in recall. However, in our testing, we give equal importance to false negatives and false positives and aim to maximize precision and recall together (e.g using F1-score). When the mean precision values are inspected in Table S4, it is seen that rule-based methods score far higher in terms of precision compared to recall. The Ensembl-Orthology method was especially among the best in the GO BP term prediction category, in terms of precision.

We calculated model performances averaged for each GO group (Table S5), especially to investigate the scores for challenging groups. One such group is the "low" group which contains GO terms that annotate a low effective number of proteins (during dataset preparation, we eliminated highly similar proteins by filtering through UniRef50 clusters). The other challenging group is the "specific" group, consisting of GO terms that are leaf nodes, or are close to the leaf nodes, in the GO hierarchy. In Table S5, mean F1-score results indicate that a low number of sample proteins is a problem for the BP and CC categories, but not so much for MF category, where there is generally an explicit relationship between the input (i.e., sequence) and the respective label (i.e., GO term). On the other hand, we could not observe a clear performance differential based on GO term specificity. As a result, it can be stated that prediction success for these informative and specific terms may be solely related to the effective number of proteins annotated to these terms. For the MF-low category, ProtT5-XL and HMMER (F1-score: 0.89) achieved the best performance. In the BP-low category, ProtT5-XL, HMMER and ProtBERT-BFD (F1-score: 0.52) shared the top place. Finally, in the CC-low category, ProtT5-XL produced the best score (F1-score: 0.58). For the MF-specific category ProtT5-XL and HMMER were the best with (F1-score 0.92), and in the BP-specific and CC-specific categories ProtT5-XL was the best performing method (F1-score: 0.72 and 0.57, respectively).

These results show that, for the tasks where the number of labelled data points are low, the representation capability of classical (model-driven) methods can still compete with learning-based (data-driven) models in two out of three categories.

Finally, we checked for a statistically significant difference between the F1-scores of the methods. We sampled a performance score distribution for each method from the F1-scores of each fold of the five-fold cross-validation, and compared these distributions with each other using the Wilcoxon Rank Sum test[138]. We applied the Shapiro-Wilk test since for some of the methods the scores were not distributed normally. Following the calculation of p-values, we applied Benjamini-Hochberg multiple test correction[139] and presented FDR values in Table S6 as a heat map. We had to exclude rule-based methods, as they have no model to train/test, only annotations from predefined rules, as such thecross-validation analysis was inapplicable to them. Overall, the best performing method in the protein function prediction benchmark, considering all GO categories, was ProtT5-XL[77]. The performance difference between ProtT5-XL and the other high performers was statistically significant in the CC and BP categories (except ProtALBERT for the BP category). The difference was not significant for the MF category, where other representation learning-based (e.g., SeqVec and ProtALBERT) and classical (e.g., BLAST and HMMER) methods performed nearly as well as ProtT5-XL.

8.3. Drug Target Protein Family Classification

This is the extension of the results of the drug target protein family classification benchmark provided in the main text.

Similar to the ontology-based protein function prediction benchmark, here we preferred to use a multi-task linear SVM classifier in order to solely measure the ability of protein representations in extracting the complex protein attributes/properties. Since there is an imbalance in terms of the number of samples for each class (i.e., protein family), MCC was taken as the most reliable indicator in comparing the representation methods.

The success of ProtT5-XL can be explained by the high number of parameters (3B), by which ProtT5-XL probably learns distant relationships between proteins, as this is also stated in its original study[77]. The success of ESM-1b especially on challenging datasets such as 30% and 15% splits also support this idea, as ESM-1b is the second largest model in this study with 650M parameters and got the third place in terms of overall performances (MCC: 0.92 and 0.86 on 30% and 15% split datasets). ProtALBERT, on the other hand, which took the 2nd place overall, is an interesting case. The number of parameters in ProtALBERT (i.e., 224M) is lower than the top models but has the highest number of attention heads (ProtALBERT has 64 attention heads and ProtT5-XL has 32 attention heads). We discussed the probable effects of the number of attention heads under the protein-protein binding affinity prediction benchmark. Finally, the PFAM classical representation method achieved a notable performance in this benchmark; however, there is a certain data leak from training to test for PFAM in this task, as there are many domains that directly correspond to main protein families used as classification tasks. Annotations to these domains are encoded into the feature vectors of respective proteins. The topic of data leak is discussed in the Discussion section.

8.4. Protein-Protein Binding Affinity Estimation

This is the extension of the results of the protein-protein binding affinity estimation benchmark provided in the main text.

We first calculated protein representation vectors using sequences that correspond to PDB structure models in our dataset. Afterwards, we trained a simple regression model to predict the affinity scores in the SKEMPI dataset using 10-fold cross-validation over the binding affinity dataset consisting of 2950 measurements. We benchmarked the 11 learned representation models (3 small-scale & 8 large-scale) and the 4 classic representations. We also compared our results with a state-of-the-art method from the literature, PIPR, which is a protein-protein interaction predictor based on Siamese residual recurrent convolutional neural networks[140]. In the PIPR study, word2vec was employed to generate pre-trained protein representation vectors. Siamese Residual RCNN, Siamese Residual GRU, Siamese CNN, and baseline methods (i.e., autocovariance and composition-transition-distribution) were used to predict the protein-protein binding affinities. In our prediction models, we employed a Bayesian Ridge Regression model[141] for the estimation of real-valued binding affinities. We employed Pearson correlation, mean squared error (MSE) and mean absolute error (MAE) as performance metrics, as these were also used in the PIPR study. Additionally, we calculated statistical significance between the benchmarked representation models using the Wilcoxon Rank Sum Test. Methods from the PIPR study could not be included into the statistical significance test since fold-based results were not provided by that study.

