Computational and Structural Biotechnology Journal 19 (2021) 1750–1758

COMPUTATIONAL AND STRUCTURAL BIOTECHNOLOGY JOURNAL

journal homepage: www.elsevier.com/locate/csbj

The language of proteins: NLP, machine learning & protein sequences

Dan Ofer [a], Nadav Brandes [b,*], Michal Linial [c]

[a] Medtronic, Inc, Israel
[b] The Rachel and Selim Benin School of Computer Science and Engineering, The Hebrew University of Jerusalem, Jerusalem, Israel
[c] Department of Biological Chemistry, Institute of Life Sciences, The Hebrew University of Jerusalem, Jerusalem, Israel

ARTICLE INFO

Article history:
Received 28 January 2021
Received in revised form 19 March 2021
Accepted 19 March 2021
Available online 25 March 2021

Keywords:
Natural language processing
Deep learning
Language models
BERT
Bag of words
Tokenization
Word embedding
Contextualized embedding
Transformer
Artificial neural networks
Word2vec
Bioinformatics

ABSTRACT

Natural language processing (NLP) is a field of computer science concerned with automated text and language analysis. In recent years, following a series of breakthroughs in deep and machine learning, NLP methods have shown overwhelming progress. Here, we review the success, promise and pitfalls of applying NLP algorithms to the study of proteins. Proteins, which can be represented as strings of amino-acid letters, are a natural fit to many NLP methods. We explore the conceptual similarities and differences between proteins and language, and review a range of protein-related tasks amenable to machine learning. We present methods for encoding the information of proteins as text and analyzing it with NLP methods, reviewing classic concepts such as bag-of-words, k-mers/n-grams and text search, as well as modern techniques such as word embedding, contextualized embedding, deep learning and neural language models. In particular, we focus on recent innovations such as masked language modeling, self-supervised learning and attention-based models. Finally, we discuss trends and challenges in the intersection of NLP and protein research.

Contents

* Corresponding author.
E-mail address: nadav.brandes@mail.huji.ac.il (N. Brandes).

https://doi.org/10.1016/j.csbj.2021.03.022

1. Proteins and natural language

Like human language, protein sequences can be naturally represented as strings of letters (Fig. 1A). The protein alphabet consists of 20 common amino acids (AAs) (excluding unconventional and rare amino acids). Furthermore, like natural language, naturally evolved proteins are typically composed of reused modular elements exhibiting slight variations that can be rearranged and assembled in a hierarchical fashion. By this analogy, common protein motifs and domains, which are the basic functional building blocks of proteins, are akin to words, phrases and sentences in human language [96,102,98,93,117].

Another central feature shared by proteins and human language is information completeness. Even though a protein is much more than a mere sequence of amino acids – it is also a three-dimensional machine with a determined structure and function – these other aspects are all predetermined by its amino-acid sequence. While protein structure and function is dynamic and context-dependent (e.g. on cellular state, other molecules and PTMs), it is still defined by the underlying amino-acid sequence. This means that from an information-theory perspective, the protein's information (e.g. its structure) is contained within its sequence [74].

Given these similarities in shape and substance, it seems natural to apply natural language processing (NLP) methods to protein sequences. Although the term NLP refers to natural languages, the same computational methods are also used to study non-natural languages such as programming code [96,49,30]. Past decades have seen a continuous trickle of statistical and machine-learning algorithms from the field of NLP into bioinformatics [67,117,66,6,34,60,88,109,117,105].

It should be kept in mind that the analogies between proteins and human language only go so far. Most importantly, we can read and understand human languages. Additionally, unlike proteins,

