nature machine intelligence

ANALYSIS

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

# Learning functional properties of proteins with language models

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

**Data-centric approaches have been used to develop predictive methods for elucidating uncharacterized properties of proteins; however, studies indicate that these methods should be further improved to effectively solve critical problems in biomedicine and biotechnology, which can be achieved by better representing the data at hand. Novel data representation approaches mostly take inspiration from language models that have yielded ground-breaking improvements in natural language processing. Lately, these approaches have been applied to the field of protein science and have displayed highly promising results in terms of extracting complex sequence–structure–function relationships. In this study we conducted a detailed investigation over protein representation learning by first categorizing/explaining each approach, subsequently benchmarking their performances on predicting: (1) semantic similarities between proteins, (2) ontology-based protein functions, (3) drug target protein families and (4) protein–protein binding affinity changes following mutations. We evaluate and discuss the advantages and disadvantages of each method over the benchmark results, source datasets and algorithms used, in comparison with classical model-driven approaches. Finally, we discuss current challenges and suggest future directions. We believe that the conclusions of this study will help researchers to apply machine/deep learning-based representation techniques to protein data for various predictive tasks, and inspire the development of novel methods.**

Protein science is a broad discipline that analyses both individual proteins as well as whole proteomes of organisms via laboratory experiments (that is, proteomics) and computational approaches (for example, molecular modelling, machine learning, data science) to ultimately create accurate and reusable methods for use in biomedicine and biotechnology. Protein informatics can be defined as the computational and data-centric branch of protein science through which the quantitative aspects of proteins are modelled.

The functional characterization of proteins is critical for developing new and effective biomedical strategies and biotechnological products. As of May 2021, there are around 215 million protein entries in the UniProt protein sequence and annotation knowledgebase; however, only 0.56 million (~0.26%) of them have been manually reviewed and annotated by expert curators, indicating a large gap between the current sequencing (data production) and annotation (labelling) capabilities. This gap is mainly due to the cost and time intensive nature of obtaining results from wet-lab experiments and the manual curation thereof. To supplement experimental and curation-based annotation, in silico approaches are being used. In this context, many research groups have been working on developing new computational methods to predict proteins' enzymatic activities[1–3], biophysical properties[4–6], protein and ligand interactions[7–11], three-dimensional structures[12–14] and, ultimately, their functions[15–17]. Protein function prediction (PFP) can be defined as the assignment of functional definitions to proteins, automatically or semi-automatically. The primary terminology for the functions of biomolecules is codified in the Gene Ontology (GO) system, a hierarchical network of concepts (that is, a controlled vocabulary) that annotates the molecular functions of genes and proteins, as well as their subcellular localizations and the biological processes in which they are involved[18]. The most comprehensive benchmarking project for PFP is the Critical Assessment of Functional Annotation (CAFA) challenge[19], in which participants predict GO-based functional associations for a set of target proteins, functions of which are later identified by manual curation, to be used in the assessment of the performance of participating predictors; CAFA challenges so far indicate that PFP is still an open problem.

It has been shown in literature that complex computational problems, where features are high dimensional and have complex/non-linear relationships, are amenable to deep learning-based techniques[20]. These techniques can efficiently learn task-related representations from noisy and high-dimensional input data. Deep learning has thus been successfully applied to various domains such as computer vision, natural language processing and the life sciences[21–24]. Features of biomolecules (for example, genes, proteins, RNAs and so on) should be extracted and encoded as quantitative/numerical vectors (that is, representations) to be used in machine/deep learning-based predictive modelling. Given the raw and high-dimensional input features of a biomolecule, a representation model calculates this feature vector as a succinct and orthogonal representation of that biomolecule. An optimally trained supervised predictive system can efficiently learn features of samples in the dataset and perform the prediction tasks (for example, DNA binding regions on the sequence, biochemical properties, subcellular localization and so on) using these representations as input.

Protein representation approaches can be grouped into two main categories; (1) classical representations (that is, the model-driven approach), which are generated using predefined rules about properties such as the evolutionary relationships between genes/proteins or the physicochemical properties of amino acids (Supplementary Table 1), and (2) data-driven representations, which are constructed

[1]Cancer Systems Biology Laboratory (KanSiL), Graduate School of Informatics, Middle East Technical University, Ankara, Turkey. [2]Department of Biostatistics and Medical Informatics, Karadeniz Technical University, Trabzon, Turkey. [3]Department of Computer Engineering, Hacettepe University, Ankara, Turkey. [4]Institute of Informatics, Hacettepe University, Ankara, Turkey. ✉e-mail: tuncadogan@gmail.com

using statistical and machine learning algorithms (for example, artificial neural networks) that are trained for predefined tasks such as the prediction of the next amino acid on the sequence (Table 1). Later, the output of the trained model—namely, the representation feature vector—can be used for other protein informatics-related tasks such as function prediction. In this sense, representation learning models leverage the transfer of knowledge from one task to another. The generalized form of this process is known as transfer learning[25] and it is reported to be a highly efficient data-analysis approach in terms of time and cost[26]. Hence, protein representation learning models minimize the need for data labelling[27].

Representation learning on proteins is a young but highly active area of research, and is mainly inspired by approaches proposed for natural language processing (NLP). Protein representation learning methods are therefore frequently called protein language models in the literature.

The literature shows that various protein representation learning methods, especially the ones that incorporate deep learning, have been successful at extracting relevant inherent features of proteins (Table 1). Although there are studies that evaluate learned protein representation models[27–29], there is a requirement for a comprehensive survey and benchmark to systematically evaluate these methods in the context of learning multiple aspects of proteins including ontology-based functional definitions, semantic relationships, families and interactions.

In this study we conduct a comprehensive investigation of the available protein representation learning methods that were proposed since 2015, with detailed benchmark analyses measuring the potential of these methods to capture the functional properties of proteins. We cover both classical and artificial learning-based methods and provide insight into their respective approaches to represent proteins. We classify these methods according to their technical features and their applications (Supplementary Section 5). Aiming to evaluate how much each representation model captures different facets of functional information, we constructed and applied benchmarks based on; (1) semantic similarity inference between proteins, (2) ontology-based PFP, (3) drug target protein family classification, (4) protein–protein binding affinity estimation (see the 'Results' section). Finally, we discuss the results and current issues and provide a perspective on the future of learned protein representations (see the 'Discussion' section).

The whole study is schematically summarized in Fig. 1a. Furthermore, we provide the benchmarking software we implemented for this task (Protein Representation Benchmark, PROBE), which allows one to easily evaluate the performance of any representation method over the four benchmarking tasks we defined above, and to compare the results with those reported in this study. We hope that the discussion and conclusions of this study will inform researchers who would like to apply machine/deep learning-based representation techniques on biomolecular data for predictive modelling. Finally, we hope this study will inspire new ideas for the development of novel, sophisticated and robust data-centric approaches to solve open problems in protein science.

## Results

We have selected 23 representation learning methods for our benchmarking tasks (inference of semantic similarities between protein pair, GO-based PFP, drug target protein family classification and protein–protein binding affinity prediction), according to their previously reported success in predictive tasks, and subject to their availability as open access tools or as ready to use pre-constructed feature vectors. Mean pooling is used to aggregate residue features into protein features (see Methods). During the selection process, we also considered the source protein features/attributes used to train these methods (for example, sequence, PPIs and so on) and the algorithmic approaches, with the aim of covering a wide variety of methodologies. The methods included in the benchmark are thus: Learned-Vec[30], SeqVec[31], Mut2Vec[32], Gene2Vec[33], TCGA_EMBEDDING[34], ProtVec[4], TAPE-BERT-PFAM[27], MSA-Transformer[35], CPCProt[36], ProtBERT-BFD[29], UniRep[37], ESM-1b[38], ProtALBERT[29], ProtXLNet[29] and ProtT5-XL[29], along with the classical representations BLAST[39], HMMER[40], PFAM[41], AAC[42], APAAC[43], K-Sep[44], InterPro2GO[45], UniRule2GO[46] and Ensembl-Orthology[47] as baselines. The review of the relevant literature, including the construction and application of protein representations (Fig. 1b), and their technical and application-based classification and evaluation (Supplementary Fig. 15) are given in the Methods and in Supplementary Section 5. A comprehensive summary of 39 protein representation learning methods obtained from the literature, including the above-mentioned benchmark methods, is given in Table 1.

Some of the methods listed above were not applicable to specific benchmark tasks. For example, InterPro2GO[45] (GO_REF: 0000042), UniRule2GO[46] (GO_REF: 0000104), GO projections using Ensembl-Orthology[47] (GO_REF: 0000107) are only suitable for the ontological function prediction task as these methods are not model-based (that is, they do not have feature vectors, only protein–GO term associations). Furthermore, BLAST-[39] and HMMER-based[40] protein sequence similarity feature vectors could not be used in the binding affinity prediction task, as the input sequences in this benchmark are not full protein sequences. Average performances of all methods on all four benchmarks and are summarized in Table 2.

We have plotted the distribution of the GO terms used in our benchmark tasks to confirm visually that they are distributed uniformly over the recorded biomolecular function space, covering nearly all branches in the GO graph (Supplementary Fig. 14), in an effort to show that our GO-based datasets are sufficiently representative.

It is important to note that, protein representation learning methods fall into one of the two categories as protein- or residue-level features, according to the resolution of predicted properties. Our benchmarks (and the methods they test) are mostly in the former category (one partial exception is the estimation of protein–protein binding affinity change following mutations). There are also methods that predict residue level features[6,48,49] (Table 1), and benchmarking studies evaluating these methods[27,28], in the literature.

**Semantic similarity inference.** This analysis aims to measure how much information representation models capture about biomolecular functional similarity. In this context we used GO annotations that represent the molecular functions, large-scale biological roles and subcellular localization of proteins. We first calculated pairwise quantitative similarities between representation vectors of proteins in our dataset using cosine, Manhattan and Euclidean distances/similarities. We then compared these with the ground truth (functional) similarities between these proteins, which are measured on the basis of the actual GO annotations of these proteins using standard semantic similarity measures (for example, Lin similarity[50]). To compare the success of different protein representation methods, we calculated Spearman rank-order correlation values between representation vector similarities and the actual GO-based semantic similarities of the same protein pairs, using three different test datasets (explained in detail in the Methods). The higher the correlation values, the better the success of the representation.

Results based on the Manhattan distance are given in Fig. 2 and Supplementary Fig. 5. Performance results considering the cosine similarity and Euclidean distance measures can be found in Supplementary Figs. 3, 4, 6 and 7, with the statistical significance of correlations indicated with asterisks (* represents a correlation *P*-value between 0.05 and 0.005; **, a correlation *P*-value between 0.005 and 0.00005; ***, a correlation *P*-value equal to or below 0.00005.

**Table 1 | A comprehensive list of protein representation learning methods**

| Method/study name and reference | Learning approach | Depth of the system | Machine learning algorithm | Training input data | Vector size (no. of dim.) | General objective(s) of the system | Specific application(s) of the method | Importance of the study | Data repository |
|---|---|---|---|---|---|---|---|---|---|
| ProtVec[4] | Unsupervised (local) | Shallow | Word2vec | Protein sequences | 100 | Structural feature/ physicochemical feature prediction | Disordered protein/region prediction | First word vector-based protein representation | https://github.com/ehsanasgari/Deep-Proteomics |
| Seq2Vec[5] | Unsupervised (global) | Shallow | Doc2vec | Protein sequences | 250 | Sequence-based feature prediction | Protein sequence classification and retrieval | First Doc2vec-based protein representation | N/A |
| Wan et al.[94] | Supervised (single task) | Shallow | Word2vec (modified for negative examples) | Protein sequences and Morgan fingerprints | 100 | Interaction prediction | Ligand–target protein interaction prediction | Protein representation model for drug–target interaction prediction | N/A |
| ProtVecX[95] | Unsupervised (local) | Shallow | Word2vec | Protein sequences | 500 | Sequence-based feature prediction | Motif discovery, enzyme activity prediction and toxin prediction | Variable length protein sequence representation | https://github.com/ehsanasgari/dimotif |
| DeepDTA[96] | Supervised (single task) | Deep | CNN | Protein and ligand sequences | 128 | Interaction prediction | Ligand–target protein interaction prediction | Unsupervised trained representation for protein ligand binding affinity prediction | https://github.com/hkmztrk/DeepDTA |
| Oubounyt et al.[97] | Unsupervised (global) | Deep | Word2vec, Doc2vec and CNN | Protein sequences | 100 | Genetic feature prediction | Alternative splicing prediction | Use of both Word2vec and Doc2vec for alternative splicing | N/A |
| DeepCon-QA[6] | Unsupervised (global) | Shallow | Word2vec, hidden Markov, CNN | Protein sequences and structures | 200 | Structural feature prediction | Protein quality assessment | Application of protein representations on protein structure model quality assessment | N/A |
| Choy et al.[34] | Unsupervised | Shallow | Artificial neural network | Gene expression profiles (RNAseq) | 50 | Genetic feature prediction | Prediction of immunotherapy responders | Gene expression-based protein representation | https://github.com/zeochoy/tcga-embedding |
| rawMSA[98] | Unsupervised (global) | Deep | CNN–LSTM | Protein sequences | 300 | Structural feature prediction | Secondary structure prediction, relative solvent accessibility prediction and inter-residue contact map prediction | Multiple sequence alignment-based protein representation | https://bitbucket.org/clami66/rawmsa |
| SpliceVec[99] | Unsupervised (global) | Shallow | Word2vec, Doc2vec and multilayered perceptron | Protein sequences | 100 | Genetic feature prediction | Alternative splicing prediction | Unsupervised trained representation for alternative splicing | N/A |
| PhosContext2Vec[49] | Unsupervised (global) | Shallow | Word2vec and Doc2vec | Protein sequences and residue-level features | 126 | Sequence-based feature prediction | Post-translational modification prediction | A protein representation model for phosphorylation site prediction | https://github.com/yxu132/prot2vec_contextualvec |
| Mejía-Guerra et al.[100] | Unsupervised (local) | Shallow | Word2vec | Protein sequences | 300 | Sequence-based feature prediction | Regulatory region prediction | A protein representation model for regulatory region prediction | https://bitbucket.org/bucklerlab/k-mer_grammar |