The ProtVec model, another method included in our benchmark, also used word2vec to construct protein representation vectors, similar to PIPR. However, PIPR performed notably better than ProtVec (Fig 6.), confirming the effectiveness of supervised training using Siamese networks in protein-protein binding affinity prediction, compared to directly applying Bayesian Ridge Regression on word2vec-pre-trained representation vectors.

Two methods in our benchmark analysis, namely CTD and KSEP, use amino acid compositions; KSEP incorporates evolutionary information on top of this. The performance results showed that KSEP achieved notably better results compared to CTD. AAC, another amino acid composition-based method in our benchmark, produced a similar performance to CTD. APAAC, a method that utilizes physicochemical properties together with amino acid compositions, had no significant performance advantage over CTD. These results indicate that evolutionary information is especially useful for protein-protein binding affinity prediction.

9. Extended Discussion

Representation learning-based methods often perform better than the classical methods in the functional analysis of proteins

We believe that learned protein representations, in their current state, are also essential for other reasons. First of all, learning-based models produce reusable vectors, which can initially be constructed on unsupervised or semi-supervised tasks using large-scale datasets, and later optimized with further training/fine-tuning (transfer learning) for some other predictive tasks, which may be challenging and/or for which the available data may be scarce, such as predicting the BP GO term or 3-D structure prediction. Second, studies indicate that protein representation learning models can also be employed for designing new proteins using the learned probability distributions of the proteins in the training set[44,142], in the framework of generative modeling. The topic of protein design is discussed further below. Third, the effect of small but critical changes on proteins (i.e., a single amino acid variation in a 500 amino acid protein sequence) is difficult to capture through conventional sequence modeling approaches. However, learned models are sensitive to these effects, as can be seen from the results of our protein-protein binding affinity estimation benchmark. We expect that, with further research, this ability will ultimately have a translational impact in biomedicine by artificial learning of disease mechanisms and proposing new treatment strategies.

Model design and training data type/source are critical factors in representation learning

The first type of source dataset is mostly composed of entire protein databases made up of millions or billions of sequences. In our study, we directly used already trained models provided by the original authors of the various methods, and as a result, their respective choices of source datasets were preserved. The second type of source dataset is used for training/testing (mostly) supervised models to predict the functions, properties, interactions etc. of proteins, which use these learned representation vectors. Our benchmark datasets fall into this second type. In all benchmarking tasks except the protein-protein binding affinity estimation, we utilized the human proteome (or a subset thereof) as our main dataset. This choice offers multiple advantages; *(i)* the human proteome is the main proteome of interest in the field of biomedicine, *(ii)* the functional ontology is richer and the annotation coverage is higher for the human proteome compared to other species, and annotations are more reliable in general, *(iii)* analyzing and interpreting results produced for a single organism is more straightforward, and *(iv)* most of the representation learning methods in the literature have pre-trained models for datasets composed of human proteins. Although we use human proteins, evolutionary information (homology) has also been incorporated into the representation vectors. For example, to construct feature vectors using the k-sep-bigrams method, the target sequence is queried on a database of sequences to find its homologs from other species. Afterwards, multiple sequence alignments of these sequences leverage the evolutionary conservation information. Similarly, multi-species datasets were used in the training of most of the representation learning-based methods we incorporated in our study. With the aim of discussing the ability of different methods in leveraging homology information, we compared the performance results of these representation learning methods against a BLAST-, and an HMMER-based annotation transfer approach, both of which only utilize sequence similarity. If we had incorporated multi-species datasets in the supervised learning phase of our benchmark experiments it would not be possible for us to characterize to what extent homology was captured by pre-trained representations, since the homology relationships could also have been learned during the supervised training.

Potential data leaks should be considered during the construction and evaluation of protein representation learning methods

For example, Gene2Vec utilized a hyperparameter optimization task, which aims to maximize the clustering of genes within MSigDB[64] functional pathways. This task is assumed to contain latent knowledge about BP and CC based protein semantic similarity inference and PFP benchmarks, where Gene2Vec showed a performance above the average score in 3 out of 4 tasks. Likewise, Mut2Vec[17] uses protein-protein interaction data for training, and we found that this model is successful in predicting BP and CC related tasks. It is highly probable that two interacting proteins are localized to the same cellular compartment or have a role in the same biological process. The PFAM representation is directly constructed using Pfam domain annotations. Since protein domains are directly associated with functions, this case can be evaluated as an implicit data leak. Finally, TAPE-BERT-PFAM, one of the best performers especially in the MF-based PFP and drug target protein family classification, is trained on amino acid sequence fragments that correspond to Pfam domain entries. Employing sequence fragments that correspond to protein families in the unsupervised training process (instead of using full protein sequences) may lead to knowledge transfer from Pfam to the protein representation vectors calculated by TAPE-BERT-PFAM. As a summary, we suspect that a data leak might be possible for Gene2Vec, Mut2Vec, PFAM and TAPE-BERT-PFAM in semantic similarity inference and ontology-based PFP benchmarks, and Mut2Vec, PFAM and TAPE-BERT-PFAM in the drug target protein family classification benchmark. In our opinion, most of these cases of knowledge transfer are unlikely to be counted as true examples of data leak from training to test (except in the case of PFAM), since the data and the tasks used in train and test were completely independent, even though the tasks are related to each other. Hence, these protein representation models should be considered successful in terms of inferring relevant information from the input data in the scope of this study. Nonetheless, particulars of such knowledge transfer is an interesting topic to be further investigated in future studies.