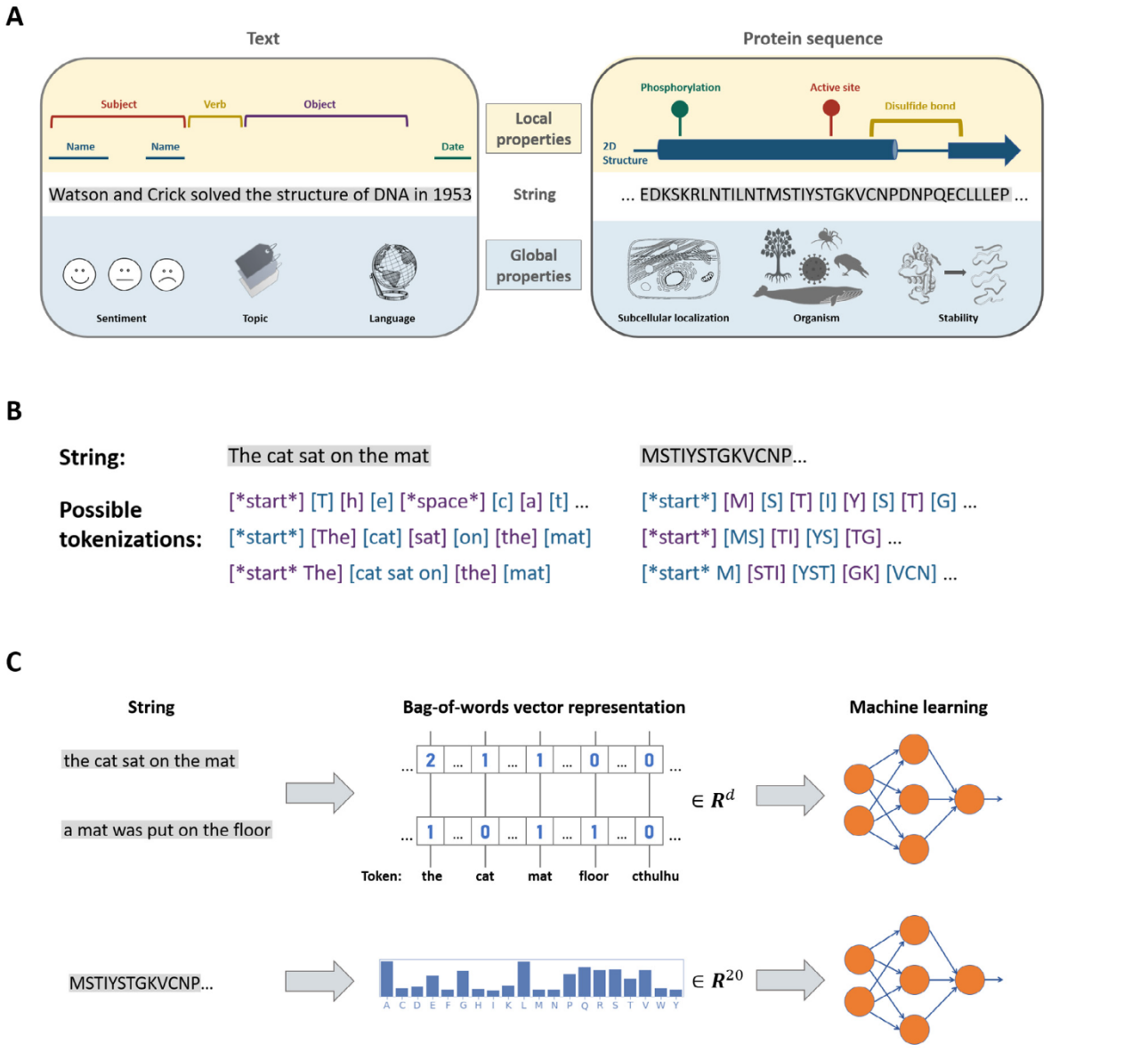

Fig. 1. Computational analysis of natural language and proteins (A) Texts and proteins can be represented as strings of letters and processed with NLP methods to study local and global properties. (B) A common preprocessing step in NLP is the tokenization of text or protein sequences into distinct tokens, which are the atomic units of information. There are many different ways to tokenize text, e.g. as letters, words, or other substring pieces of equal or unequal length. (C) Bag-of-words representation can be used to count unique tokens in a text, turning every input text into a fixed-size vector. Subsequently, these vector representations can be analyzed through any machine-learning algorithm.

most human languages include uniform punctuation and stop words, with clearly separable structures such as words, sentences and paragraphs. With proteins, we do not always know whether a sequence of amino-acids is part of a functional unit (e.g. a domain). There is no clear analogy between the building blocks of language and those of proteins. For example, considering protein domains as being equivalent to words is often misleading. Furthermore, protein functional units often overlap. As a result, while natural languages have a well-defined vocabulary (with ~ million words in English), proteins lack a clear vocabulary. From an information-theory perspective, the entropy of sequences within protein domains was shown to be lower than the English language, while still being significantly different from a random distribution [117]. Proteins also exhibit high variability in length, which in human ranges over three orders of magnitudes (from less than 20 AAs for peptide hormones to tens of thousands of AAs in some structural proteins). Such a wide range of protein sequence lengths is prevalent in all domains of life, from viruses to humans [17]. In NLP, specific words might have critical influence (e.g. "I love you" vs. "I loved you"), while in proteins, effects may be more aggregate (e.g. hydrophilic chains in intermembrane sequences). Finally, natural languages typically have fewer distant interactions, while proteins, due to their 3D structure, commonly form interactions between residues that are far away on the linear sequence.

In this review, we present a modern view on applications of NLP methods to the study of protein sequences. We begin by exemplifying the types of prediction tasks that one might be interested in when studying proteins at a global or local level. We then discuss the concept of tokenization, namely the choosing of a discrete set of tokens for representing given text or protein sequences, which is typically the first preprocessing step in NLP tasks. Next, we present classical NLP methods such as bag-of-words and k-mer, which provide a strong baseline for many applications. We also mention other classic searching and text-similarity methods such as BLAST and hashing. We then move to more modern approaches, focusing on word embedding, contextualized embedding and deep-learning methods. We concentrate on deep language models (especially protein-language models). We end by reflecting on some limitations of deep-learning models and current trends in the field.

The aim of this review is to introduce readers to applications of NLP methods to protein research, and to inform them about recent developments in the field. While aimed at a broad audience, we assume familiarity with basic concepts in biology (e.g. amino acids, phosphorylation) and machine learning (e.g. feature extraction, deep learning). To assist the reader with this background knowledge, we provide a short glossary with some important terms.

2. Sequence-based prediction tasks: Global vs. Local

NLP methods have been used to address a large spectrum of sequence-based prediction tasks in text and proteins. At the most fundamental level, sequence-based tasks are either global or local (Fig. 1A). Global tasks output predictions for the entire sequence. For example, in classic NLP, the sentiment of a movie review (e.g. "It was excellent": +1, "It was terrible": −1) is a global property of the text. Local tasks, on the other hand, attempt to make a prediction over every element of a given sequence. Part-of-speech tagging, namely the categorization of the grammatical role of every word in a text (e.g. noun, verb), is a classic example of a local NLP task.