Continued

**Table 1 | A comprehensive list of protein representation learning methods (Continued)**

| Method/study name and reference | Learning approach | Depth of the system | Machine learning algorithm | Training input data | Vector size (no. of dim.) | General objective(s) of the system | Specific application(s) of the method | Importance of the study | Data repository |
|---|---|---|---|---|---|---|---|---|---|
| Gene2Vec[33] | Unsupervised (local) | Shallow | Word2vec | Gene co-expression profiles | 200 | Sequence-based feature prediction | Gene function prediction | Gene co-expression-based protein representation for gene–gene interaction | https://github.com/jingcheng-du/Gene2vec |
| Yang et al.[30] | Unsupervised (global) | Shallow | Doc2vec | Protein sequences | 64 | Physicochemical feature prediction | Prediction of localization, thermostability, absorption and enantioselectivity | Application of protein representations to predict the functional properties of proteins | https://github.com/fhalab/embeddings_reproduction |
| Cohen et al.[101] | Unsupervised | Shallow | Vector symbolic architectures | Protein sequences and amino acid properties | 1,000 | Sequence-based feature prediction | West Nile virus specific immunoglobulin receptor search | Application of protein representations on immunoglobulin receptor search | N/A |
| Mut2Vec[32] | Unsupervised (local) | Shallow | Word2vec | Gene mutations, biomedical literature, PPIs | 300 | Genetic feature prediction | Classification of driver and passenger mutations | Mutation-based gene representation | http://infos.korea.ac.kr/mut2vec |
| DNA2Vec[102] | Unsupervised (local) | Shallow | Word2vec | Gene sequences | 100 | Genetic feature prediction | Nucleotide sequence similarity search | Variable length DNA sequence representation | https://github.com/pnpnpn/dna2vec |
| Mol2Vec[103] | Unsupervised (local) | Shallow | Word2vec | Morgan substructures | 300 | Sequence-based feature prediction | Kinase activity prediction | Word vector-based molecule representation | https://github.com/samoturk/mol2vec |
| Viehweger et al.[104] | Unsupervised (global) | Shallow | Doc2vec | Protein domains | 100 | Sequence-based feature prediction | Prediction of growth medium and growth temperature of bacteria | Protein domain-based representation in metagenomics | https://github.com/phiweger/nanotext |
| Qi et al.[105] | Supervised (multitask) | Shallow | Feed-forward neural network | Multiple sequence alignments and protein sequences | 35 | Sequence-based feature/structural feature/interaction prediction | Secondary structure, solvent accessibility, DNA binding, signal peptide, PPI, transmembrane topology and coiled coil predictions | Multitask distributed continuous protein representation | N/A |
| ProtEmbed[106] | Supervised (single task) | Shallow | Maximum margin ordinal regression | Protein domain sequences | 250 | Sequence-based feature prediction | Remote homology prediction | Distributed continuous protein representation | N/A |
| G2Vec[107] | Unsupervised (local) | Shallow | Word2vec | Gene expression profiles and PPI | 128 | Genetic feature prediction | Cancer biomarker prediction | Gene expression-based representation for cancer biomarker prediction | https://github.com/mathcom/G2Vec |
| DeepText2GO[108] | Unsupervised (global) | Shallow | TF-IDF and Doc2vec | Biomedical literature and protein sequences | 201 | Sequence-based feature prediction | Protein functional annotation | Text and protein sequence integration for protein representation | N/A |
| WideDTA[62] | Unsupervised (global) | Deep | CNN | Protein and ligand sequences, protein domains, maximum common substructures | 256 | Interaction prediction | Ligand–target protein interaction prediction | Hybrid representation for protein binding affinity prediction | N/A |

Continued

**Table 1 | A comprehensive list of protein representation learning methods (Continued)**

| Method/study name and reference | Learning approach | Depth of the system | Machine learning algorithm | Training input data | Vector size (no. of dim.) | General objective(s) of the system | Specific application(s) of the method | Importance of the study | Data repository |
|---|---|---|---|---|---|---|---|---|---|
| SeqVec[31] | Unsupervised (global) | Deep | LSTM (ELMO) | Protein sequences | 1,024 | Structural feature prediction | Secondary structure prediction and disordered region prediction | Dynamic language model implementation for protein representation | https://github.com/Rostlab/SeqVec |
| UniRep[37] | Unsupervised (global) | Deep | mLSTM | Protein sequences | 5,700 | Sequence-based feature/ structural feature/ physicochemical feature prediction | Secondary structure prediction, protein stability prediction, protein semantic similarity prediction, and protein engineering/ design | Dynamic protein representation to be used for diverse protein related tasks | https://github.com/churchlab/UniRep |
| TAPE[27] | Unsupervised (global) | Deep | LSTM, Transformer and ResNet | Protein sequences | 2,048 (LSTM)<br>100 (ResNet)<br>768 (Transformer) | Sequence-based feature/structural feature prediction | Three-dimensional structure prediction, homology detection, protein engineering/ design | Benchmark framework for protein embeddings | https://github.com/songlab-cal/tape |
| Bepler et al.[109] | Supervised (multitask) | Deep | Bidirectional LSTM | Global structural similarity and pairwise residue contact maps | 100 | Structural feature prediction | Structural similarity search and protein domain prediction | A novel similarity measure between arbitrary-length sequences of vector embeddings based on a soft symmetric alignment | https://github.com/tbepler/protein-sequence-embedding-iclr2019 |
| ESM-1b[38] | Unsupervised (global) | Deep | Transformer (BERT) | Protein sequences | 1,280 | Structural feature/ physicochemical feature prediction | Secondary structure prediction and inter-residue contact map prediction | First bidirectional transformer implementation validated with multiple protein related tasks | N/A |
| D-Space[110] | Supervised (multitask) | Deep | CNN | Protein sequences | 256 | Sequence-based feature prediction | Protein mutagenesis analysis, protein profile search, protein annotation and protein similarity search | Multitask large-scale trained protein representation | https://github.com/syntheticgenomics/sgidspace |
| Tubiana et al.[61] | Unsupervised (global) | Shallow | RBM | Protein sequences | 100 | Structural feature prediction | Protein engineering/ design and inter-residue contact map prediction | RBM-based model | https://github.com/jertubiana/ProteinMotifRBM |
| Kane et al.[111] | Unsupervised (global) | Shallow | Node2vec, OhmNet, Doc2vec | Protein sequences and PPIs | 128 | Sequence-based feature prediction | PFP | Tissue-based function prediction | N/A |
| Faisal et al.[112] | Supervised (multitask) | Shallow | Random Forest and SVM | Protein sequences | 355 | Sequence-based feature prediction | Classification of nuclear receptors, protein family classification, cell penetrating peptide prediction | Use of protein sequence fragments to represent a protein using multiple descriptors | N/A |

Continued

**Table 1 | A comprehensive list of protein representation learning methods (Continued)**

| Method/study name and reference | Learning approach | Depth of the system | Machine learning algorithm | Training input data | Vector size (no. of dim.) | General objective(s) of the system | Specific application(s) of the method | Importance of the study | Data repository |
|---|---|---|---|---|---|---|---|---|---|
| UDSMProt[113] | Supervised (multitask) | Deep | Bidirectional LSTM | Protein sequences | 256 | Sequence-based feature/structural feature prediction | Enzymatic activity prediction, remote homology and fold detection | Application of unsupervised protein representations for small datasets and Enzyme Comission prediction | https://github.com/nstrodt/UDSMProt |
| DeepPrime2Sec[114] | Unsupervised (global) | Deep | Bidirectional LSTM, CNN, ELMO and Word2vec | Protein sequences | 16 to 2,000 (best results with 300) | Structural feature prediction | Secondary structure prediction | Comparison of multiple deep representation learning models for secondary structure prediction | http://llp.berkeley.edu/DeepPrime2Sec |
| CPCProt[36] | Unsupervised (global) | Deep | Contrastive predictive coding | Protein sequences | 512 | Sequence-based feature/structural feature prediction | Structure prediction, homology detection, protein engineering | First model uses contrastive predictive coding for protein representation. | https://github.com/amyxlu/CPCProt |
| ProtTrans (ProtBERT-BFD, ProtXLNet, ProtALBERT, ProtT5-XL)[29] | Unsupervised (global) | Deep | Transformer | Protein sequences | 1,024 (BERT)<br>1,024 (ProtXLNet)<br>4,096 (ProtALBERT)<br>1,024 (ProtT5-XL) | Structural feature/physicochemical feature/sequence-based feature prediction | Secondary structure/subcellular localization prediction,m embrane versus water solubility classification | First comprehensive study that compares large transformer models for protein representation learning | https://github.com/agemagician/ProtTrans |
| ProtCNN[115] | Unsupervised (global) | Deep | CNN | Protein sequences | 1,100 | Protein sequence feature prediction | Protein family prediction | First CNN that uses dilated convolution on protein sequence, trained with the whole Pfam database. | https://github.com/google-research/google-research/tree/master/using_dl_to_annotate_protein_universe |
| MSA-Transformer[35] | Unsupervised (global) | Deep | Transformer | Protein sequences | 768 | Structural feature prediction | Secondary structure prediction, contact prediction. | First transformer-based model that exploits MSAs | https://github.com/facebookresearch/esm |
| DeepSequence[63] | Unsupervised (global) | Deep | Variational autoencoder | Protein sequences | 30 | Sequence-based feature prediction | Mutational effect prediction | First variational autoencoder that exploits MSAs for mutational effect prediction. | https://github.com/debbiemarkslab/DeepSequence |

Vector sizes vary for some of the methods. In such cases we indicate the vector sizes that yield the best predictive performance. LSTM, long short-term memory; CNN, convolutional neural network; RBM, restricted Boltzmann machine; PPI, protein–protein interaction; MSA, multiple sequence alignments; SVM, support vector machine.

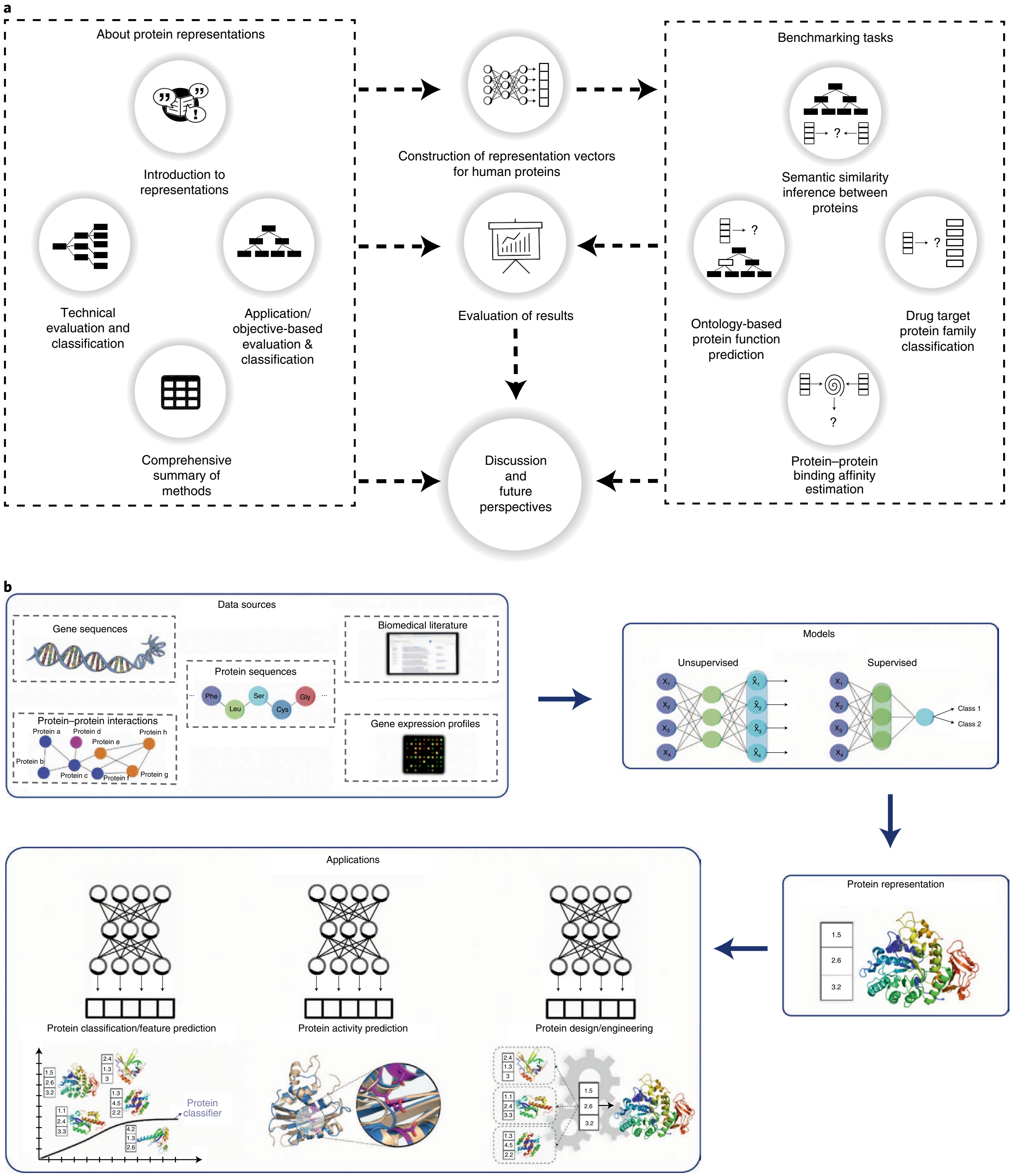
a
About protein representations
Introduction to representations
Technical evaluation and classification
Application/ objective-based evaluation & classification
Comprehensive summary of methods
Construction of representation vectors for human proteins
Evaluation of results
Discussion and future perspectives
Benchmarking tasks
Semantic similarity inference between proteins
Ontology-based protein function prediction
Drug target protein family classification
Protein–protein binding affinity estimation
b
Data sources
Gene sequences
Biomedical literature
Protein sequences
Phe
Leu
Ser
Cys
Gly
Protein–protein interactions
Protein a
Protein b
Protein c
Protein d
Protein e
Protein f
Protein g
Protein h
Gene expression profiles
Models
Unsupervised
Supervised
Class 1
Class 2
Protein representation
1.5
2.6
3.2
Applications
Protein classification/feature prediction
Protein activity prediction
Protein design/engineering
Protein classifier

**Fig. 1 | Schematic representation of the study.** **a**, Overview of the protein representation benchmark study. **b**, Various data sources/types can be used to construct representations and these data can be used to train unsupervised or supervised models, and the output representation vectors can be used for diverse applications.