The current state and challenges in protein representation learning

Another key challenge is associated with model sizes. In the NLP domain, the number of parameters is steadily increasing with every new high-performance model (e.g., the state-of-the-art GPT-3 model has 175 billion parameters). As most of the successful protein representation learning approaches are based on NLP models, this trend is also observed in the protein representation field[35]. This may pose a critical problem as it increases computational costs to extreme scales, especially for embedding large samples[143] using sequence-based protein representations, where each amino acid in a sequence is modeled as a word in a sentence[77]. As a simple comparison, consider that the average size of a sentence in English is 21.7 words[144]; however, the median number of amino acids in human proteins is 361[145], which makes the problem even more pronounced for protein informatics. There are potential solutions for this issue in the literature[143,146–148], though mainly proposed for NLP-related tasks. These solutions may also be adapted to protein sequence representations. It should also be noted that model sizes (e.g., number of hidden layers, total number of parameters) are not necessarily correlated with performance in protein representation models[77]. We observed this lack of correlation in general in our benchmarks as well. However, there also was a general trend where models classified as "large-scale" performed better than "small-scale" models, at least in some cases (Figures 4, 5, and 6, and Table 2). For example, the SeqVec model has 93M parameters but could compete with much larger models such as ProtT5-XL (3B parameters), in most of the benchmarks. Therefore, constructing larger and more complex models may not always be the way to obtain better representations. Instead, investing time and resources into the incorporation of diverse types of biological data into the models would be a better choice, as smaller models which have larger context seem to be better alternatives and need to be investigated. When smaller models are considered, one important point is to evaluate the total cost, since these models may require a significantly higher number of training cycles (epochs) to achieve the performance provided by larger models.

Protein representation learning methods can be used to design new proteins

Greener *et al.*[149] utilized variational auto-encoders to design metal-binding proteins. In another study, Gupta and Zou[150] show that generative adversarial networks (GANs) could be used for designing proteins through the construction of synthetic encodings of DNA sequences. In the work by Biswas *et al.*, variants of two different proteins (a fluorescent protein and a hydrolase) could successfully be designed with improved functional activity[142]. In another study, Tubiana *et al.* showed that proteins can be designed by defining preferred functions and thus conditioning a Restricted Boltzmann Machine-based protein representation model[44]. Furthermore, it was possible to generate direct 3-D coordinates of full-atom antibody backbones[151], nanobody libraries[152], and to design peptides with anticancer properties[153] (validated by *in vitro* experiments) with deep generative modeling. In the field of drug discovery and development, learned representations have been employed for molecular property prediction[154], drug-target interaction prediction[31] [155] [156] and *de novo* drug design[157].

Supplementary References

1. Uversky, V. N., Gillespie, J. R. & Fink, A. L. Why are “natively unfolded” proteins unstructured under physiologic conditions? *Proteins* 41, 415–427 (2000).

2. Klein, P., Kanehisa, M. & DeLisi, C. The detection and classification of membrane-spanning proteins. *Biochim. Biophys. Acta* 815, 468–476 (1985).

3. Klein, P., Jacquez, J. A. & Delisi, C. Prediction of protein function by discriminant analysis. *Math. Biosci.* 81, 177–189 (1986).

4. Liao, L. & Noble, W. S. Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships. *J. Comput. Biol.* 10, 857–868 (2003).

5. Chen, Z. *et al.* iFeature: a Python package and web server for features extraction and selection from protein and peptide sequences. *Bioinformatics* 34, 2499–2502 (2018).

6. Wang, J. *et al.* POSSUM: a bioinformatics toolkit for generating numerical sequence feature descriptors based on PSSM profiles. *Bioinformatics* 33, 2756–2758 (2017).

7. Jiang, Y. *et al.* An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome Biol.* 17, 184 (2016).

8. Brown, T. B. *et al.* Language Models are Few-Shot Learners. Preprint at https://arxiv.org/abs/2005.14165 (2020).

9. Huang, E. H., Socher, R., Manning, C. D. & Ng, A. Y. Improving word representations via global context and multiple word prototypes. in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* 873–882 (aclweb.org, 2012).

10. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. & Dean, J. Distributed Representations of Words and Phrases and their Compositionality. Preprint at https://arxiv.org/abs/1310.4546 (2013).

11. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Networks* vol. 61 85–117 (2015).

12. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Preprint at https://arxiv.org/abs/1810.04805 (2018).

13. Wan, F. & Zeng, J. (michael). Deep learning with feature embedding for compound-protein interaction prediction. *Bioinformatics* e1004157 (2016).

14. Mejía-Guerra, M. K. & Buckler, E. S. A k-mer grammar analysis to uncover maize regulatory architecture. *BMC Plant Biol.* 19, 103 (2019).

15. Ng, P. dna2vec: Consistent vector representations of variable-length k-mers. Preprint at https://arxiv.org/abs/1701.06279 (2017).

16. Choi, J., Oh, I., Seo, S. & Ahn, J. G2Vec: Distributed gene representations for identification of cancer

prognostic genes. *Sci. Rep.* 8, 13729 (2018).

17. Kim, S., Lee, H., Kim, K. & Kang, J. Mut2Vec: distributed representation of cancerous mutations. *BMC Med. Genomics* 11, 33 (2018).

18. Jaeger, S., Fulle, S. & Turk, S. Mol2vec: Unsupervised Machine Learning Approach with Chemical Intuition. *J. Chem. Inf. Model.* 58, 27–35 (2018).