When dealing with global protein tasks, we are interested in making some inference or predictions about the protein sequence as a whole. For example, we may want to determine what type of protein we are dealing with (e.g. an enzyme, a receptor or a structural protein) [52,51], or where it is expressed in the cell (e.g. in the nucleus, cytoplasm or extracellular space) [37,5,88]. Other global properties include thermal stability, source organism, and functional protein annotations (e.g. gene ontology, GO) such as antiviral activity [79,64,41,84,5,25].

With local protein tasks, on the other hand, the goal is to make claims about specific residues in the protein sequence, potentially about all of them. For example, a common task is the prediction of 2D and 3D structure from an AA sequence (which, as mentioned, in theory contains all the necessary information). The output of this task would be the 2D structure for each residue in the protein (e.g. helix, turn, beta strand). In the case of 3D structure, the output might be the exact 3D coordinates of each residue [91,14] or its location relative to other residues (contact-map prediction). Another local task is the prediction of post-translational modifications (PTMs) such as phosphorylation or cleavage sites [90,18].

Sequence-based predictions can also include other inputs (that could be collected experimentally or computed) in addition to the sequence itself, such as publication date, organism, protein annotations (e.g. PTMs) and domains [41,88].

3. The atomic unit of information: Tokenization

Computational text analysis requires tokenization, i.e. splitting the text into individual tokens, which are the atomic units of information in a chosen language representation (Fig. 1B). English NLP models typically use words as tokens, although some approaches use individual characters. Individual-character tokens offer greater flexibility, especially for out-of-vocabulary or misspelled words, or for languages without clear separation between words such as Mandarin. Character-level tokenization also entails smaller vocabulary, which often results in lower memory requirement. However, more tokens must be used to form a sentence, leading to more long-distance dependencies. A middle-ground approach between words and characters is subword segmentation. Common examples are WordPiece, SentencePiece and Byte Pair Encoding (BPE) [92,48], where a vocabulary is initialized to individual characters, and the most frequent combinations of symbols in the vocabulary are iteratively merged into the vocabulary. For example, the frequently co-occurring "i" and "t" would be merged into a token "it", while "ending" might be represented by the two tokens "end" and "ing". This approach offers good coverage of words (including rare and out-of-vocabulary words), while still limiting the vocabulary size.

In proteins, the simplest and most common tokenization method is to regard individual AAs as character-level tokens. Since proteins do not have a well-defined vocabulary of words, word-level tokenization is not a well-defined option in the case of proteins. Subword segmentation, on the other hand, does not require any predefined knowledge of words in the target language, making it a potentially interesting approach for discovering "words" or motifs in proteins [107,8,12,53].

4. Counting: bag-of-words and k-mers

Most machine-learning algorithms (e.g. logistic regression, SVM, random forest) require a fixed-size input vector of features. Bag-of-words (BoW) is the simplest and most popular feature extraction method in text. In BoW, a text is split into its constituent parts (tokens), which are then counted without regard to their original order (Fig. 1C). The assumption is that texts using similar words are also similar in other ways (e.g. topic, author, sentiment). BoW can normalize the resulting vector with respect to the total number of tokens in the same text to produce token frequencies, or with respect to their normalized counts in all texts in a dataset to produce Term Frequency–Inverse Document Frequency (TF-IDF)

vectors [87]. BoW can also be encoded as binary features marking the occurrence of tokens in a text (instead of counts). In addition to counting individual tokens, BoW can also count multi-token combinations, with overlap (k-mers) or without (n-grams). Since they are largely the same, we refer to n-grams and k-mers interchangeably. BoW is fast, efficient, simple and suited for large texts of varying lengths. On the other hand, since the order of tokens in the input text is lost in BoW representation, this approach might be too simplistic for many tasks (yet it still often succeeds surprisingly well) [33,63,111].

BoW has been used in many bioinformatics studies on proteins [66,18,64,11,62,103,51]. Protein BoW commonly count character-level tokens (i.e. AAs). However, one can go beyond AAs and extract BoW features from other sequence representations such as 2D structure or reduced AA alphabets (e.g. hydrophobic/polar binary tokens) [66,108]. An advantage of the latter is a smaller alphabet, allowing for longer k-mers or n-grams. For example, AA-level 4-grams would result in 160,000 (20^4) features, whereas 7-grams over a 3-state alphabet (e.g. for the major classes of secondary structure) would result in only 2187 (3^7) features. The latter would be more discriminative in capturing longer patterns, while maintaining lower sparsity (i.e. less zeros) in the feature space. Another possible variation is k-mer mirror symmetry, meaning that AA k-mers such as "PG" and "GP" are treated as the same feature, yielding a more compact feature space [64,66]. Ideally, we would count 3D protein topologies, domains or motifs instead of letters to characterize protein function [20]. However, we lack reliable annotations for most proteins [17]. Despite its simplicity, BoW is an effective method for many protein-related tasks [42,113,79,66,64,69,86].