According to the results presented in Fig. 2a and Supplementary Fig. 5, ProtT5-XL is the most successful representation model in the GO molecular function (MF) category, considering all three datasets. Mut2Vec[32] is the best performer in the GO biological process (BP), TCGA_EMBEDDING and PFAM achieved the highest correlation score in the GO cellular component (CC) category. SeqVec, ProtXLNet, ProtBERT-BFD and Learned-Vec are other notable methods that follow the top performers in these categories. More

**Table 2 | Categorization of the benchmarked representation methods and their respective predictive performance**

| Grouping | General approach used in representation | Specific data source/methodology used in representation | Representation method name | Semantic similarity inference (based on the Manhattan distance) | | | | Ontology-based PFP | | | | Drug target protein family classification | | | | PPI binding affinity estimation |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Spearman correlation | | | | F1-score | | | | MCC (average) | | | | MSE (average) |
| | | | | MF | BP | CC | Ave. | MF | BP | CC | Ave. | Random | 50% | 30% | 15% | |
| Classical representation methods and rule/association-based methods | Homology | Similarity (annotation transfer between similar sequences) | BLAST | 0.20 | 0.14 | 0.05 | 0.13 | 0.87 | 0.56 | 0.57 | 0.67 | 0.85 | 0.83 | 0.81 | 0.68 | NA |
| | | | HMMER | 0.25 | 0.30 | 0.24 | 0.26 | 0.89 | 0.61 | 0.60 | 0.70 | 0.85 | 0.84 | 0.83 | 0.73 | NA |
| | | Functional/structural regions | PFAM | 0.35 | 0.42 | **0.51** | 0.43 | 0.86 | 0.56 | 0.58 | 0.67 | 0.90 | 0.90 | 0.90 | 0.81 | 2.26 |
| | | Transition probability between amino acids | K-Sep | 0.22 | 0.29 | 0.29 | 0.27 | 0.81 | 0.52 | 0.50 | 0.61 | 0.67 | 0.72 | 0.71 | 0.64 | 0.97 |
| | | Annotation transfer between orthologues | Ensembl-Orthology | NA | NA | NA | NA | 0.20 | 0.24 | 0.26 | 0.23 | NA | NA | NA | NA | NA |
| | | Expert curation | UniRule2GO | NA | NA | NA | NA | 0.01 | 0.01 | 0.04 | 0.02 | NA | NA | NA | NA | NA |
| | | | InterPro2GO | NA | NA | NA | NA | 0.37 | 0.11 | 0.27 | 0.25 | NA | NA | NA | NA | NA |
| | Composition | Amino acid composition | AAC | −0.01 | 0.21 | 0.09 | 0.10 | 0.41 | 0.19 | 0.23 | 0.28 | 0.50 | 0.43 | 0.43 | 0.45 | 1.85 |
| | | Amino acid composition and physicochemical properties | APAAC | 0.17 | 0.27 | 0.24 | 0.23 | 0.58 | 0.34 | 0.40 | 0.44 | 0.29 | 0.16 | 0.38 | 0.09 | 1.79 |
| Representation learning methods[a,b] | Automatically learned sequences | Amino acid sequence | ProtVec[a] | 0.19 | 0.30 | 0.21 | 0.23 | 0.64 | 0.36 | 0.38 | 0.46 | 0.34 | 0.31 | 0.39 | 0.37 | 1.13 |
| | | | Learned-Vec[a] | 0.41 | 0.30 | 0.31 | 0.34 | 0.68 | 0.39 | 0.41 | 0.49 | 0.59 | 0.60 | 0.58 | 0.54 | 1.18 |
| | | | UniRep[b] | 0.42 | 0.47 | 0.32 | 0.41 | 0.82 | 0.48 | 0.53 | 0.61 | 0.69 | 0.75 | 0.75 | 0.63 | 0.73 |
| | | | SeqVec[b] | 0.42 | 0.24 | 0.42 | 0.36 | 0.89 | 0.60 | 0.61 | 0.70 | 0.89 | 0.88 | 0.88 | 0.85 | 0.53 |
| | | | MSA-Transformer[b] | 0.38 | 0.31 | 0.30 | 0.33 | 0.67 | 0.47 | 0.50 | 0.55 | 0.67 | 0.72 | 0.73 | 0.63 | 0.91 |
| | | | CPCProt[a] | 0.06 | 0.11 | −0.09 | 0.03 | 0.65 | 0.40 | 0.44 | 0.50 | 0.63 | 0.66 | 0.62 | 0.64 | 0.73 |
| | | | TAPE-BERT-PFAM[b] | 0.50 | 0.21 | 0.22 | 0.31 | 0.85 | 0.54 | 0.58 | 0.65 | 0.77 | 0.79 | 0.76 | 0.73 | 2.35 |
| | | | ProtBERT-BFD[b] | 0.29 | 0.32 | 0.42 | 0.34 | 0.85 | 0.61 | 0.62 | 0.69 | 0.84 | 0.84 | 0.84 | 0.81 | 0.57 |
| | | | ESM-1b[b] | 0.38 | 0.42 | 0.37 | 0.39 | 0.83 | 0.53 | 0.61 | 0.66 | 0.87 | 0.84 | **0.92** | 0.86 | 0.48 |
| | | | ProtXLNet[b] | 0.23 | 0.31 | 0.25 | 0.26 | 0.82 | 0.50 | 0.59 | 0.63 | 0.81 | 0.80 | 0.85 | 0.72 | 0.61 |
| | | | ProtALBERT[b] | 0.22 | 0.37 | 0.32 | 0.30 | 0.89 | 0.63 | 0.64 | 0.72 | **0.92** | 0.91 | **0.92** | 0.88 | **0.42** |
| | | | ProtT5-XL[b] | **0.57** | 0.21 | 0.40 | 0.39 | **0.90** | **0.66** | **0.68** | **0.75** | **0.92** | **0.92** | **0.92** | **0.90** | 0.60 |
| | Others | Mutations, biomedical literature, PPI | Mut2Vec[a] | 0.55 | **0.58** | 0.39 | **0.51** | 0.57 | 0.43 | 0.46 | 0.49 | 0.44 | 0.45 | 0.44 | 0.46 | NA |
| | | Gene expression | TCGA-Embedding[a] | 0.04 | 0.48 | 0.50 | 0.34 | 0.34 | 0.32 | 0.41 | 0.36 | 0.33 | 0.33 | 0.32 | 0.29 | NA |
| | | Gene co-expression | Gene2Vec[a] | 0.18 | 0.41 | 0.36 | 0.31 | 0.53 | 0.44 | 0.50 | 0.49 | 0.33 | 0.32 | 0.34 | 0.27 | NA |
| Mean performances considering all methods | | | | 0.28 | 0.32 | 0.29 | 0.30 | 0.66 | 0.44 | 0.47 | 0.52 | 0.67 | 0.66 | 0.68 | 0.62 | 1.08 |

[a] Small-scale learned representations. [b] Large-scale learned representations. The performance of representation methods on each benchmark (and its subtasks) are shown with average scores. The best performance for each benchmark and subtask is shown in bold. Details can be found in the Results and Methods. NA, method is not included in the benchmark. MCC, Matthew's correlation coefficient. MSE, mean squared error.

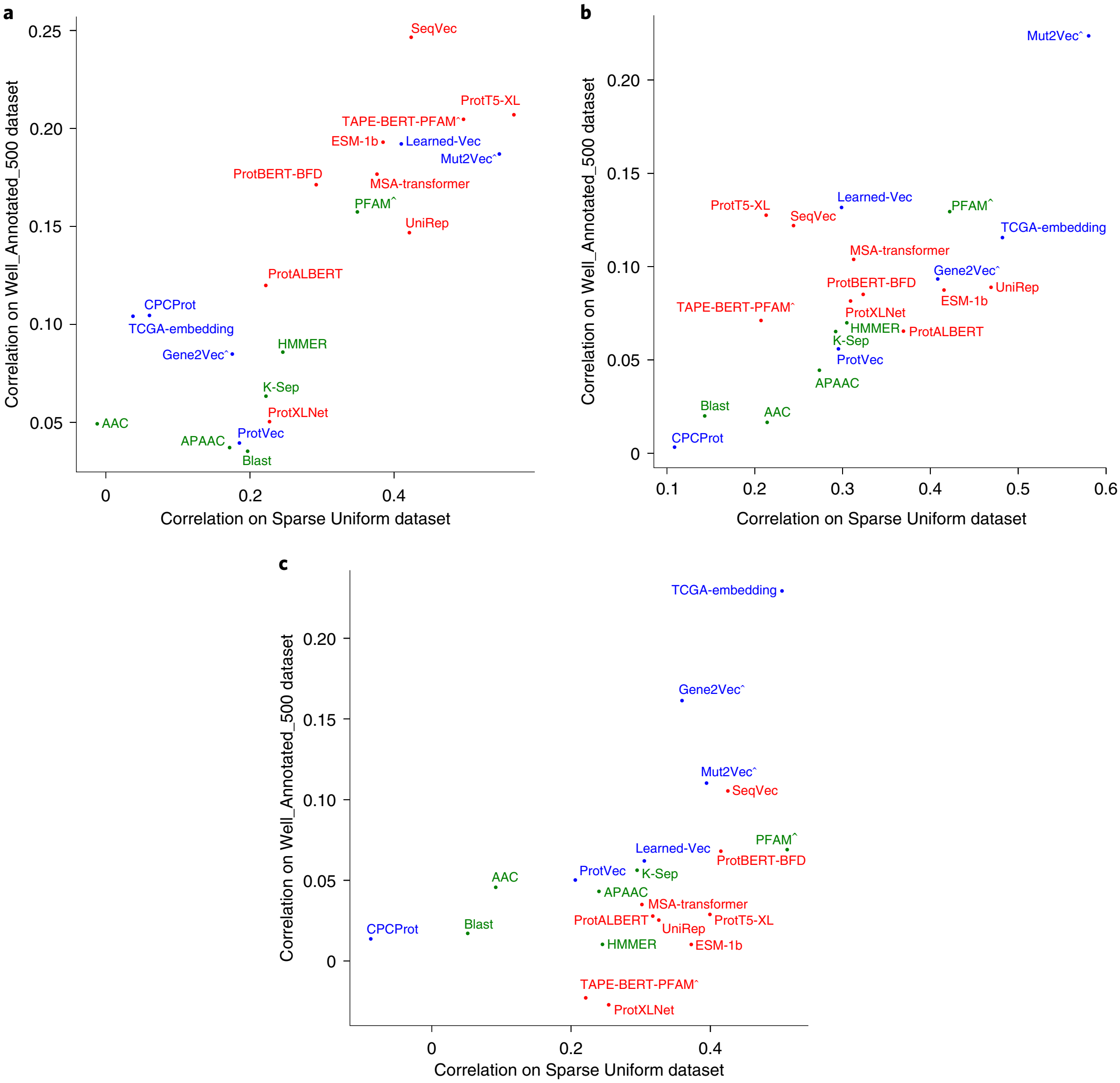

**Fig. 2 | Protein semantic similarity inference benchmark results. a–c**, Performance (Spearman correlation) of protein representation methods in inferring pairwise semantic similarities between proteins considering GO categories of molecular function (Manhattan, **a**), biological process (Manhattan, **b**) and cellular component (Manhattan, **c**). Scatter plots show the performance on Sparse Uniform and Well_Annotated_500 datasets on the *x*- and *y*-axes, respectively. Scores are calculated in terms of Spearman correlation between the ranked true pairwise GO-based semantic similarity list (calculated using Lin similarities[50] between documented GO annotations of proteins with experimental and manual curation evidence codes) and the representation-based ranked pairwise similarity list (calculated using '1 – normalized Manhattan distances' between numerical feature vectors of proteins). Methods with data leak suspicion are marked by ^ symbols. The colours indicate groups of models (green, classical representations; blue, small-scale learned representations; red, large-scale learned representations). See the Methods for details.

information about best performers in this benchmark are given in Supplementary Section 8.1.