19. Du, J. *et al.* Gene2vec: distributed representation of genes based on co-expression. *BMC Genomics* 20, 82 (2019).

20. Le, Q. & Mikolov, T. Distributed Representations of Sentences and Documents. in *International Conference on Machine Learning* 1188–1196 (PMLR, 2014).

21. Kimothi, D., Soni, A., Biyani, P. & Hogan, J. M. Distributed Representations for Biological Sequence Analysis. Preprint at https://arxiv.org/abs/1608.05949 (2016).

22. Dutta, A., Dubey, T., Singh, K. K. & Anand, A. SpliceVec: Distributed feature representations for splice junction prediction. *Computational Biology and Chemistry* vol. 74 434–441 (2018).

23. Xu, Y., Song, J., Wilson, C. & Whisstock, J. C. PhosContext2vec: a distributed representation of residue-level sequence contexts and its application to general and kinase-specific phosphorylation site prediction. *Sci. Rep.* 8, 8240 (2018).

24. You, R., Huang, X. & Zhu, S. DeepText2GO: Improving large-scale protein function prediction with deep semantic text representation. *Methods* 145, 82–90 (2018).

25. Yang, K. K., Wu, Z., Bedbrook, C. N. & Arnold, F. H. Learned protein embeddings for machine learning. *Bioinformatics* 34, 2642–2648 (2018).

26. Viehweger, A., Krautwurst, S., Parks, D. H., König, B. & Marz, M. An encoding of genome content for machine learning. *Genomics* 1533 (2019).

27. Kané, H., Coulibali, M., Abdalla, A. & Ajanoh, P. Augmenting protein network embeddings with sequence information. *Bioinformatics* 1080 (2019).

28. Peters, M. E. *et al.* Deep contextualized word representations. Preprint at https://arxiv.org/abs/1802.05365 (2018).

29. Rao, R. *et al.* Evaluating Protein Transfer Learning with TAPE. *Adv. Neural Inf. Process. Syst.* 32, 9689–9701 (2019).

30. Heinzinger, M. *et al.* Modeling the language of life-deep learning protein sequences. bioRxiv. *Bioinformatics* 360, 540 (2019).

31. Öztürk, H., Özgür, A. & Ozkirimli, E. DeepDTA: deep drug-target binding affinity prediction. *Bioinformatics* 34, i821–i829 (2018).

32. Schwartz, A. S. *et al.* Deep Semantic Protein Representation for Annotation, Discovery, and Engineering. *Bioinformatics* D36 (2018).

33. Öztürk, H., Ozkirimli, E. & Özgür, A. WideDTA: prediction of drug-target binding affinity. Preprint at https://arxiv.org/abs/1902.04166 (2019).

34. Asgari, E., Poerner, N., McHardy, A. C. & Mofrad, M. R. K. DeepPrime2Sec: Deep Learning for Protein Secondary Structure Prediction from the Primary Sequences. Preprint at https://www.biorxiv.org/content/10.1101/705426v1 (2019).

35. Rives, A. *et al.* Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. PNAS 118 (15), (2019).

36. Alley, E. C., Khimulya, G., Biswas, S., AlQuraishi, M. & Church, G. M. Unified rational protein engineering with sequence-only deep representation learning. Nature Methods 16, 1315–1322 (2019).

37. Bepler, T. & Berger, B. Learning protein sequence embeddings using information from structure. Preprint at https://arxiv.org/abs/1902.08661 (2019).

38. Strodthoff, N., Wagner, P., Wenzel, M. & Samek, W. UDSMProt: universal deep sequence models for protein classification. *Bioinformatics* 36, 2401–2409 (2020).

39. Vaswani, A. *et al.* Attention is All you Need. in *Advances in Neural Information Processing Systems 30* (eds. Guyon, I. et al.) 5998–6008 (Curran Associates, Inc., 2017).

40. Jain, S. & Wallace, B. C. Attention is not Explanation. Preprint at https://arxiv.org/abs/1902.10186 (2019).

41. Brunner, G. *et al.* On Identifiability in Transformers. Preprint at https://arxiv.org/abs/1908.04211 (2019).

42. Smolensky, P. *Information processing in dynamical systems: Foundations of Harmony theory*. https://apps.dtic.mil/sti/citations/ADA620727 (1986).

43. Hinton, G. E. & Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. *Science* 313, 504–507 (2006).

44. Tubiana, J., Cocco, S. & Monasson, R. Learning protein constitutive motifs from sequence data. *Elife* 8, (2019).

45. Oubounyt, M., Louadi, Z., Tayara, H. & To Chong, K. Deep Learning Models Based on Distributed Feature Representations for Alternative Splicing Prediction. *IEEE Access* 6, 58826–58834 (2018).

46. Mirabello, C. & Wallner, B. rawMSA: End-to-end Deep Learning using raw Multiple Sequence Alignments. *PLoS One* 14, e0220182 (2019).

47. Riesselman, A. J., Ingraham, J. B. & Marks, D. S. Deep generative models of genetic variation capture the effects of mutations. *Nat. Methods* 15, 816–822 (2018).

48. Ding, X., Zou, Z. & Brooks, C. L., Iii. Deciphering protein evolution and fitness landscapes with latent space models. *Nat. Commun.* 10, 5644 (2019).

49. Davidsen, K. *et al.* Deep generative models for T cell receptor protein sequences. *eLife* vol. 8 (2019).

50. UniProt Consortium. UniProt: a hub for protein information. *Nucleic Acids Res.* 43, D204–12 (2015).

51. Mitchell, A. *et al.* The InterPro protein families database: the classification resource after 15 years. *Nucleic Acids Res.* 43, D213–21 (2015).