5. Searching by similarity

A common task in NLP and bioinformatics is finding similar strings and sequences. In bioinformatics, the most common algorithm for sequence searching is BLAST. Another option is Locality-Sensitive Hashing (LSH), a popular method for indexing and finding texts at scale. In brief, a document is represented as a BoW vector. The vector is then hashed multiple times using a hash function that encourages "collisions" between similar document vectors. Retrieval of documents from a hash bucket can be done in O(1), as items within the same bucket will be similar to one another. This can be adapted to bioinformatics sequence databases to complement existing slow sequence-similarity methods such as BLAST, whose speed (~O(mn)) is adversely affected by the exponential growth in sequences [100]. A common application of fast approximate search is the identification of short peptides in mass-spectrometry proteomic databases [75,1,28]. Sequence similarity metrics are also useful for machine-learning methods. For example, support vector machines (SVMs) with string kernels compare proteins or texts by sub-sequence similarity [51,52,12]. Another method to index and search protein sequences is through AA k-mers. For example, HHblits, HHsuite and Pep2Pro use k-mers to rapidly find similar proteins, peptides and domains [82,97,10].

6. Word embeddings

Word embeddings are a family of algorithms that represent tokens (e.g. words) with fixed-size dense vectors ("embeddings"), such that similar words obtain similar vector representations [32]. Word similarity is usually based on neighbouring words, meaning that different words with a related use (e.g. "spoon" and "fork") will obtain similar representations, while unrelated words that rarely appear together (e.g. "Darth Vader" and "mRNA") are expected to obtain distinct embeddings. Word embeddings provide useful low-dimensional representations (compared to the sparse, high-dimensional BoW representations) that still preserve semantic information about the input sequence. For example, taking the average embedding over all words in a text often leads to strong results with downstream supervised or clustering algorithms 7,9].

The method has been popularized with efficient algorithms such as word2vec [59,70,42,16]. These popular algorithms are all fundamentally "vector space models", and conceptually similar to decomposition of the co-occurrence matrix, which captures the probabilities of tokens to occur next to each other in the text [32]. Word2vec has two model architectures: continuous bag-of-words or skip-grams. In the bag-of-words architecture, the model predicts the current word from a window of surrounding context words (while the order of context words is ignored). The skip-gram architecture is exactly the opposite: the model uses the current word to predict the surrounding window of context words. FastText combines words with subword information when learning representations, to better handle unknown or syntactically similar words.

Low-dimensional embeddings are popular in NLP due to the huge vocabulary (often >100 k of words) of natural languages. In proteins we have only ~20 AAs. While we can embed AAs onto a lower-dimensional space, it is not as clearly beneficial [8]. While dimensionality reduction is of limited use when working on single AAs, it can provide useful compact representations when considering extended AA combinations. For example, ProtVec used word2-vec on AA 3-mers to extract a 300-dimensional vector representation instead of 8 k distinct trigrams [9]. It is interesting to note that AA embeddings learned by machine-learning models closely resemble those resulting from decomposing AA substitution matrices in terms of the functional clusters they induce [83,65,61,72].

7. Contextualized embeddings & deep learning

Word2vec & similar methods do not take local word order into account. When representing words as embeddings (after training), they also do not consider the surrounding context of words. For example, in the sentences "man *bites* dog", "dog *bites* man" and "love *bites*", the vector representation of "bites" will always be the same, regardless of its context. Contextualized embeddings, on the other hand, are aware of the surrounding word context (including order), yielding different representations for the same word in different contexts [95,57]. While more complex and computationally expensive, contextualized embedding models yielded state-of-the-art results (at the time) on a number of benchmarks [57,71].

Contextualized embeddings are typically based on neural networks. Popular deep-learning architectures are long short-term memory (LSTM) [36], sequence-to-sequence (seq2seq) [101] and attention [104]. In seq2seq models, a text is transformed using an encoder component, then a separate decoder uses the encoded representation to solve some task (e.g. translating between English and French). Attention models use attention layers (also called attention heads) that allow the network to concentrate on specific tokens in the text [104]. For example, in the sentence "The law will never be perfect, but its application should be just", when the network analyzes the word "its" we expect its attention heads to concentrate on the words "The law". Deep-learning architectures commonly rely on transfer learning, where a model is trained on one problem and then fine-tuned (transferred) to another problem in a similar domain, for which data is often scarce.

Early contextualized embedding models included ELMO [71] and CoVe [57]. ELMO uses representations derived from the hidden

states of bidirectional LSTMs. CoVe is a seq2seq model with attention, originally developed for language translation. A CoVe model pretrained on translation was then used on other NLP tasks. There have been works using contextualized embedding models (e.g. ELMO) on proteins for supervised-learning tasks (such as GO annotation, subcellular localization or structure prediction), as well as clustering sequences based on the learned representations [15,29,54,34].

8. Deep language models

Many fields hope for an "Imagenet moment" [85] – namely, a model, dataset and pretraining tasks that provide strong off-the-shelf performance for most tasks, even with little data. Arguably, the field of NLP has recently reached this milestone, thanks to deep language models such as ULMFiT, BERT, XLNet and a range of other BERT variations (e.g. ALBERT, RoBERTA) [30,27,114,77,50,55,38]. In language modelling, a model is trained to predict tokens in a text, based on their surrounding context (Fig. 2). For example, an English language model might be given a masked sentence such as "The ____ sat on the mat" and be tasked to predict what English words are plausible candidates for the mask token (e.g. "cat" or "dog"). While language modeling problems may not have unique solutions (e.g. both cats and dogs are plausible mat-sitting entities), it serves as an excellent generalizable proxy for understanding general language structures. A good English language model should score the sentence "Moriarty on Cthulhu sat" as less probable than (the grammatically correct) "The cat sat on the mat" (Cthulhu being a Lovecraftian entity larger than mountains). A crucial advantage of language models for pretraining is that they are self-supervised (as in the masked language task): the model predicts an explicit ground truth, but it doesn't require labelled data, making it usable on any corpus, at potentially massive scale [77,21].