In building our benchmark, we initially employed the whole reference human proteome as our test dataset; however, all pairwise combinations between ~20.000 proteins proved to be a sparse comparison space, making differences between the methods tested statistically insignificant. Apart from the dataset, another important parameter in this benchmark was the distance metric. We calculated the performance bed on multiple distance metrics (for example, Cosine, Manhattan and Euclidean). We advise the reader to inspect the results of all four benchmarking tasks over all provided datasets and metrics to reach an unbiased evaluation over representation methods.

**Ontology-based PFP.** As the second benchmark of our study, we aimed to assess the success of representation models in classification-based automated PFP. Here, GO[18] term annotations of proteins were used to train and test the same 23 protein representation models via supervised machine learning-based classification. In this benchmark we preferred a linear classifier (that is, the linear support vector classification with stochastic gradient descent (SGD) optimizer from scikit-learn[51]). We do this to decouple the final performance of the classification from the classifier. If we had used a more sophisticated classifier (for example, kernel SVM, random forest and so on) it would not be possible to tell whether a certain result was due to the power of the representation model or some non-linear transformation performed by the classifier; however, by using a linear boundary classifier we make sure that the representations under test are up to the task of presenting the protein space in a linearly separable fashion. We discussed further details regarding the selection of GO terms under Supplementary Section 8.2.

The PFP performance results are given for the nine GO groups ([low, middle, high] × [shallow, normal, specific]) using F1-score-based heat maps in Fig. 3. The overall GO term prediction performance results (averaged over the nine groups)—in terms of recall, precision, F1-score, accuracy and Hamming distance—are given in Supplementary Table 4. It is important to mention that these performance figures are better than the results reported for the CAFA challenges, due to the way we modelled the experiment. We only run a test sample on the model that contains its true label as one of the five tasks (that is, GO terms), instead of running all test samples on all prediction models. This experimental design choice was made to prevent accumulation of the scores of all benchmarked methods in low-performance regions (especially for hard-to-predict ontologies such as BP), which would prevent clear comparison of the performances. Our aim is to compare the methods with each other from different perspectives within a highly controlled environment, rather than finding the best overall method for PFP, which was the objective of the CAFA challenge. It should also be noted that learned protein representations displayed notable performances in the CAFA challenge[52,53].

It is shown in both Fig. 3 and Supplementary Table 4 that the top methods showed similar performances in the MF prediction task across almost all GO groups (for example, low, high, specific, shallow and so on), among which ProtT5-XL[29] achieved first place and the ProtBERT-BFD[29], SeqVec[31], ProtALBERT[29] and HMMER[40] models ranked next with similar scores. For the BP prediction task, ProtT5-XL was again the best performer and ProtALBERT, SeqVec, ProtBERT-BFD and HMMER were the runners-up. Finally, for the CC prediction task, ProtT5-XL preserved its place as the best performer, and ProtALBERT, SeqVec and HMMER were the runners-up. A detailed discussion on these results is given in Supplementary Section 8.2.

In the PFP benchmark, some of the learned representation models performed considerably better than classical methods, statistically speaking. The overall performances observed in the CC and BP GO term prediction tasks were lower than the MF prediction tasks. This is plausible as most of the learning-based methods use protein sequence data as input, and the sequence is not a direct indicator for localization (as the cleaved signal peptides are absent) or the biological role of the protein in a large-scale process. We also observed that the success rate in CC term prediction decreases with decreasing number of annotated proteins. A similar observation also holds for the MF and BP categories; however, the effect was less pronounced. We did not observe a similar performance delta with increasing or decreasing term specificities (that is, shallow/generic terms versus specific/informative terms). Nevertheless, it is possible to state that there is still an issue regarding the prediction of specific/informative GO terms, as many of them have a low number of annotated proteins.

**Drug target protein family classification.** In our third benchmark analysis, we measured the performance of protein representations in the framework of drug discovery, with the prediction of drug target proteins' main families (that is, enzymes, membrane receptors, transcription factors, ion channels and others), as listed in the ChEMBL database[54]. As these families are made up of proteins with distinct structural characteristics, this benchmark analysis is also expected to reflect the ability of these models in learning structural properties. Furthermore, by using a data source other than functional annotations, we seek to diversify our benchmark and to evaluate the representations from a different perspective. We also incorporated an extra layer of detail to this benchmark by preparing four different versions of the protein family annotation dataset, each filtered in terms of a different predetermined sequence similarity threshold (that is, Random Split dataset, and 50%, 30% and 15% similarity threshold datasets using Uniclust50, Uniclust30 and MMSEQ-15 clustering, respectively) to be used in train/validation dataset splits in the tenfold cross-validation analysis. As a result, no pair of sequences—in which one is in the training and the other in the validation fold—exists that has a sequence similarity of more than the selected threshold (that is, 50%, 30% and 15%) in any case. The similarity-based split dataset statistics are shown in Supplementary Table 11. The aim behind benchmarking methods over these datasets was to inspect how much of the learning is based on simple sequence similarity, as opposed to learning complex and hidden patterns that correspond to the prediction tasks at hand. In this benchmark, we evaluated six small-scale and eight large-scale protein representation learning models, together with six classical representation methods.

According to the mean tenfold cross-validation results of our multitask classification model (Fig. 4 and Table 2), ProtT5-XL and ProtALBERT are the best performers on all datasets. PFAM, ESM-1b and SeqVec models also had remarkable predictive performance. As expected, there is a general trend of decreasing performance as one uses train/test datasets with lower similarity-based split thresholds; however, this decrease is much more evident in classical representations than representation learning methods. For example, BLAST is ranked as the sixth best method on the random split dataset (MCC: 0.85), whereas it ranked eighth, ninth and tenth on the 50%, 30% and 15% similarity-based split datasets (with mean MCCs of 0.83, 0.81 and 0.68), respectively. On the other hand, ProtT5-XL preserved its top performance for nearly all datasets (with mean MCCs of 0.92 for the first three datasets and 0.90 for the 15% split). Other representation learning-based methods such as ESM-1b, SeqVec and ProtBERT-BFD gained ranks from random split to 15% similarity-based split (Fig. 4 and Table 2). The statistical significance of the performance differences is provided in Supplementary Table 8. Protein family specific scores (Supplementary Figs. 8–12) showed that ProtT5-XL provided the best accuracy in the classification of enzymes (Supplementary Fig. 8). ProtT5-XL, ProtALBERT, PFAM, ProtXLNet, ESM-1b, SeqVec, HMMER and BLAST are top representation methods for membrane receptors (Supplementary Fig. 9). For transcription factors, ProtT5-XL, ProtALBERT, ESM-1b and PFAM took top places (Supplementary Fig. 10). For ion channels, ProtT5-XL, BLAST, ProtALBERT, PFAM and ESM-1b are the best scoring models (Supplementary Fig. 11). Finally, ProtALBERT, ProtT5-XL, SeqVec and ESM-1b are the best performers for the others class (Supplementary Fig. 12).

What is interesting here is that when the similarity threshold is dropped to 15%, which is even lower than the so-called twilight zone to transfer structural and functional annotations between proteins (that is, ~25% sequence similarity), top representation learning-based methods still perform very well. These results suggest that representation learning methods may have the ability to capture patterns beyond simple sequence similarities; however, further investigation is required to discuss this topic. ProtT5-XL and ProtALBERT are the best performing models in this benchmark (for example, MCC = 0.92 and 0.91 on the Uniclust50 dataset). Possible underlying reasons for this success are explained in Supplementary Section 8.3.

**Protein–protein binding affinity estimation.** In this benchmark we assessed the performance of representation methods in predicting experimentally identified protein–protein binding affinities. More specifically, the change in binding affinities due to mutations observed in one of the interaction partners is predicted. We used the SKEMPI dataset, which contains PPI binding affinity scores (that is, $K_d$ values) between co-crystalized complexes (from PDB) of both wild-type proteins and variants. The benchmark evaluates representation methods in terms of their ability to extract residue and/or region-level structural features that have critical importance for physical interactions between protein pairs to occur; and how

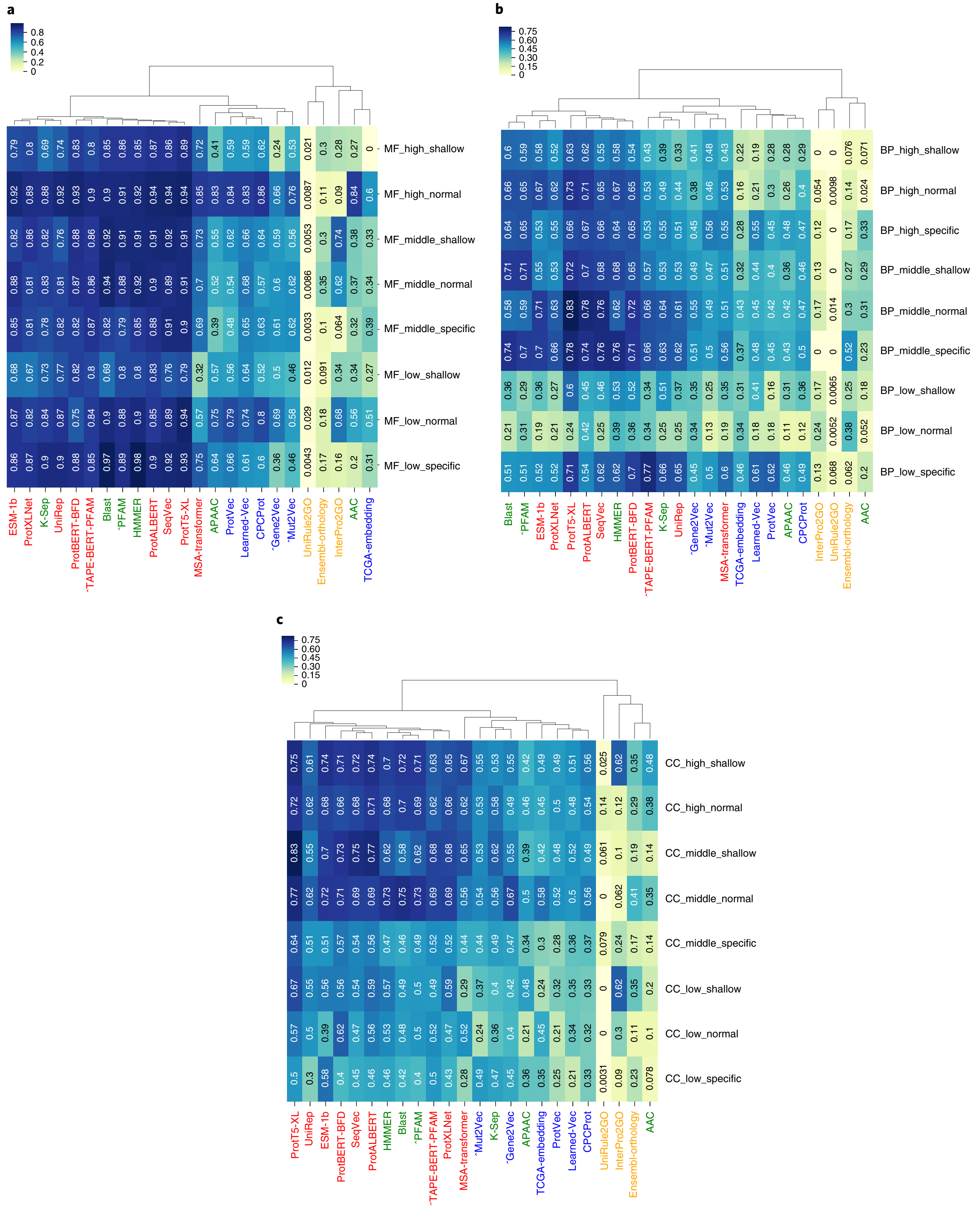

**Fig. 3 | Ontology-based protein function prediction benchmark results. a–c**, Heat maps indicating the clustered performance results (weighted F1-scores) of protein representation methods in ontology-based PFP benchmark in terms of GO categories of molecular function (**a**), biological process (**b**) and cellular component (**c**). The colours indicate groups of models (yellow, rule-based annotation methods; green, classical representations; blue, small-scale learned representations; red, large-scale learned representations). See Methods for details.