52. Howe, K. L. *et al.* Ensembl 2021. *Nucleic Acids Res.* 49, D884–D891 (2021).

53. 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 (2019).

54. Buchfink, B., Reuter, K. & Drost, H.-G. Sensitive protein alignments at tree-of-life scale using DIAMOND. *Nat. Methods* 18, 366–368 (2021).

55. Johnson, L. S., Eddy, S. R. & Portugaly, E. Hidden Markov model speed heuristic and iterative HMM search procedure. *BMC Bioinformatics* 11, 431 (2010).

56. Kulmanov, M., Khan, M. A., Hoehndorf, R. & Wren, J. DeepGO: predicting protein functions from sequence and interactions using a deep ontology-aware classifier. *Bioinformatics* 34, 660–668 (2018).

57. Mistry, J. *et al.* Pfam: The protein families database in 2021. *Nucleic Acids Res.* 49, D412–D419 (2021).

58. Madeira, F. *et al.* The EMBL-EBI search and sequence analysis tools APIs in 2019. *Nucleic Acids Res.* 47, W636–W641 (2019).

59. Moal, I. H. & Fernández-Recio, J. SKEMPI: a Structural Kinetic and Energetic database of Mutant Protein Interactions and its use in empirical models. *Bioinformatics* 28, 2600–2607 (2012).

60. Gromiha, M. M. Chapter 2 - Protein Sequence Analysis. in *Protein Bioinformatics* (ed. Gromiha, M. M.) 29–62 (Academic Press, 2010). doi:10.1016/B978-8-1312-2297-3.50002-3.

61. Chen, Z. *et al.* iFeature: a Python package and web server for features extraction and selection from protein and peptide sequences. *Bioinformatics* 34, 2499–2502 (2018).

62. Chou, K.-C. Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes. *Bioinformatics* 21, 10–19 (2005).

63. Miller, G. A. WordNet. *Communications of the ACM* vol. 38 39–41 (1995).

64. Liberzon, A. *et al.* Molecular signatures database (MSigDB) 3.0. *Bioinformatics* 27, 1739–1740 (2011).

65. Clough, E. & Barrett, T. The Gene Expression Omnibus Database. *Methods Mol. Biol.* 1418, 93–110 (2016).

66. Choy, C. T., Wong, C. H. & Chan, S. L. Infer related genes from large scale gene expression dataset with embedding. *Cancer Biology* 2524 (2018).

67. Lu, A. X., Zhang, H., Ghassemi, M. & Moses, A. Self-Supervised Contrastive Learning of Protein Representations By Mutual Information Maximization. Preprint at https://www.biorxiv.org/content/10.1101/2020.09.04.283929v2 (2020)

68. van den Oord, A., Li, Y. & Vinyals, O. Representation Learning with Contrastive Predictive Coding. Preprint at https://arxiv.org/abs/1807.03748 (2018).

69. Krause, B., Lu, L., Murray, I. & Renals, S. Multiplicative LSTM for sequence modelling. Preprint at https://arxiv.org/abs/1609.07959 (2016).

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

71. Andreeva, A., Howorth, D., Chothia, C., Kulesha, E. & Murzin, A. G. SCOP2 prototype: a new approach to protein structure mining. *Nucleic Acids Res.* 42, D310–4 (2014).

72. Mizuguchi, K., Deane, C. M., Blundell, T. L. & Overington, J. P. HOMSTRAD: a database of protein structure alignments for homologous families. *Protein Sci.* 7, 2469–2471 (1998).

73. Raghava, G. P. S., Searle, S. M. J., Audley, P. C., Barber, J. D. & Barton, G. J. OXBench: a benchmark for evaluation of protein multiple sequence alignment accuracy. *BMC Bioinformatics* 4, 47 (2003).

74. Vaswani, A. *et al.* Attention Is All You Need. *arXiv [cs.CL]* (2017).

75. Rao, R., Meier, J., Sercu, T., Ovchinnikov, S. & Rives, A. Transformer protein language models are unsupervised structure learners. Preprint at https://arxiv.org/abs/1706.03762 (2017).

76. Leinonen, R. *et al.* UniProt archive. *Bioinformatics* 20, 3236–3237 (2004).

77. Elnaggar, A. *et al.* ProtTrans: Towards Cracking the Language of Life's Code Through Self-Supervised Deep Learning and High Performance Computing. Preprint at https://arxiv.org/abs/2007.06225 (2020).

78. Klapper-Rybicka, M., Schraudolph, N. N. & Schmidhuber, J. Unsupervised Learning in LSTM Recurrent Neural Networks. *Artificial Neural Networks — ICANN 2001* 684–691 (2001) doi:10.1007/3-540-44668-0_95.

79. He, K., Zhang, X., Ren, S. & Sun, J. Identity Mappings in Deep Residual Networks. in *Computer Vision – ECCV 2016* 630–645 (Springer International Publishing, 2016). doi:10.1007/978-3-319-46493-0_38.

80. Finn, R. D. *et al.* The Pfam protein families database. *Nucleic Acids Res.* 36, D281–8 (2008).

81. Choy, C. T., Wong, C. H. & Chan, S. L. Infer related genes from large scale gene expression dataset with embedding. *Cancer Biology* 2524 (2018).

82. Asgari, E. & Mofrad, M. R. K. Continuous Distributed Representation of Biological Sequences for Deep Proteomics and Genomics. *PLoS One* 10, e0141287 (2015).