BERT [27] uses bidirectional masked language modelling, where a fraction of words are masked out and have to be predicted by the model. In bidirectional language modelling, the model looks at all surrounding context of a masked token, instead of just at the tokens preceding it. ELECTRA [24] (a BERT variant) predicts which tokens have been replaced with adversarially-generated tokens. This pretraining task was demonstrated as more efficient than masked language modeling, presumably because the task is defined over all input tokens (which might have been replaced by the adversarial model) rather than just a subset of mask tokens. ELECTRA was shown to yield comparable results to BERT with less than 10% of the compute time. Effectively all state-of-the-art neural language models are attention-based, typically using the Transformer architecture [27]. Research into why deep language models work so well is ongoing. These models share the following characteristics: 1) state-of-the-art performance on a wide variety of benchmarks [106]; 2) self-supervised language modelling pretraining on large text corpuses [27]; 3) huge, deep neural networks, with continued improvement from ever larger, deeper models and datasets [21].

Deep language models have started to show promise in protein and genomic research [119,45,83,79,29,107,34,40]. Successful architectures used in protein language modeling include LSTM and attention [104] (in particular BERT). Examples of LSTM-based protein language models include UDSMProt [99] and Unirep (which is based on ULMFiT) [3]. Downstream tasks for protein language models include the detection of the taxonomic origin of proteins, or scoring the likelihood or stability of natural or synthetically-designed sequences. For example, protein language models trained on different taxa could be used to identify viral protein sequences in metagenomic samples (e.g. from mass spectrometry) [34,56].

Progress in the development of protein models is dependent on representative evaluation benchmarks encompassing a variety of protein-related tasks, such as the TAPE collection of benchmarks (Tasks Assessing Protein Embeddings) [79]. TAPE combined a diverse set of tasks in a convenient, standardized format, and evaluated different models. For each model, they showed the benefit of language-model pretraining. Evaluated models included a BERT-like Transformer pretrained on ~30 M Pfam domains, deep convolutional networks pretrained on 3D-structure contact prediction [13,116], LSTM language models [3], as well as non-deep-learning methods which excelled at some tasks (specifically secondary structure prediction). A recent work sought to interpret

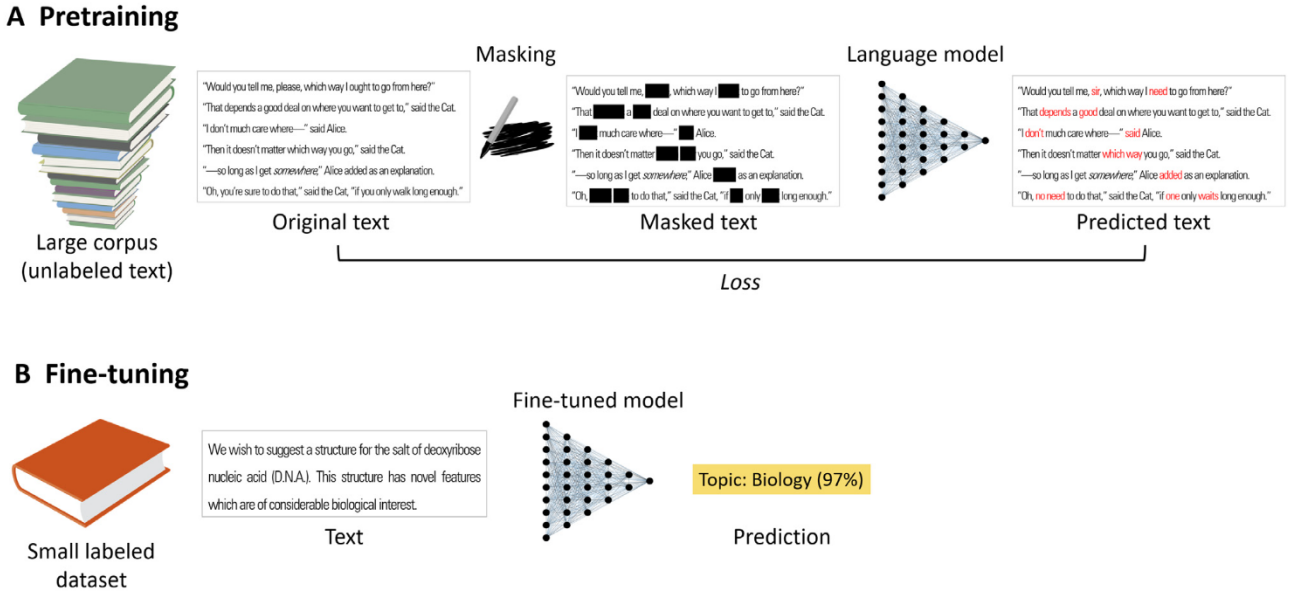

Fig. 2. Language models (A) Language models are trained on self-supervised tasks over huge corpuses of unlabeled text. For example, in the masked language task, some fraction of the tokens in the original text are masked at random, and the language model attempts to predict the original text. (B) (Pre-)trained language models are commonly fine-tuned on downstream tasks over labeled text, through a standard supervised-learning approach. Fine-tuning is typically much faster and provides superior performance than training a model from scratch, especially when labeled data is scarce.

how attention-based models work in proteins, and the parts of proteins that attention heads focus on across different tasks (e.g. which residues are most relevant in predicting which AAs interact in a protein's 3D structure) [105]. Facebook's Evolutionary Scale Model is currently the largest developed protein language model, with 36 layers and over 700 million parameters [83]. This attention-based model was pretrained on 250 M protein sequences with a masked language task. Like in NLP, ever larger models and protein datasets yield consistent improvement on TAPE's benchmark tasks, while language modelling performance is not saturated even with the largest models.