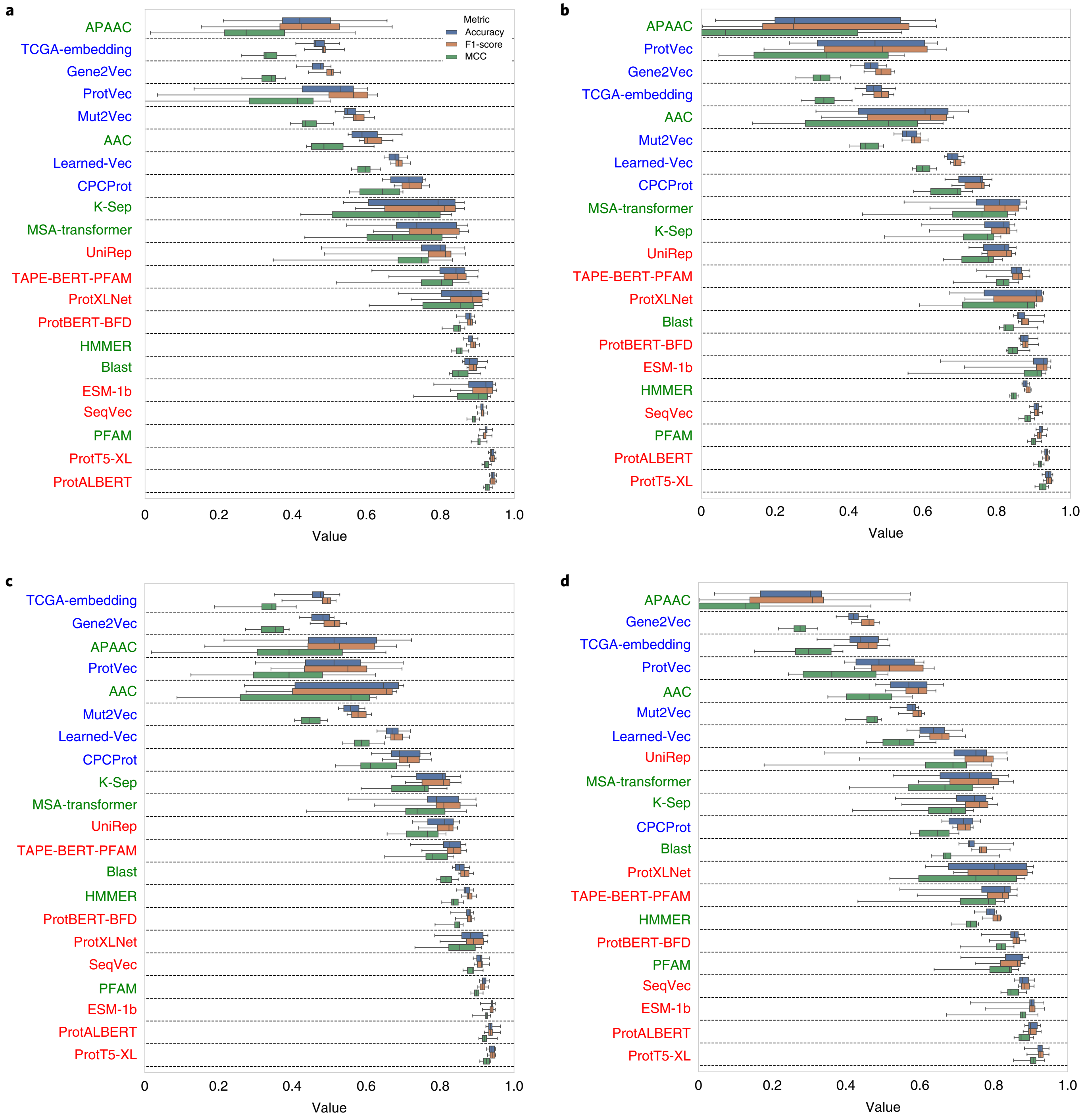

**Fig. 4 | Drug target protein family classification benchmark results. a–d**, Box plots displaying the overall performance results (F1-score, accuracy and MCC) of protein representation methods in the drug target protein family classification benchmark on the Random Split (**a**), UniClust50 (**b**), UniClust30 (**c**) and MMSEQ-15 datasets (**d**). Models are sorted according to mean MCC scores which can be found in Table 2. Colours of names indicate groups of models (green, classical representations; blue, small-scale learned representations; red, large-scale learned representations). See the Methods for details. Whiskers indicate minimum/maximum values.

single, double or triple amino acid changes affect the binding affinities. This subject has significant translational value in terms of understanding the underlying molecular mechanism of many genetic diseases and of proposing new and effective treatments.

Details regarding the dataset, tests, metrics and extended results can be found in Supplementary Section 8.4. Performance scores are given in Fig. 5 and Supplementary Table 9, and the statistical significance of the differences in performance of tested methods are presented in Supplementary Table 10.

According to these results, ProtALBERT produced the best estimations with MSE = 0.43 and MAE = 4.57, correlation = 90.7%. These are approximately 25% better than the results of the baseline PPI prediction based on Siamese residual RCNN (PIPR) model (MSE = 0.63, MAE = 5.48, corr = 87.3%). Please see Supplementary

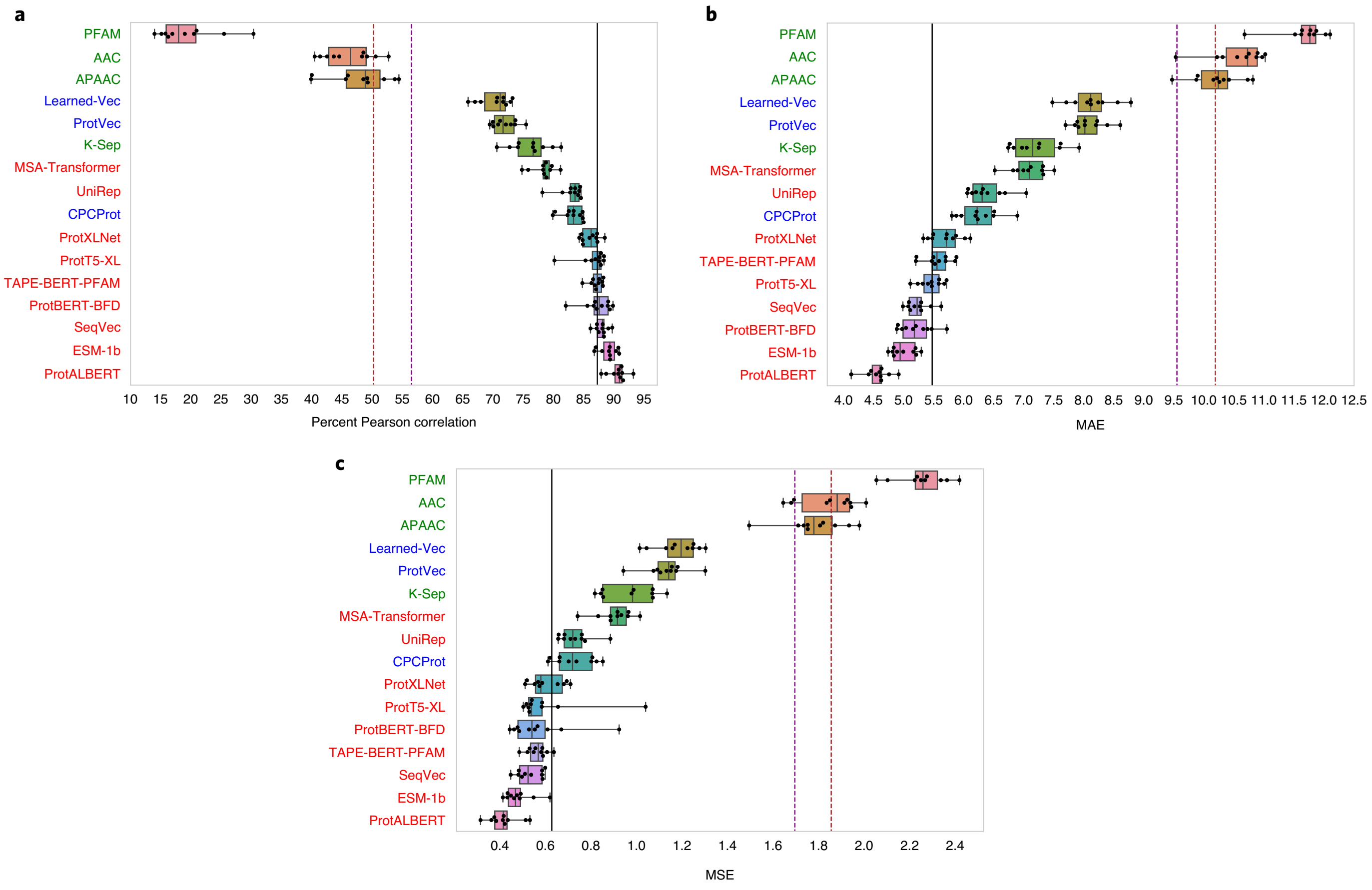

**Fig. 5 | Protein–protein binding affinity estimation benchmark results. a–c**, Box plots indicating performance results of protein representation methods in the protein–protein binding affinity estimation benchmark: percent Pearson correlation values indicating the correlation between predicted values and the true binding affinities (higher values are better) (**a**), MSE values multiplied by $10^2$ (lower values are better) (**b**) and MAE values multiplied by $10^2$ (lower values are better) (**c**). Each dot indicates the performance over a fold in the tenfold cross-validation. Colours of names indicate groups of models (green, classical representations; blue, small-scale learned representations; red, large-scale learned representations). See Methods for details. Black vertical line indicates the best result in the PIPR[92] study. Dashed lines indicate baseline scores from the PIPR study (purple, scores of the model that uses autocovariance feature with Bayesian ridge regression; brown, scores of the model that uses composition-transition-distribution feature with Bayesian ridge regression). Whiskers indicate minimum/maximum values. Each box colour represents the results of a specific model and provided for easy comparison between panels.

Section 8.4 for more information on the baseline models used in this benchmark. Moreover, the ESM-1b and the SeqVec models had performances that surpassed PIPR in all scoring metrics. These results are interesting as PIPR learns input sequences in a supervised framework (in an end-to-end fashion), aiming to maximize the binding affinity prediction performance. In contrast, the protein representations in our benchmark are learned (during pre-training) via tasks (for example, predicting the next amino acid in the sequence) completely unrelated to binding affinity prediction, and then trained in a supervised manner on binding affinity values via simple regression. These results might be explained by the attention mechanism learning the amino acid substitution information. This was also shown in past literature[55]. Moreover, ProtTrans study showed that attention heads may capture the interaction of amino acids[29]. This observation could also explain the best-in-class performance produced by ProtALBERT: this model has a lower number of parameters compared with other transformers in our benchmark (except TAPE-BERT-PFAM), yet the highest number of attention heads. The attention heads are high-dimensional associative data structures that consist of query, key and value variables. When an input (query) is given, attention values are calculated on the basis of the similarity between query and value. These variables are learned during training. In protein representation learning, attention heads learn/discover sequence motifs, which can later be associated with the defined tasks by the remainder of the model[56].

## Discussion

The number of AI-based protein informatics studies has been growing lately to further the understanding of complex relations between sequence, structure and function[57]. In this study we evaluated protein representation learning methods in terms of their ability to capture functional properties of proteins to be used for—and ultimately overcome—the critical challenges in the protein science, biotechnology and biomedicine domains. These models, with their high representation power and modest resource requirements (at least during inference), can be (re)used for a variety of tasks. We therefore argue that learned representations will play an essential role in protein research and development in the near future. Below we discuss critical points relevant to the field of protein representation learning by referring to the results of our benchmarks. A summary of available protein representation learning studies and methods is given in Table 1. The overall performance of selected methods in our four predictive benchmarks can be found in Table 2.

**Representation learning-based methods often perform better than the classical methods in the functional analysis of proteins.** In all of our benchmarks, we observed that the learned representations (particularly the large-scale models) were superior to the classical models in terms of predictive performance, confirming the benefit of the artificial learning-based data-driven approach in representing functional properties of biomolecules. On the other hand, in the molecular function category of the PFP prediction benchmark, HMMER, a classical approach in biomolecular similarity detection and functional annotation that is built on hidden Markov models (HMMs), could compete with deep learning-based protein representation methods. This result is in accordance with previous studies in the sense that sequence similarities are correlated with biochemical properties of proteins to such a high degree that a simple vectorial representation that uses this feature can perform nearly as well as complex sequence modelling methods[19]. In light of these results, we claim that the explicit incorporation of homology information into the training of representation learning models may lead to improvements considering predictive performances. This is also evident from high performance deep learning-based protein structure predictors such as RoseTTAFold[14] and AlphaFold2[13], which use multiple sequence alignments to dramatically enrich the sequence-based input.

We believe that learned protein representations, in their current state, are also essential for other reasons, which are discussed in Supplementary Section 9.

**Model design and training data type/source are critical factors in representation learning.** Our experiments show that one of the most crucial factors in protein representation learning is the design of the representation model. For example, in our benchmark, we included two types of BERT models. The TAPE-BERT-PFAM was trained with 32 million protein domain sequences. ProtBERT-BFD was trained with 2.1 billion metagenomic sequence fragments; however, the performance difference between these two is insignificant (Table 2). On the other hand, more complex models trained with the same 2.1B dataset (such as ProtT5-XL) showed much better performance in most of the benchmarks. Hence, we believe that model design/architecture is of prime importance (information related to the design/architecture of these methods are given in the Methods, and discussed, in relation to predictive performances, in the Results section).

Another finding about training data sources is that incorporating multiple data types may lead to better performance in function-related prediction tasks. As an example, AAC and APAAC both use amino acid composition; however, APAAC also adds physicochemical properties to its representation model and performs significantly better in the semantic similarity inference and PFP benchmarks. Likewise, Mut2Vec incorporates mutation profiles, PPI and text data, and achieved top performance. especially in the semantic similarity inference benchmark.

In the context of our study, it is possible to talk about two entirely different source datasets: the first one is used for the actual representation learning (that is, for training the representation learning model), and the second is for different supervised predictive modelling applications. Details on these datasets are given in Supplementary Section 9.

**Potential data leaks should be considered during the construction and evaluation of protein representation learning methods.** A data leakage can be defined as the accidental leakage of knowledge between the training and validation phases of a machine learning method, leading to overoptimistic performance measurements and is a critical issue that should be considered during performance testing[58]. In our analyses, we observed that certain representation models performed well in tasks that are biologically related to the tasks that these models were pre-trained on; although the data and the actual tasks were different from each other. This discussion is continued in Supplementary Section 9.