83. Melvin, I., Weston, J., Noble, W. S. & Leslie, C. Detecting remote evolutionary relationships among proteins by large-scale semantic embedding. *PLoS Comput. Biol.* 7, e1001047 (2011).

84. Qi, Y., Oja, M., Weston, J. & Noble, W. S. A unified multitask architecture for predicting local protein properties. *PLoS One* 7, e32235 (2012).

85. Collobert, R. & Weston, J. A unified architecture for natural language processing: deep neural networks with multitask learning. in *Proceedings of the 25th international conference on Machine learning* 160–167 (Association for Computing Machinery, 2008). doi:10.1145/1390156.1390177.

86. Asgari, E., McHardy, A. & Mofrad, M. R. K. Probabilistic variable-length segmentation of protein sequences for discriminative motif discovery (DiMotif) and sequence embedding (ProtVecX). *Bioinformatics* 707 (2018).

87. Gage, P. A new algorithm for data compression. *C Users J.* 12, 23–38 (1994).

88. Cortes, C. & Vapnik, V. Support-vector networks. *Mach. Learn.* 20, 273–297 (1995).

89. Hunter, S. *et al.* InterPro: the integrative protein signature database. *Nucleic Acids Res.* 37, D211–5 (2009).

90. Bairoch, A. The ENZYME database in 2000. *Nucleic Acids Res.* 28, 304–305 (2000).

91. The Gene Ontology Consortium & The Gene Ontology Consortium. The Gene Ontology Resource: 20 years and still GOing strong. *Nucleic Acids Research* vol. 47 D330–D338 (2019).

92. Hulo, N. *et al.* The PROSITE database. *Nucleic Acids Res.* 34, D227–30 (2006).

93. Lin, Q., Liang, L., Huang, Y. & Jin, L. Learning to Generate Realistic Scene Chinese Character Images by Multitask Coupled GAN. in *Pattern Recognition and Computer Vision* 41–51 (Springer International Publishing, 2018). doi:10.1007/978-3-030-03338-5_4.

94. Chen, D., Mak, B., Leung, C.-C. & Sivadas, S. Joint acoustic modeling of triphones and trigraphemes by multi-task learning deep neural networks for low-resource speech recognition. in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* 5592–5596 (ieeexplore.ieee.org, 2014). doi:10.1109/ICASSP.2014.6854673.

95. Schulz, C., Eger, S., Daxenberger, J., Kahse, T. & Gurevych, I. Multi-Task Learning for Argumentation Mining in Low-Resource Settings. Preprint at https://arxiv.org/abs/1804.04083 (2018).

96. Kriventseva, E. V. *et al.* OrthoDB v10: sampling the diversity of animal, plant, fungal, protist, bacterial and viral genomes for evolutionary and functional annotations of orthologs. *Nucleic Acids Res.* 47, D807–D811 (2019).

97. Cohen, T., Widdows, D., Heiden, J. A. V., Gupta, N. T. & Kleinstein, S. H. Graded vector representations of immunoglobulins produced in response to west Nile virus. in *Quantum Interaction* (eds. de Barros, J. A., Coecke, B. & Pothos, E.) vol. 10106 135–148 (Springer International Publishing, 2017).

98. Levy, S. D. & Gayler, R. Vector Symbolic Architectures: A New Building Material for Artificial General Intelligence. in *Proceedings of the 2008 conference on Artificial General Intelligence 2008: Proceedings of the First AGI Conference* 414–418 (IOS Press, 2008).

99. You, R. & Zhu, S. DeepText2Go: Improving large-scale protein function prediction with deep semantic text representation. in *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*

42–49 (2017). doi:10.1109/BIBM.2017.8217622.

100. Lindberg, D. A. Internet access to the National Library of Medicine. *Eff. Clin. Pract.* 3, 256–260 (2000).

101. Weininger, D., Weininger, A. & Weininger, J. L. SMILES. 2. Algorithm for generation of unique SMILES notation. *J. Chem. Inf. Comput. Sci.* 29, 97–101 (1989).

102. Figueras, J. Morgan revisited. *J. Chem. Inf. Comput. Sci.* 33, 717–718 (1993).

103. Faisal, M. R. *et al.* Improving Protein Sequence Classification Performance Using Adjacent and Overlapped Segments on Existing Protein Descriptors. *JBiSE* 11, 126–143 (2018).

104. Merity, S., Keskar, N. S. & Socher, R. Regularizing and Optimizing LSTM Language Models. Preprint at https://arxiv.org/abs/1708.02182 (2017)

105. Bileschi, M. L. *et al.* Using Deep Learning to Annotate the Protein Universe. Preprint at https://www.biorxiv.org/content/10.1101/626507v4 (2019).

106. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 770–778 (IEEE, 2016). doi:10.1109/cvpr.2016.90.

107. Hopf, T. A. *et al.* Mutation effects predicted from sequence co-variation. *Nat. Biotechnol.* 35, 128–135 (2017).

108. Wan, F. & Zeng, J. (michael). Deep learning with feature embedding for compound-protein interaction prediction. *Bioinformatics* e1004157 (2016).

109. Mendez, D. *et al.* ChEMBL: towards direct deposition of bioassay data. *Nucleic Acids Res.* 47, D930–D940 (2019).

110. Wishart, D. S. *et al.* DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic Acids Res.* 46, D1074–D1082 (2018).

111. Öztürk, H., Özgür, A. & Ozkirimli, E. DeepDTA: deep drug-target binding affinity prediction. *Bioinformatics* 34, i821–i829 (2018).