Some pretraining methods add additional tasks, such as subsequence order prediction (e.g. in NLP, which of two sentence comes first). Pretraining tasks that are relevant to the final task are usually more helpful than generic tasks such as sentence order prediction [73], but should be weighed against the amount, quality and representativeness of data available. For example, a pretraining task of predicting whether one Wikipedia article links to another is relevant to a downstream task of link prediction between entities in knowledge graphs, and indeed improves performance [78,115]. In protein research, an analogous task of pretraining on protein–protein interactions is more problematic due to the far smaller, sparser and more biased data [94,89].

9. Language generation

Natural language generation is a challenging NLP task, where a model generates realistic-looking text (e.g. article writing, chatbots). Language generation based on deep language models has shown great improvement, with increasingly massive models such as OpenAI's GPT-2 and GPT-3 often fooling humans. These models demonstrate the trend of increased parameter size and larger datasets resulting in improved performance, with GPT-3 having over 170 billion parameters. Generated texts can be controlled to match user-defined styles, task-specific behaviour and other attributes [76,19,43].

Language models have been used to understand and predict viral mutations that evade neutralizing antibodies [35]. Language generation models can also be applied to synthetic protein design, as in ProGen and other works [56,110,4]. For example we might generate peptides with the gene-ontology attribute "defense response to virus" and an initial primer sequence amenable to binding a sequence of interest, such as the ACE2 receptor targeted by SARS-CoV-2 [112].

10. No (deep) silver bullet

Deep-learning models are not a magical panacea, and have a number of disadvantages. They are slow to train, and can underperform simpler methods (such as BLAST-based nearest-neighbour search or logistic regression over BoW representation [79,58,2]. In particular, simpler methods are more suitable for small datasets, noisy data, or when the underlying signal follows simple rules (e.g. an exact motif). Deep learning is also sensitive to the lengths of the texts or sequences involved – a dataset containing protein sequence lengths ranging from 10 to 10,000 AAs can be challenging to process. The high memory and computation requirements of large, deep models (such as BERT) make processing long sequences, or pairs of sequences (e.g. for predicting interactions between proteins) challenging [23]. Deep-learning models also easily overfit (even on random noise), and may not necessarily generalize to new, unseen data [120,118]. Another major disadvantage of deep learning is its sensitivity to hyperparameters (such as the optimizer or learning rate) and other choices. A related problem is its lack of stability, as opposed to more robust algorithms such as logistic regression or random forests. Deep-learning models are also hard to interpret, even by experienced practitioners. Even with deep models, the incorporation of expert knowledge and feature engineering can improve performance more than sophisticated models, especially with features that are not directly derived from the sequence, such as protein post-translational modifications or evolutionary information [80].

11. Where are we heading?

NLP methods are becoming an important inspiration for bioinformatics. Deep learning and NLP methods are making inroads into protein research, and the recent successes of AlphaFold in practically solving the protein structure prediction problem may well be considered an "imagenet moment" for the field [26]. The trend in NLP is towards deeper and larger language models such as GPT-3. In particular, unsupervised and self-supervised learning on huge datasets are an important feature of state-of-the-art methods. For how long this trend will last and how far it will enable us to push the boundaries of NLP and protein research – only time will tell. As of the present day, fine-tuning of pretrained deep models have shown considerable promise for improving our ability to solve problems, even on small data [79,83,19].

Exciting as recent progress may have been, the impressive performance of state-of-the-art models should not lead us to neglect more mundane but crucially important efforts in the field. Above all, abundant and high-quality data plays a major role in the progress of any domain of machine learning, and especially so in protein research. High-throughput molecular assays and data curation in resources such as UniProt [17] are the field's engine. Open, standardized data and methods (including the open-source Keras [22] and Pytorch [68] libraries) are valuable research catalyzers.

Competitions such as CAFA (GO annotations and protein function prediction) [31,41], CASP (3D structure prediction) [47] and CAPRI (protein–protein docking prediction) [39] provide the protein research community with rigorous tests for evaluating and comparing prediction algorithms, with new, unseen test sets on every competition cycle. The protein research community excels in coordinating such competitions, and is considered an inspiration for other fields. According to DeepMind, this was one of the primary factors pushing them to develop AlphaFold and participate in CASP13 and CASP14 (2020), which ultimately led to what appears as one of the major scientific breakthroughs of recent years [26]. In terms of open benchmarks, on the other hand, the field of computational protein research is still behind compared to NLP and other machine-learning domains, where datasets such as Imagenet and GLUE are de-facto standards for model evaluation [46,106]. Benchmarks, unlike competitions, are instantaneously accessible at any given point in time and, as a result, are crucial for continuous research work and publication. The existence of standardized, objective yardsticks for comparing methods is crucial to focusing efforts on the most promising methods and ideas.