**The current state and challenges in protein representation learning.** There are several challenges within the field of protein representation learning. Although most of the protein representation learning models (proposed so far) are derived from NLP models (LSTM/transformer-based deep learning models), there is a structural difference between the problems of modelling language and proteins. In particular, it has been estimated that an adult native American English speaker uses 46,200 lemmas and multiword expressions on average[59]; however, there are only 20 different amino acids in a protein, which are treated in a manner analogous to lemmas of a language by representation models. These NLP models calculate a representation vector for each word. Similarly, when this approach is applied to protein sequence data, a representation vector is calculated for each amino acid. These vectors are pooled to create fixed sized vectors for each sentence/document and protein, for NLP and protein informatics tasks, respectively. Hence, the low number of building blocks in protein representations (that is, 20 amino acids) may pose an advantage for smaller models in competing with larger ones in the protein representation learning domain, in contrast to NLP. Thus, more investigation is encouraged for protein sequence specific learning models. A related key challenge is associated with model sizes which is discussed in detail in Supplementary Section 9.

Model interpretability is critical for understanding why a model behaves the way it does. In an interpretable (that is, explainable) representation, all features are encoded in a segregated form, which means that the feature(s) corresponding to each position on the vector is known; however, most of the learned protein representations investigated in this study are not interpretable/explainable. For example, presence of a TIM barrel structure in a protein might be encoded in the fifth position of its representation vector, whereas the molecular weight information may be shared between the third and fourth positions. In the data science field in general, disentanglement studies try to associate the real properties of samples with individual positions of the output vectors[60]. The disentanglement of protein representations is a new subject, and only a few representation model developers have explored this issue thus far[61,37]. As a result, a systematic approach does not yet exist and new frameworks are required for the standardized evaluation of protein representation model interpretability.

Most of the protein representation models proposed so far are trained using only one type of data (for example, protein sequences). However, protein knowledge is associated with multiple types of biological information, such as PPIs, post-translational modifications, gene/protein (co)expressions and so on. To the best of our knowledge, only a few of the available protein representation models used multiple types of data[32,62]. Among the methods in our benchmark study, Mut2Vec[32,62] was one such example, incorporating PPIs, mutations and biomedical texts, and produced more accurate results than many of the solely sequence-based representations in GO BP- and CC-based PFP. We propose that the integration of additional types of protein related data, especially evolutionary relationships, may further augment the accuracy in predictive tasks. MSA-Transformer[35] and undirected graphical models (for example, DeepSequence[63]) exploit homology information through deep learning. While DeepSequence calculates latent factors using the posterior distribution of MSAs, MSA-Transformer uses row- and column-based attention to combine MSAs and protein language models. Although MSA-Transformer showed average performance in our benchmarks, it was found to be successful on secondary structure and contact prediction tasks in the literature, which suggests MSA-Transformer's ability to capture evolutionary relationships. Related to this, there is a clear requirement in the literature for holistic protein vectors that can effectively represent proteins from a generalized point of view, to be used for various different protein informatics-related purposes. In our opinion, it may be possible to create these holistic representations by concatenating multiple representation vectors that were previously and independently constructed using different types of biological data (as a means of pre-training), and training new models using the integrated version of these vectors for high-level supervised tasks such as predicting biological processes and/or complex structural features (Supplementary Fig. 13). Another way of constructing these holistic representations is directly learning on heterogeneous graphs that integrate multiples types of protein relationships (for example, other proteins, ligands, diseases, phenotypes, functions, pathways and so on)[64] via graph representation learning.

**Protein representation learning methods can be used to design new proteins.** Protein design is one of the key challenges in biotechnology[65]. Rational protein design involves evaluating the activities and functions of many different alternative sequences/structures to provide the most promising candidates for experimental validation, which can be seen as an optimization problem[66]. The sequence space to be explored for this purpose is enormous. For example, the mean length of human proteins is around 350 amino acids, for which $20^{350}$ different combinations exist, even though most of them would be non-functional sequences. In the past couple of decades, computational approaches have been used for protein design, and these have produced promising results particularly in enzyme design[67–69], protein folding and assembly[70] and protein surface design. Efficient antibodies[71] and biosensors[72] have thus been developed. Some of these methods use quantum mechanical calculations[73,74], molecular dynamics[75,76] and statistical mechanics[77,78], each having exceptionally high computational cost[79], and require expert knowledge. Similar shortcomings can also be stated for major protein design software such as Rosetta[80]. Recent studies have shown that artificial learning-based generative modelling can be employed for de novo protein design. In the machine learning domain[81], generative modelling, as opposed to discriminative modelling, is an approach where synthetic samples are produced that obey a probability distribution learned from real samples. This is accomplished by effectively learning the representations of samples in the training dataset. Deep learning has recently become the key approach for generative model architectures[82], and has been applied in various fields including protein/peptide design. For example, Madani et al. used protein language models to design new functional proteins belonging to different protein families from scratch and validated their designs by wet-lab experiments[83]. More examples can be found in Supplementary Section 9. These studies indicate that representation learning is critical for novel applications in both protein and ligand (drug) design.

We believe protein representation learning approaches will have influence on various fields of the protein science with real-world applications in the near future, thanks to their flexibility to integrate heterogeneous protein data (that is, physical and chemical properties/attributes, functional annotations and so on) at the input level, and their ability to efficiently extract complex latent features.

## Methods

In this section, together with relevent sections in the Supplementary Information, we explain different approaches to representing proteins (Supplementary Section 1), classical representation methods (Supplementary Section 2), an evaluation of representation learning approaches from a technical point of view (Supplementary Section 3) and detailed information on representation methods included in our benchmark analyses (Supplementary Section 4).

We group protein representation learning methods' technical approaches (Supplementary Fig. 15a) and objectives/applications reported in their respective publications (Supplementary Fig. 15b) in Supplementary Section 5. Here we formed five main categories according to the application domains: (1) protein interaction prediction (essential for understanding molecular mechanisms and pathways), (2) physicochemical feature prediction (important for protein engineering and drug discovery related tasks), (3) genetic feature prediction, (4) PFP and (5) structural feature prediction. Supplementary Fig. 15b categorizes the main domains and specific application fields under each one. Methods with more than one objective are classified according to their major objective. Common hallmarks possessed by most of the successful protein representations are explained and discussed in Supplementary Section 6.

We present methodological details regarding the datasets, modelling approaches, training/test procedures and performance evaluation for each benchmark task below (metrics are explained in Supplementary Section 7). We share the source code, models and datasets related to this study so that the data can be used by other groups for benchmarking new representation models and to compare the results with those we provide here.

The methods that we included in our benchmark study are Learned-Vec[30], SeqVec[31], Mut2Vec[32], Gene2Vec[33], TCGA_EMBEDDING[34], ProtVec[4], TAPE-BERT-PFAM[27], MSA-Transformer[35], CPCProt[36], ProtBERT-BFD[29], UniRep[37], ESM-1b[38], ProtALBERT[29], ProtXLNet[29], ProtT5-XL[29]. Furthermore, classical representation methods BLAST[39], HMMER[40], AAC[42], APAAC[43], K-Sep[44], PFAM[41] and rule/association-based models, UniRule2GO[46], InterPro2GO[45] and Ensembl-Orthology[47] are employed. All protein representation methods are summarized in terms of their technical aspects (for example, learning approach, algorithm and so on), input data types, vector sizes, objectives, applications, importance and available data repositories in Table 1.

Most of the protein representation learning methods produce outputs as residue features, which means that a separate representation vector is calculated for each amino acid of the protein. Later, residue level features are aggregated to obtain an overall representation for the protein. In our study we chose to use mean pooling for the aggregation procedure, due to its unbiased and conservative structure. It is important to note that the aggregation mechanism is a critical factor affecting model performance and this topic is evaluated with ablation studies in the literature[84].

**Semantic similarity inference benchmark.** To construct the full semantic similarity inference benchmark dataset, we downloaded all human protein entries in the UniProtKB/Swiss-Prot database as well as their GO term annotations from the UniProt-GOA database (2019_11 release). The electronically inferred annotations—labelled with the IEA evidence code—were excluded from the dataset, leaving only the annotations reviewed by human experts. We subsequently enriched the dataset by propagating the annotations to the parent terms of the asserted terms in the GO graph, according to the true path rule. Our finalized full annotation dataset contained 14,625 distinct GO terms (3,374 of them belonged to MF, 9,820 belonged to BP and 1,431 belonged to CC) and 326,009 annotations (75,884 of them belonged to MF, 154,532 belonged to BP and 95,593 belonged to CC).

We calculated the true (that is, ground truth) pairwise GO-based semantic similarities between all proteins in our dataset independently for all GO aspects (that is, MF, BP and CC) using Lin similarity in the GoSemSim package[85]. Lin similarity[50] is based on Shannon's information theory, which states that the information content (IC) of an event is negatively proportional to the observation probability ($P$) of the event; IC is formulated as;

$$\mathrm{IC}\,(P) = \log\,(1/P) \tag{1}$$

Another concept used in Lin similarity is the least common subsumer (LCS), which is the first common ancestor of the two GO terms when travelling to the root in the GO-directed acyclic graph. Lin similarity is thus defined as:

$$\mathrm{sim}_{\mathrm{lin}} = \frac{2\mathrm{IC}\,(\mathrm{LCS}\,(c_1, c_2))}{\mathrm{IC}\,(c_1) + \mathrm{IC}\,(c_2)} \tag{2}$$

More information on semantic similarity measures can be found in the literature[86].

The original/unfiltered semantic similarity dataset included pairwise GO-based semantic similarities between all proteins in our dataset. In this set, 3,077 proteins were used to calculate MF-based pairwise semantic similarities, 6,154 proteins were used for BP-based similarities and 4,531 proteins for CC-based similarities; however, there are numerous poorly annotated proteins, most of which contain insufficient information on their functional properties and might have introduced a bias in the similarity measurements. To mitigate this, we prepared subsets and used these subsets for our analysis. We prepared three semantic similarity subsets (Well_Annotated_500, Well_Annotated_200 and Sparse Uniform) for each GO category (MF, BP and CC), by filtering the semantic similarities in the full dataset. This way, nine datasets were generated in total. The first subset, containing only the top 500 proteins sorted by the number of GO annotations (labelled as well annotated 500 in the relevent figures). The second subset consists only of the top 200 such proteins (labelled as Well_Annotated_200 in the relevent figures). The similarity distribution is not uniform in the three datasets described above, creating very dense similarity score regions (Supplementary Fig. 2) that substantially decrease the correlation values due to rank changes among the pairs with proximal similarities. This caused an accumulation around low correlation values that diminished the discriminative power of the measurements. To prevent this, we sampled every thousandth protein pair from the ranked list of pairwise similarities from the well annotated 500 set to generate a uniformly distributed dataset. This final dataset contains 247 similarity scores between 40 different proteins (labelled as sparse uniform in the relevant figures). Thus, among our three datasets, Sparse Uniform is the most trivial one to predict and Well_Annotated_500 is the most challenging.

In the benchmark phase, we compiled the protein representation vectors for the human protein entries in our dataset using the selected representation learning methods: Learned-Vec[30], SeqVec[31], Mut2Vec[32], Gene2Vec[33], TCGA_EMBEDDING[34], ProtVec[4], TAPE-BERT-PFAM[27], MSA-Transformer[35], CPCProt[36], ProtBERT-BFD[29], UniRep[37], ESM-1b[38], ProtALBERT[29], ProtXLNet[29] and ProtT5-XL[29]. Pre-calculated vectors, when available, were used directly; in other cases these were generated from their respective models. Furthermore, classical representation methods BLAST[39], HMMER[40], AAC[42], APAAC[43], K-Sep[44] and PFAM[41] are included as baselines. We subsequently calculated pairwise similarities between the proteins, using the compiled representation vectors. Cosine similarity, normalized Manhattan distance and normalized Euclidean distance are used to evaluate pairwise similarity (normalized Manhattan and Euclidean distances are converted to similarities by subtracting them from 1).

At this point, we had two sets of pairwise similarity arrays at hand; the first was calculated by taking the GO-derived semantic similarities between the proteins in our dataset into account (that is, the ground truth semantic similarities), and the second consisted of pairwise similarities calculated directly from representation vectors.

Finally, to observe and compare the performance of protein representation models in inferring semantic similarities, we calculated the Spearman rank-order correlation[87] values (as explained in Supplementary Section 7) between the ranked lists of representation vector similarities and true semantic similarities.

**Ontology-based PFP benchmark.** The details of the dataset preparation procedure for the PFP benchmark are explained below in six steps. For each GO category (that is, MF, BP, CC);

1. We obtained human proteins and their GO term annotations from UniProtKB/Swiss-Prot and UniProtGOA databases, respectively (release 2019_10 for both).
2. We excluded all electronically made annotations (evidence code: IEA) from the list of GO term annotations with the aim of increasing the reliability of annotations and to prevent error propagation during prediction.
3. For each GO term, we created an individual list that includes the accessions of the annotated proteins, to be used in model training and testing via cross-validation. We filtered each protein list using the UniRef clusters[88] by only selecting the representative protein entry from each cluster. UniRef provides protein clusters that are formed based on sequence similarity. We used UniRef50 clusters, to ensure that there were no protein sequences with more than 50% sequence similarity in each list. Here the aim is to create train/test datasets without similar proteins that could otherwise introduce a bias to the analysis.
4. GO terms were grouped as either low, middle or high according to the number of annotated proteins. GO terms with 2 to 30 annotated proteins were placed in the low group, terms with 100 to 500 annotated proteins were placed in the middle group and terms with more than 1,000 annotated proteins were placed in the high group. We deliberately left margins between groups to obtain a clear separation.
5. The specificity of the GO terms was determined as either shallow, normal and specific. In the GO graph, terms within the first third of the maximum depth of their respective branches were considered as shallow, terms in the second third were categorized to normal and the deepest third were placed into the specific group. It should be noted that the max depth varies according to the GO category.
6. Based on the combinations of groups constructed in steps 4 and 5; a total of nine GO term groups (3 × 3) were formed for each GO category (MF-low-specific, BP-high-shallow and so on), making a total of 27 groups (9 × 3). There are no GO terms that correspond to two of these groups (for example, MF-high-specific and CC-high-specific) and thus these groups were left out of this analysis. As most of the remaining 25 groups were highly crowded, we selected five terms from each group for further evaluation. Four groups already had less than five GO terms. Hence, they were directly incorporated without further selection. We tried to select dissimilar GO terms in order to generalize the results over the whole functional spectrum, as much as possible. For this, we calculated pairwise semantic similarities between GO terms using Lin similarity, and the five most dissimilar terms were chosen for each group. The statistics of the finalized datasets are given in Supplementary Table 2 and the identifiers of the selected GO terms are given in Supplementary Table 3.