112. Yao, Y., Du, X., Diao, Y. & Zhu, H. An integration of deep learning with feature embedding for protein-protein interaction prediction. *PeerJ* 7, e7126 (2019).

113. Zhang, D. & Kabuka, M. Multimodal deep representation learning for protein interaction identification and protein family classification. *BMC Bioinformatics* 20, 531 (2019).

114. Nguyen, S., Li, Z. & Shang, Y. Deep Networks and Continuous Distributed Representation of Protein Sequences for Protein Quality Assessment. in *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)* 527–534 (IEEE, 2017). doi:10.1109/ICTAI.2017.00086.

115. Cuff, J. A. & Barton, G. J. Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. *Proteins* 34, 508–519 (1999).

116. Ramanujan, V., Wortsman, M., Kembhavi, A., Farhadi, A. & Rastegari, M. What's Hidden in a

Randomly Weighted Neural Network? Preprint at https://arxiv.org/abs/1911.13299 (2019).

117. Oubounyt, M., Louadi, Z., Tayara, H. & To Chong, K. Deep Learning Models Based on Distributed Feature Representations for Alternative Splicing Prediction. *IEEE Access* 6, 58826–58834 (2018).

118. Szegedy, C. *et al.* Going deeper with convolutions. in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 1–9 (2015). doi:10.1109/CVPR.2015.7298594.

119. Dutta, A., Dubey, T., Singh, K. K. & Anand, A. SpliceVec: Distributed feature representations for splice junction prediction. *Computational Biology and Chemistry* vol. 74 434–441 (2018).

120. Simmons, M. P. & Ochoterena, H. Gaps as characters in sequence-based phylogenetic analyses. *Syst. Biol.* 49, 369–381 (2000).

121. Bengio, Y., Courville, A. & Vincent, P. Representation learning: a review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 1798–1828 (2013).

122. Chen, R. T. Q., Li, X., Grosse, R. B. & Duvenaud, D. K. Isolating Sources of Disentanglement in Variational Autoencoders. in *Advances in Neural Information Processing Systems 31* (eds. Bengio, S. et al.) 2610–2620 (Curran Associates, Inc., 2018).

123. Achille, A. *et al.* Life-Long Disentangled Representation Learning with Cross-Domain Latent Homologies. in *Advances in Neural Information Processing Systems 31* (eds. Bengio, S. et al.) 9873–9883 (Curran Associates, Inc., 2018).

124. Jain, S., Banner, E., van de Meent, J.-W., Marshall, I. J. & Wallace, B. C. Learning Disentangled Representations of Texts with Application to Biomedical Abstracts. *Proc Conf Empir Methods Nat Lang Process* 2018, 4683–4693 (2018).

125. Andreeva, A., Kulesha, E., Gough, J. & Murzin, A. G. The SCOP database in 2020: expanded classification of representative family and superfamily domains of known protein structures. *Nucleic Acids Res.* 48, D376–D382 (2020).

126. Sillitoe, I. *et al.* CATH: expanding the horizons of structure-based functional annotations for genome sequences. *Nucleic Acids Res.* 47, D280–D284 (2019).

127. Spearman, C. The Proof and Measurement of Association between Two Things. *Am. J. Psychol.* 15, 72–101 (1904).

128. Matthews, B. W. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim. Biophys. Acta* 405, 442–451 (1975).

129. Bookstein, A., Kulyukin, V. A. & Raita, T. Generalized Hamming Distance. *Inf. Retr. Boston.* 5, 353–375 (2002).

130. Raffel, C. *et al.* Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. Preprint at https://arxiv.org/abs/1910.10683 (2019).

131. Ryngajllo, M. *et al.* SLocX: Predicting Subcellular Localization of Arabidopsis Proteins Leveraging Gene Expression Data. *Front. Plant Sci.* 2, 43 (2011).

132. Deng, M., Zhang, K., Mehta, S., Chen, T. & Sun, F. Prediction of protein function using protein-protein interaction data. *J. Comput. Biol.* 10, 947–960 (2003).

133. Coenen, A. *et al.* Visualizing and Measuring the Geometry of BERT. *arXiv [cs.LG]* (2019).

134. Clark, K., Khandelwal, U., Levy, O. & Manning, C. D. What Does BERT Look At? An Analysis of BERT's Attention. Preprint at https://arxiv.org/abs/1906.04341 (2019).

135. Peng, Y., Yan, S. & Lu, Z. Transfer Learning in Biomedical Natural Language Processing: An Evaluation of BERT and ELMo on Ten Benchmarking Datasets. Preprint at https://arxiv.org/abs/1906.05474 (2019).

136. Rifaioglu, A. S., Doğan, T., Jesus Martin, M., Cetin-Atalay, R. & Atalay, V. DEEPred: Automated Protein Function Prediction with Multi-task Feed-forward Deep Neural Networks. *Sci. Rep.* 9, 7344 (2019).

137. Kim, Y., Jernite, Y., Sontag, D. & Rush, A. M. Character-aware neural language models. in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence* 2741–2749 (AAAI Press, 2016).

138. Wilcoxon, F. Individual Comparisons by Ranking Methods. *Biometrics Bulletin* 1, 80–83 (1945).

139. Benjamini, Y. & Hochberg, Y. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *J. R. Stat. Soc. Series B Stat. Methodol.* 57, 289–300 (1995).

140. Chen, M. *et al.* Multifaceted protein-protein interaction prediction based on Siamese residual RCNN. *Bioinformatics* 35, i305–i314 (2019).