Glossary

Term	Definition
Artificial neural networks	Artificial neural networks are a class of machine-learning models that can fit nonlinear, complex data.
Attention layer	A type of layer used in deep learning that allows the network to concentrate on specific elements in the input sequence [104].
Deep learning	Neural networks with many hidden layers are commonly referred to as "deep learning". Deep-learning architectures include convolutional, recurrent and attention layers.
Features	Input properties fed into machine-learning algorithms (e.g. the length of a sequence) are commonly referred to as "features".
Feature engineering	The creation and selection of features to extract from data, which is considered a crucial part of machine-learning projects.
Low-dimensional embedding	A mathematical mapping from a high-dimensional space of inputs to a lower-dimensional space of representations.
Post-translational modification (PTM)	Chemical enzymatic alterations of amino-acid residues in proteins which often lead to functional changes. Major PTMs include phosphorylation, glycosylation and proteolytic cleavage.
Protein domain	An evolutionary-conserved protein region with independent, well-defined 3D structure and function. Many proteins contain multiple domains.
Protein motif	A short, conserved segment of amino acids in a protein associated with some function such as binding properties.
Self-supervised learning	A machine-learning paradigm for training supervised models over unsupervised (namely unlabeled) datasets by automatically generating labels. With text, this might be the prediction of the next word in a text [38,27].
Transfer learning	Taking a model trained to solve one problem, and fine-tuning its parameters to solve another, related task. For example, training a computer-vision model to recognize cars, and then teaching it to recognize trucks [81].
Transformers	A deep-learning architecture consisting of attention-based layers that is particularly suited for sequence inputs and outputs ([27,104,44].

CRediT authorship contribution statement

Dan Ofer: Conceptualization, Investigation, Writing. **Nadav Brandes:** Conceptualization, Visualization, Writing. **Michal Linial:** Conceptualization, Supervision, Writing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

We would like to thank Tair Shauli for her feedback on this review. We also thank the Hebrew University Center for Interdisciplinary Data-science Research (CIDR) for their financial support.

References

[1] Akhtar Malik N, Southey Bruce R, Andrén Per E, Sweedler Jonathan V, Rodriguez-Zas Sandra L. Evaluation of Database Search Programs for Accurate Detection of Neuropeptides in Tandem Mass Spectrometry Experiments. J Proteome Res 2012;11(12):6044–55. https://doi.org/10.1021/pr3007123.
[2] Allam Ahmed, Nagy Mate, Thoma George, Krauthammer Michael. Neural networks versus logistic regression for 30 days all-cause readmission prediction. Sci Rep 2019;9(1):9277. https://doi.org/10.1038/s41598-019-45685-z.
[3] Alley Ethan C, Khimulya Grigory, Biswas Surojit, AlQuraishi Mohammed, Church George M. Unified rational protein engineering with sequence-based deep representation learning. Nat Methods 2019;16(12):1315–22. https://doi.org/10.1038/s41592-019-0598-1.
[4] Almagro Armenteros, Jose Juan, Alexander Rosenberg Johansen, Ole Winther, and Henrik Nielsen. Language Modelling for Biological Sequences – Curated Datasets and Baselines. BioRxiv 2020. March, 2020.03.09.983585. 10.1101/2020.03.09.983585.
[5] Almagro Armenteros, José Juan, Casper Kaae Sønderby, Søren Kaae Sønderby, Henrik Nielsen, and Ole Winther. 2017. "DeepLoc: Prediction of Protein Subcellular Localization Using Deep Learning." Edited by John Hancock. Bioinformatics 33 (21): 3387–95. 10.1093/bioinformatics/btx431.
[6] Angermueller Christof, Pärnamaa Tanel, Parts Leopold, Stegle Oliver. Deep learning for computational biology. Mol Syst Biol 2016;12(7):878. https://doi.org/10.15252/msb.20156651.
[7] Arora, Sanjeev, Yingyu Liang, and Tengyu Ma. 2016. "A Simple but Tough-to-Beat Baseline for Sentence Embeddings," November. https://openreview.net/forum?id=SyK00v5xx.
[8] Asgari Ehsaneddin, McHardy Alice C, Mofrad Mohammad RK. Probabilistic variable-length segmentation of protein sequences for discriminative motif discovery (DiMotif) and sequence embedding (ProtVecX). Sci Rep 2019. https://doi.org/10.1038/s41598-019-38746-w.
[9] Asgari Ehsaneddin, Mofrad Mohammad RK. Continuous Distributed representation of biological sequences for deep proteomics and genomics. PLoS ONE 2015;10(11). https://doi.org/10.1371/journal.pone.0141287.
[10] Askenazi Manor, Marto Jarrod A, Linial Michal. The complete peptide dictionary – a meta-proteomics resource. Proteomics 2010;10(23):4306–10. https://doi.org/10.1002/pmic.201000270.
[11] Barla Annalisa, Jurman Giuseppe, Riccadonna Samantha, Merler Stefano, Chierici Marco, Furlanello Cesare. Machine learning methods for predictive proteomics. Briefings Bioinf 2008;9(2):119–28. https://doi.org/10.1093/bib/bbn008.
[12] Ben-hur Asa, Brutlag Douglas, Ben-hur Douglas Brutlag Asa. Protein Sequence Motifs: Highly Predictive Features of Protein Function. Stud Fuzziness Soft Comput 2006;207.
[13] Bepler, Tristan, Bonnie Berger. 2019. "Learning Protein Sequence Embeddings Using Information from Structure." ArXiv:1902.08661 [Cs, q-Bio, Stat], October. http://arxiv.org/abs/1902.08661.
[14] Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, et al. The protein data bank. Nucleic Acids Res 2000;28(1):235–42. https://doi.org/10.1093/nar/28.1.235.
[15] Bileschi ML, Belanger D, Bryant D, Sanderson T, Brandon Carter D, Sculley MA, et al. Using deep learning to annotate the protein universe. BioRxiv 2019. https://doi.org/10.1101/626507.