Using these datasets, multitask prediction models were constructed (one for each GO group mostly made up of five GO terms, and for each protein representation method) using linear SVM classification with SGD learning (with Hinge loss) as implemented in the scikit-learn library[51], making a total number of 500 prediction models (25 GO groups × 20 representation methods). This is in addition to the predictions of the three rule/association-based methods (there are no prediction models for these methods as they are not vector-based). Fivefold cross-validation was used to evaluate performance for each model. The default values were selected for the hyperparameters of the SGD classifier (that is, L2 norm for error penalty and the hinge loss function). Due to the simplicity of the linear classification model, we assume that the effect of hyperparameter selection will be minimal.

Rule/association-based models, InterPro2GO[45], UniRule2GO[46] and Ensembl-Orthology[47] were included, in addition to the classical and representation learning-based methods used in the previous (semantic similarity prediction) benchmark. As rule/association-based methods are non-vector-based, their pre-calculated GO annotations were directly obtained from the UniProt database (considering the selected 275 GO terms) and used in the performance evaluation.

**Drug target protein family classification benchmark.** To construct our drug target protein family classification benchmark dataset, we employed the ChEMBL database (v.25)[54], which contains curated collections of drug/compound–target protein interaction data (that is, bioactivities) for experimental and computational research in drug discovery and development. Considering the hierarchical target protein categorization system presented in ChEMBL, we use four broad target protein families and grouped the rest of the targets as a fifth category (that is, enzymes, membrane receptors, transcription factors, ion channels and others). Moreover, we have collected additional human proteins using UniProt's curated keyword annotations (for example, GPCR and ion channel) and UniProt Enzyme Commission number annotations. Furthermore, ChEMBL human drug target single proteins with the family annotations transporter, epigenetic regulator, secreted, other cytosolic, other nuclear, other categories and unclassified are merged as others. Finally, with the aim of collecting the transcription factor family members, the list provided in a highly cited and comprehensive study that catalogues human transcription factors[89] was used. To only include the transcription factors with high confidence, we manually filtered this dataset (that is, eliminated proteins with attributes: TF tested by HT-SELEX? = not tested, CisBP considers it a TF? = no, TFclass considers it a TF? = no, Vaquerizas 2009 classification = no, Motif status = no motif). After an additional manual filtering operation to eliminate proteins with ambiguous or redundant family annotations, we ended up with 4,365, 835, 347, 1,034 and 1,019 proteins for enzymes, GPCRs, ion channels, transcription factors and others, respectively. With these enrichments, we believe that the dataset has become more representative considering the space of known and potential drug targets in the human proteome.

We constructed four different datasets by splitting data into training and test datasets at various degrees of similarity. For this, we used the protein sequence similarity-based clustering scheme UniClust[90] which has pre-calculated sequence clusters at different granulation levels such as 50%, 30%. Moreover, we follow the same protocol with the UniClust study and create another cluster at granulation level 15% for human proteins. We used the MMSeq tool as defined in the UniClust protocol and named our cluster dataset as MMSEQ-15. We separated our train/test datasets at these levels. Namely, for the 50% similarity level, there are no sequence pairs that have a similarity greater than 50% between train and test splits. To yield a fair comparison between the performances on different datasets, we kept the test/validation dataset exactly the same, and discarded the sequences from the training datasets which have a sequence similarity higher than the selected threshold (this operation is repeated independently for each fold of the tenfold cross-validation, for each dataset). The overall number of proteins for each protein family and representation method in the raw/unfiltered dataset (which also corresponds to the random-split dataset) is shown in Supplementary Table 7. The number of proteins per family, cross-validation fold and similarity-based split dataset is provided in Supplementary Table 11. Small differences between the dataset sizes of different representation methods were due to the availability of vectors and are assumed to be negligible (the largest difference was around 3%). These family annotations were used as class labels for the multitask training of the target protein family classification models.

We tested the performance of protein representation methods by training four models (one for the random-split and three for the similarity-based splits) for each representation method and calculated the prediction performances on the respective test datasets. The results show a performance difference between conventional sequence similarity-based methods and novel representation learning-based methods when the similarity threshold is changed from 100% (that is, random split) to 50%, 30% and 15%. We have discussed this in detail in the discussion section. We used the scikit-learn[51] SGD linear-SVM classifier, as in the previous task, with the OneVsRestClassifier mode to handle the multiple classes. The classifier was used with default parameters: hinge loss and L2 norm. The models were trained and tested with tenfold cross-validation.

**Protein–protein binding affinity estimation benchmark.** In this task we benchmarked the protein representation methods in terms of estimating real-valued binding free energies between protein pairs. For this task, we used the structural database of kinetics and energetics of mutant protein interactions (SKEMPI) dataset[91], which gathers experimentally measured mutation-based binding affinity change data on protein–protein heterodimeric complexes from the literature. SKEMPI includes 3,047 equilibrium dissociation constant ($K_D$) measurements for 158 structures belonging to 85 protein–protein complexes (that is, PDB models). Each data point consists of two $K_D$ measurements between a protein pair, one of which is the wild-type version and the other a documented variant (with one or more single amino acid variations). The binding affinity changes following mutations are measured by subtracting the one from the other. During the benchmarking phase, we measured the performance of protein representation methods on directly predicting binding affinity values (including measurements belonging to both wild types and mutated proteins independent from each other) using the 2,950 data points in SKEMPI as our train/test dataset. This is the same dataset used by Chen et al.[92], to whom we compare our results. We obtained the amino acid sequences that correspond to our complex structures from PDB.

For this benchmark, we selected 15 different protein representation learning methods and calculated protein representation vectors for each method using the sequences obtained in the previous step. In particular, Learned-Vec[30], SeqVec[31], Mut2Vec[32], Gene2Vec[33], TCGA_EMBEDDING[34], ProtVec[4], TAPE-BERT-PFAM[27], MSA-Transformer[35], CPCProt[36], ProtBERT-BFD[29], UniRep[37], ESM-1b[38], ProtALBERT[29], ProtXLNet[29] and ProtT5-XL[29] were selected along with classical representation methods AAC[42], APAAC[43], K-Sep[44] and PFAM[41].

We applied element-wise multiplication to the representation vector couples to calculate the input vectors of the estimation model, which associate protein pairs with labels (that is, protein–protein binding affinity values). Bayesian Ridge Regression[93] was used as the binding free energy estimator with tenfold cross-validation.

We compared our results with state-of-the-art methods; Siamese residual RCNN, Siamese residual GRU, Siamese CNN; as well as baseline methods such as autocovariance and composition-transition-distribution. These methods were proposed or employed in the PIPR study[92]. We chose the same estimator and cross-validation strategy as the PIPR study. We also used the same random states as the PIPR study (for determining the samples in each fold) in order to obtain an unbiased comparison. We compared estimation results with the ground truth from the SKEMPI dataset. We used scikit-learn[51] to train the regression model and to calculate validation scores. We used MSE and MAE to measure the performance, the details of which are given in Supplementary Section 7.

## Data availability

All of the datasets and results of this study are available for download at https://github.com/kansil/PROBE. Protein representation and MSA files are available via Zenodo at https://doi.org/10.5281/zenodo.5795850 (ref. [116]).

## Code availability

The source code of this study is available for download at https://github.com/kansil/PROBE. A ready-to-use web-tool containing all models of four benchmarks, to reproduce the results and to test new representation methods on the same predictive tasks are available on the CodeOcean platform, which is reachable from https://PROBE.kansil.org (ref. [117]).

Received: 14 August 2020; Accepted: 8 February 2022;
Published online: 21 March 2022

## References

1. Dalkiran, A. et al. ECPred: a tool for the prediction of the enzymatic functions of protein sequences based on the EC nomenclature. *BMC Bioinf.* **19**, 334 (2018).
2. Dobson, P. D. & Doig, A. J. Distinguishing enzyme structures from non-enzymes without alignments. *J. Mol. Biol.* **330**, 771–783 (2003).
3. Latino, D. A. R. S. & Aires-de-Sousa, J. Assignment of EC numbers to enzymatic reactions with MOLMAP reaction descriptors and random forests. *J. Chem. Inf. Model.* **49**, 1839–1846 (2009).
4. Asgari, E. & Mofrad, M. R. K. Continuous distributed representation of biological sequences for deep proteomics and genomics. *PLoS ONE* **10**, e0141287 (2015).
5. Kimothi, D., Soni, A., Biyani, P. & Hogan, J. M. Distributed representations for biological sequence analysis. Preprint at https://arxiv.org/abs/1608.05949 (2016).
6. 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); https://doi.org/10.1109/ICTAI.2017.00086
7. Keskin, O., Tuncbag, N. & Gursoy, A. Predicting protein–protein interactions from the molecular to the proteome level. *Chem. Rev.* **116**, 4884–4909 (2016).
8. Rifaioglu, A. S. et al. Recent applications of deep learning and machine intelligence on in silico drug discovery: methods, tools and databases. *Briefings Bioinform.* **20**, 1878–1912 (2019).
9. 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).
10. 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).
11. Doğan, T. et al. Protein domain-based prediction of compound–target interactions and experimental validation on LIM kinases. *PLoS Comput. Biol.* **17**, e1009171 (2021).
12. Moult, J., Fidelis, K., Kryshtafovych, A., Schwede, T. & Tramontano, A. Critical assessment of methods of protein structure prediction (CASP)-Round XII. *Proteins* **86**, 7–15 (2018).
13. Tunyasuvunakool, K. et al. Highly accurate protein structure prediction for the human proteome. *Nature* **596**, 590–596 (2021).
14. Baek, M. et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science* **373**, 871–876 (2021).
15. 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).
16. You, R. et al. GOLabeler: improving sequence-based large-scale protein function prediction by learning to rank. *Bioinformatics* **34**, 2465–2473 (2018).
17. Jain, A. & Kihara, D. Phylo-PFP: improved automated protein function prediction using phylogenetic distance of distantly related sequences. *Bioinformatics* **35**, 753–759 (2019).
18. The Gene Ontology Consortium. The gene ontology resource: 20 years and still GOing strong. *Nucleic Acids Res.* **47**, D330–D338 (2019).
19. 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).
20. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
21. Esteva, A. et al. A guide to deep learning in healthcare. *Nat. Med.* **25**, 24–29 (2019).
22. Liu, L. et al. Deep learning for generic object detection: a survey. *Int. J. Comput. Vision* **128**, 261–318 (2020).
23. Zhang, C., Patras, P. & Haddadi, H. Deep learning in mobile and wireless networking: a survey. *IEEE Commun. Surv. Tutor.* **21**, 2224–2287 (2019).
24. Zou, J. et al. A primer on deep learning in genomics. *Nat. Genet.* **51**, 12–18 (2019).
25. Weiss, K., Khoshgoftaar, T. M. & Wang, D. A survey of transfer learning. *J. Big Data* **3**, 1817 (2016).
26. 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).
27. Rao, R. et al. Evaluating protein transfer learning with TAPE. *Adv. Neural Inf. Process. Syst.* **32**, 9689–9701 (2019).
28. Meier, J. et al. Language models enable zero-shot prediction of the effects of mutations on protein function. In *Advances in Neural Information Processing Systems* Vol. 34 (NeurIPS, 2021).
29. 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).
30. Yang, K. K., Wu, Z., Bedbrook, C. N. & Arnold, F. H. Learned protein embeddings for machine learning. *Bioinformatics* **34**, 2642–2648 (2018).
31. Heinzinger, M. et al. Modeling the language of life-deep learning protein sequences. *Bioinformatics* **360**, 540 (2019).
32. Kim, S., Lee, H., Kim, K. & Kang, J. Mut2Vec: distributed representation of cancerous mutations. *BMC Med. Genomics* **11**, 33 (2018).
33. Du, J. et al. Gene2vec: distributed representation of genes based on co-expression. *BMC Genomics* **20**, 82 (2019).
34. Choy, C. T., Wong, C. H. & Chan, S. L. Infer related genes from large scale gene expression dataset with embedding. Preprint at *bioRxiv* https://www.biorxiv.org/content/10.1101/362848v2 (2018).
35. Rao, R. et al. MSA transformer. Preprint at *bioRxiv* https://www.biorxiv.org/content/10.1101/2021.02.12.430858v3 (2021).
36. Lu, A. X., Zhang, H., Ghassemi, M. & Moses, A. Self-supervised contrastive learning of protein representations by mutual information maximization. Preprint at *bioRxiv* https://www.biorxiv.org/content/10.1101/2020.09.04.283929v2 (2020).
37. Alley, E. C., Khimulya, G., Biswas, S., AlQuraishi, M. & Church, G. M. Unified rational protein engineering with sequence-based deep representation learning. *Nat. Methods* **16**, 1315–1322 (2019).
38. Rives, A. et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl Acad. Sci. USA* **118**, e2016239118 (2021).