141. Tipping, M. E. Sparse Bayesian learning and the relevance vector machine. *J. Mach. Learn. Res.* (2001).

142. Biswas, S., Khimulya, G., Alley, E. C., Esvelt, K. M. & Church, G. M. Low-N protein engineering with data-efficient deep learning. *Nat. Methods* 18, 389–396 (2021).

143. Zafrir, O., Boudoukh, G., Izsak, P. & Wasserblat, M. Q8BERT: Quantized 8Bit BERT. Preprint at https://arxiv.org/abs/1910.06188 (2019).

144. Conneau, A. *et al.* XNLI: Evaluating Cross-lingual Sentence Representations. Preprint at https://arxiv.org/abs/1809.05053(2019).

145. Brocchieri, L. & Karlin, S. Protein length in eukaryotic and prokaryotic proteomes. *Nucleic Acids Res.* 33, 3390–3400 (2005).

146. Sanh, V., Debut, L., Chaumond, J. & Wolf, T. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. Preprint at https://arxiv.org/abs/1910.01108 (2019).

147. Bhargava, P. Adaptive Transformers for Learning Multimodal Representations. Preprint at https://arxiv.org/abs/2005.07486 (2020)

148. Merity, S. Single Headed Attention RNN: Stop Thinking With Your Head. *https://arxiv.org/abs/1911.11423* (2019).

149. Greener, J. G., Moffat, L. & Jones, D. T. Design of metalloproteins and novel protein folds using variational autoencoders. *Sci. Rep.* 8, 16189 (2018).

150. Gupta, A. & Zou, J. Feedback GAN for DNA optimizes protein functions. *Nature Machine Intelligence* 1, 105–111 (2019).

151. Eguchi, R. R., Anand, N., Choe, C. A. & Huang, P.-S. IG-VAE: Generative Modeling of Immunoglobulin Proteins by Direct 3D Coordinate Generation. *Bioinformatics* 29 (2020).

152. Shin, J.-E. *et al.* Protein design and variant prediction using autoregressive generative models. *Nat. Commun.* 12, 2403 (2021).

153. Grisoni, F. *et al.* Designing Anticancer Peptides by Constructive Machine Learning. *ChemMedChem* 13, 1300–1302 (2018).

154. Coley, C. W., Barzilay, R., Green, W. H., Jaakkola, T. S. & Jensen, K. F. Convolutional Embedding of Attributed Molecular Graphs for Physical Property Prediction. *J. Chem. Inf. Model.* 57, 1757–1772 (2017).

155. Rifaioglu, A. S. *et al.* DEEPScreen: high performance drug-target interaction prediction with convolutional neural networks using 2-D structural compound representations. *Chem. Sci.* 11, 2531–2557 (2020).

156. Rifaioglu, A. S. *et al.* MDeePred: novel multi-channel protein featurization for deep learning-based binding affinity prediction in drug discovery. *Bioinformatics* 37, 693–704 (2021).

157. Gómez-Bombarelli, R. *et al.* Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Cent Sci* 4, 268–276 (2018).

158. Lin, D. & Others. An information-theoretic definition of similarity. in *Icml* vol. 98 296–304 (1998).

159. Cozzetto, D., Buchan, D. W. A., Bryson, K. & Jones, D. T. Protein function prediction by massive integration of evolutionary analyses and multiple data sources. *BMC Bioinformatics* 14 Suppl 3, S1 (2013).

160. Lan, L., Djuric, N., Guo, Y. & Vucetic, S. MS-kNN: protein function prediction by integrating multiple data sources. *BMC Bioinformatics* 14 Suppl 3, S8 (2013).

161. Hawkins, T., Chitale, M., Luban, S. & Kihara, D. PFP: Automated prediction of gene ontology functional annotations with confidence scores using protein sequence data. *Proteins* 74, 566–582 (2009).

162. Cao, R. & Cheng, J. Integrated protein function prediction by mining function associations, sequences, and protein-protein and gene-gene interaction networks. *Methods* 93, 84–91 (2016).

163. Piovesan, D., Giollo, M., Leonardi, E., Ferrari, C. & Tosatto, S. C. E. INGA: protein function prediction combining interaction networks, domain assignments and sequence similarity. *Nucleic Acids Res.* 43, W134–40 (2015).

164. Oates, M. E. *et al.* D2P2: database of disordered protein predictions. *Nucleic Acids Res.* 41, D508–

D516 (2012).

165. Youngs, N., Penfold-Brown, D., Drew, K., Shasha, D. & Bonneau, R. Parametric Bayesian priors and better choice of negative examples improve protein function prediction. *Bioinformatics* 29, 1190–1198 (2013).

166. Sasidharan, R., Nepusz, T., Swarbreck, D., Huala, E. & Paccanaro, A. GFam: a platform for automatic annotation of gene families. *Nucleic Acids Res.* 40, e152 (2012).

167. Van Landeghem, S. *et al.* Exploring Biomolecular Literature with EVEX: Connecting Genes through Events, Homology, and Indirect Associations. *Adv. Bioinformatics* 2012, 582765 (2012).

168. Guo, Y., Yu, L., Wen, Z. & Li, M. Using support vector machine combined with auto covariance to predict protein-protein interactions from protein sequences. *Nucleic Acids Res.* 36, 3025–3030 (2008).

169. Yang, L., Xia, J.-F. & Gui, J. Prediction of protein-protein interactions from protein sequence using local descriptors. *Protein Pept. Lett.* 17, 1085–1090 (2010).