39. Buchfink, B., Reuter, K. & Drost, H.-G. Sensitive protein alignments at tree-of-life scale using DIAMOND. *Nat. Methods* **18**, 366–368 (2021).
40. Johnson, L. S., Eddy, S. R. & Portugaly, E. Hidden Markov model speed heuristic and iterative HMM search procedure. *BMC Bioinf.* **11**, 431 (2010).
41. Mistry, J. et al. Pfam: the protein families database in 2021. *Nucleic Acids Res.* **49**, D412–D419 (2021).
42. Gromiha, M. M. Protein Sequence Analysis. In *Protein Bioinformatics* (ed. Gromiha, M. M.) Ch. 2, 29–62 (Academic, 2010); https://doi.org/10.1016/B978-8-1312-2297-3.50002-3
43. Chou, K.-C. Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes. *Bioinformatics* **21**, 10–19 (2005).
44. Wang, J. et al. POSSUM: a bioinformatics toolkit for generating numerical sequence feature descriptors based on PSSM profiles. *Bioinformatics* **33**, 2756–2758 (2017).
45. Mitchell, A. et al. The InterPro protein families database: the classification resource after 15 years. *Nucleic Acids Res.* **43**, D213–D221 (2015).
46. UniProt Consortium. UniProt: a hub for protein information. *Nucleic Acids Res.* **43**, D204–D212 (2015).
47. Howe, K. L. et al. Ensembl 2021. *Nucleic Acids Res.* **49**, D884–D891 (2021).
48. Mirabello, C. & Wallner, B. rawMSA: end-to-end deep learning using raw multiple sequence alignments. *PLoS ONE* **14**, e0220182 (2019).
49. 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).
50. Lin, D. & Others. An information-theoretic definition of similarity. In *ICML '98: Proc. 15th International Conference on Machine Learning* 296–304 (ACM, 1998).
51. Pedregosa, F., Varoquaux, G. & Gramfort, A. Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
52. Littmann, M., Heinzinger, M., Dallago, C., Olenyi, T. & Rost, B. Embeddings from deep learning transfer GO annotations beyond homology. *Sci. Rep.* **11**, 1160 (2021).
53. Villegas-Morcillo, A. et al. Unsupervised protein embeddings outperform hand-crafted sequence and structure features at predicting molecular function. *Bioinformatics* **37**, 162–170 (2021).
54. Mendez, D. et al. ChEMBL: towards direct deposition of bioassay data. *Nucleic Acids Res.* **47**, D930–D940 (2019).
55. 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, 2017).
56. Vig, J. et al. BERTology meets biology: interpreting attention in protein language models. Preprint at https://arxiv.org/abs/2006.15222 (2020).
57. Senior, A. W. et al. Improved protein structure prediction using potentials from deep learning. *Nature* **577**, 706–710 (2020).
58. Kaufman, S., Rosset, S., Perlich, C. & Stitelman, O. Leakage in data mining: formulation, detection, and avoidance. *ACM Trans. Knowl. Discov. Data* **6**, 1–21 (2012).
59. Brysbaert, M., Stevens, M., Mandera, P. & Keuleers, E. How many words do we know? Practical estimates of vocabulary size dependent on word definition, the degree of language input and the participant's age. *Front. Psychol.* **7**, 1116 (2016).
60. Higgins, I. et al. Towards a definition of disentangled representations. Preprint at https://arxiv.org/abs/1812.02230 (2018).
61. Tubiana, J., Cocco, S. & Monasson, R. Learning protein constitutive motifs from sequence data. *eLife* **8**, e39397 (2019).
62. Öztürk, H., Ozkirimli, E. & Özgür, A. WideDTA: prediction of drug-target binding affinity. Preprint at https://arxiv.org/abs/1902.04166 (2019).
63. 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).
64. Doğan, T. et al. CROssBAR: Comprehensive resource of biomedical relations with knowledge graph representations. *Nucleic Acids Res.* **49**, e96–e96 (2021).
65. Burk, M. J. & Van Dien, S. Biotechnology for chemical production: challenges and opportunities. *Trends Biotechnol.* **34**, 187–190 (2016).
66. Gainza, P., Nisonoff, H. M. & Donald, B. R. Algorithms for protein design. *Curr. Opin. Struct. Biol.* **39**, 16–26 (2016).
67. Baker, D. An exciting but challenging road ahead for computational enzyme design. *Protein Sci.* **19**, 1817–1819 (2010).
68. Röthlisberger, D. et al. Kemp elimination catalysts by computational enzyme design. *Nature* **453**, 190–195 (2008).
69. Privett, H. K. et al. Iterative approach to computational enzyme design. *Proc. Natl Acad. Sci. USA* **109**, 3790–3795 (2012).
70. Chan, H. S., Shimizu, S. & Kaya, H. Cooperativity principles in protein folding. *Methods Enzymol.* **380**, 350–379 (2004).
71. Lippow, S. M., Wittrup, K. D. & Tidor, B. Computational design of antibody-affinity improvement beyond in vivo maturation. *Nat. Biotechnol.* **25**, 1171–1176 (2007).
72. Looger, L. L., Dwyer, M. A., Smith, J. J. & Hellinga, H. W. Computational design of receptor and sensor proteins with novel functions. *Nature* **423**, 185–190 (2003).
73. Duan, Y. et al. A point-charge force field for molecular mechanics simulations of proteins based on condensed-phase quantum mechanical calculations. *J. Comput. Chem.* **24**, 1999–2012 (2003).
74. Brunk, E. & Rothlisberger, U. Mixed quantum mechanical/molecular mechanical molecular dynamics simulations of biological systems in ground and electronically excited states. *Chem. Rev.* **115**, 6217–6263 (2015).
75. Childers, M. C. & Daggett, V. Insights from molecular dynamics simulations for computational protein design. *Mol. Syst. Des. Eng.* **2**, 9–33 (2017).
76. Hollingsworth, S. A. & Dror, R. O. Molecular dynamics simulation for all. *Neuron* **99**, 1129–1143 (2018).
77. Camilloni, C. & Vendruscolo, M. Statistical mechanics of the denatured state of a protein using replica-averaged metadynamics. *J. Am. Chem. Soc.* **136**, 8982–8991 (2014).
78. Huang, S.-Y. & Zou, X. Statistical mechanics-based method to extract atomic distance-dependent potentials from protein structures. *Proteins* **79**, 2648–2661 (2011).
79. Pierce, N. A. & Winfree, E. Protein design is NP-hard. *Protein Eng.* **15**, 779–782 (2002).
80. Eguchi, R. R., Anand, N., Choe, C. A. & Huang, P.-S. IG-VAE: Generative modeling of immunoglobulin proteins by direct 3D coordinate generation. Preprint at *bioRxiv* https://www.biorxiv.org/content/10.1101/2020.08.07.242347v2 (2020).
81. Ng, A. Y. & Jordan, M. I. On discriminative vs. generative classifiers: a comparison of logistic regression and naive Bayes. In *Advances in Neural Information Processing Systems* (eds. Dietterich, T. G., Becker, S. & Ghahramani, Z.) Vol. 14, 841–848 (MIT Press, 2002).
82. Salakhutdinov, R. Learning deep generative models. *Annu. Rev. Stat. Appl.* **2**, 361–385 (2015).
83. Madani, A. et al. Deep neural language modeling enables functional protein generation across families. Preprint at *bioRxiv* https://www.biorxiv.org/content/10.1101/2021.07.18.452833v1 (2021).
84. Stärk, H., Dallago, C., Heinzinger, M. & Rost, B. Light attention predicts protein location from the language of life. *Bioinformatics Advances* **1**, vbab035 (2021).
85. Yu, G. et al. GOSemSim: an R package for measuring semantic similarity among GO terms and gene products. *Bioinformatics* **26**, 976–978 (2010).
86. McInnes, B. T. & Pedersen, T. Evaluating measures of semantic similarity and relatedness to disambiguate terms in biomedical text. *J. Biomed. Inform.* **46**, 1116–1124 (2013).
87. Spearman, C. The proof and measurement of association between two things. *Am. J. Psychol.* **15**, 72–101 (1904).
88. 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).
89. Lambert, S. A. et al. The human transcription factors. *Cell* **172**, 650–665 (2018).
90. Mirdita, M. et al. Uniclust databases of clustered and deeply annotated protein sequences and alignments. *Nucleic Acids Res.* **45**, D170–D176 (2017).
91. 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).
92. Chen, M. et al. Multifaceted protein–protein interaction prediction based on Siamese residual RCNN. *Bioinformatics* **35**, i305–i314 (2019).
93. Tipping, M. E. Sparse Bayesian learning and the relevance vector machine. *J. Mach. Learn. Res.* **1**, 211–244 (2001).
94. Wan, F. & Zeng, J. (M.). Deep learning with feature embedding for compound–protein interaction prediction. Preprint at *bioRxiv* https://www.biorxiv.org/content/10.1101/086033v1 (2016).
95. Asgari, E., McHardy, A. C. & Mofrad, M. R. Probabilistic variable-length segmentation of protein sequences for discriminative motif discovery (DiMotif) and sequence embedding (ProtVecX). *Sci. Rep.* **9**, 3577 (2019).
96. Öztürk, H., Özgür, A. & Ozkirimli, E. DeepDTA: deep drug-target binding affinity prediction. *Bioinformatics* **34**, i821–i829 (2018).
97. 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).
98. Mirabello, C. & Wallner, B. rawMSA: End-to-end deep learning makes protein sequence profiles and feature extraction obsolete. *Bioinformatics* 228 (2018).
99. Dutta, A., Dubey, T., Singh, K. K. & Anand, A. SpliceVec: distributed feature representations for splice junction prediction. *Comput. Biol. Chem.* **74**, 434–441 (2018).
100. Mejía-Guerra, M. K. & Buckler, E. S. A k-mer grammar analysis to uncover maize regulatory architecture. *BMC Plant Biol.* **19**, 103 (2019).

101. 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.) 135–148 (Springer, 2017).
102. Ng, P. dna2vec: Consistent vector representations of variable-length k-mers. Preprint at https://arxiv.org/abs/1701.06279 (2017).
103. Jaeger, S., Fulle, S. & Turk, S. Mol2vec: Unsupervised machine learning approach with chemical intuition. *J. Chem. Inf. Model.* **58**, 27–35 (2018).
104. Viehweger, A., Krautwurst, S., Parks, D. H., König, B. & Marz, M. An encoding of genome content for machine learning. Preprint at https://www.biorxiv.org/content/10.1101/524280v3 (2019).
105. Qi, Y., Oja, M., Weston, J. & Noble, W. S. A unified multitask architecture for predicting local protein properties. *PLoS ONE* **7**, e32235 (2012).
106. 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).
107. Choi, J., Oh, I., Seo, S. & Ahn, J. G2Vec: distributed gene representations for identification of cancer prognostic genes. *Sci. Rep.* **8**, 13729 (2018).
108. 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 (IEEE, 2017); https://doi.org/10.1109/BIBM.2017.8217622
109. Bepler, T. & Berger, B. Learning protein sequence embeddings using information from structure. Preprint at https://arxiv.org/abs/1902.08661 (2019).
110. Schwartz, A. S. et al. Deep semantic protein representation for annotation, discovery, and engineering. Preprint at *bioRxiv* https://www.biorxiv.org/content/10.1101/365965v1 (2018).
111. Kané, H., Coulibali, M., Abdalla, A. & Ajanoh, P. Augmenting protein network embeddings with sequence information. Preprint at *bioRxiv* https://www.biorxiv.org/content/10.1101/730481v3 (2019).
112. Faisal, M. R. et al. Improving protein sequence classification performance using adjacent and overlapped segments on existing protein descriptors. *JBiSE* **11**, 126–143 (2018).
113. Strodthoff, N., Wagner, P., Wenzel, M. & Samek, W. UDSMProt: universal deep sequence models for protein classification. *Bioinformatics* **36**, 2401–2409 (2020).
114. 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 *bioRxiv* https://www.biorxiv.org/content/early/2019/07/18/705426 (2019)
115. Bileschi, M. L. et al. Using deep learning to annotate the protein universe. *Nat. Biotechnol.* https://doi.org/10.1038/s41587-021-01179-w (2022).
116. Unsal, S. et al. *Learning Functional Properties of Proteins with Language Models Data Sets* (Zenodo, 2020); https://doi.org/10.5281/zenodo.5795850
117. Unsal, S. et al. *PROBE (Protein Representation Benchmark): Function-Centric Evaluation of Protein Representation Methods* (Code Ocean, 2021); https://doi.org/10.24433/CO.5123923.v2

## Acknowledgements

This work was supported by TUBITAK (project no. 318S218). We thank G. Tatar (faculty member, KTU, Turkey) for reading and commenting on the manuscript, and to G.M.Ç. Şılbır (PhD Candidate, KTU, Turkey) for contributing to the drawing of figures.

## Author contributions

T.D., S.U. and A.C.A. conceived the idea and planned the work. S.U. evaluated the literature, constructed representation vectors and prepared the datasets and carried out the analysis for the semantic similarity inference and protein–protein binding affinity estimation benchmarks. H.A. and S.U. prepared the datasets and carried out the analysis for the ontology-based protein family classification benchmark. M.A. and S.U. prepared the datasets and carried out the analysis for the drug target protein family classification benchmark. S.U., A.C.A. and T.D. have written the manuscript. T.D., A.C.A and K.T. supervised the overall study. All authors approved the manuscript.

## Competing interests

The authors declare no competing interests.

## Additional information

**Supplementary information** The online version contains supplementary material available at https://doi.org/10.1038/s42256-022-00457-9.

**Correspondence and requests for materials** should be addressed to Tunca Doğan.

**Peer review information** *Nature Machine Intelligence* thanks Christian Dallago, Céline Marquet and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

**Reprints and permissions information** is available at www.nature.com/reprints.

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.