[16] Bojanowski P, Grave E, Joulin A, Mikolov T. Enriching word vectors with subword information. Trans Assoc Computat Linguis 2017;5 (December):135–46. https://doi.org/10.1162/tacl_a_00051.
[17] Boutet E, Lieberherr D, Tognolli M, Schneider M, Bairoch A. UniProtKB/Swiss-Prot: The manually annotated section of the uniprot knowledgebase. Methods Mol Biol 2007;406:89–112.
[18] Brandes N, Ofer D, Linial M. ASAP: A machine learning framework for local protein properties. Database 2016;2016. https://doi.org/10.1093/database/baw133.
[19] Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, et al. 2020. Language Models Are Few-Shot Learners. ArXiv:2005.14165 [Cs], July. http://arxiv.org/abs/2005.14165.
[20] Budowski-Tal, Inbal, Yuval Nov, and Rachel Kolodny. FragBag, an Accurate Representation of Protein Structure, Retrieves Structural Neighbors from the Entire PDB Quickly and Accurately. Proceedings of the National Academy of Sciences of the United States of America. 2010. 107 (8): 3481–86. 10.1073/pnas.0914097107.
[21] Chen, Ting, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. 2020. "Big Self-Supervised Models Are Strong Semi-Supervised Learners." Advances in Neural Information Processing Systems 33.
[22] Chollet, François. 2015. Keras.
[23] Choromanski, Krzysztof, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, et al. 2020. "Rethinking Attention with Performers." ArXiv:2009.14794 [Cs, Stat], September. http://arxiv.org/abs/2009.14794.
[24] Clark, K., Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. "ELECTRA: Pre-Training Text Encoders as Discriminators Rather Than Generators." ArXiv abs/2003.10555.
[25] Cozzetto, Domenico, Federico Minneci, Hannah Currant, and David T. Jones. 2016. "FFPred 3: Feature-Based Function Prediction for All Gene Ontology Domains." Sci Rep 6 (August). 10.1038/srep31865.
[26] Demis Hassabis. 2020. "High Accuracy Protein Structure Prediction Using Deep Learning." Fourteenth Critical Assessment of Techniques for Protein Structure Prediction (Abstract Book), December. https://predictioncenter.org/casp14/doc/CASP14_Abstracts.pdf.
[27] Devlin, J., Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding." In NAACL-HLT. 10.18653/v1/N19-1423.
[28] Dutta D, Chen T. Speeding up Tandem Mass Spectrometry Database Search: Metric Embeddings and Fast near Neighbor Search. Bioinformatics 2007;23 (5):612–8. https://doi.org/10.1093/bioinformatics/btl645.
[29] Elnaggar, Ahmed, Michael Heinzinger, Christian Dallago, Ghalia Rihawi, Yu Wang, Llion Jones, Tom Gibbs, et al. 2020. "ProtTrans: Towards Cracking the Language of Life's Code Through Self-Supervised Deep Learning and High Performance Computing," July. http://arxiv.org/abs/2007.06225.
[30] Feng, Zhangyin, Daya Guo, Duyu Tang, Nan Duan, Xiaocheng Feng, Ming Gong, Linjun Shou, et al. 2020. "CodeBERT: A Pre-Trained Model for Programming and Natural Languages," February. https://arxiv.org/abs/2002.08155v4.
[31] Gillis, Jesse, Paul Pavlidis. 2013. "Characterizing the State of the Art in the Computational Assignment of Gene Function: Lessons from the First Critical Assessment of Functional Annotation (CAFA)." BMC Bioinformatics 14 Suppl 3 (January): S15.
[32] Goldberg Y, Levy O. Word2vec explained: Deriving Mikolov et al'.s negative-sampling word-embedding method. *ArXiv:1402.3722 [Cs, Stat]* 2014.
[33] Halevy A, Norvig P, Pereira F. The unreasonable effectiveness of data. IEEE Intell Syst 2009;24(2):8–12. https://doi.org/10.1109/MIS.2009.36.
[34] Heinzinger M, Ahmed Elnaggar Yu, Wang CD, Nechaev D, Matthes F, Rost B. Modeling aspects of the language of life through transfer-learning protein sequences. BMC Bioinf 2019;20(1):1–17. https://doi.org/10.1186/s12859-019-3220-8.
[35] Hie B, Zhong ED, Berger B, Bryson B. Learning the language of viral evolution and escape. Science 2021;371(6526):284–8. https://doi.org/10.1126/science.abd7331.
[36] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput 1997;9(8):1735–80.
[37] Höglund A, Dönnes P, Blum T, Adolph H-W, Kohlbacher O. MultiLoc: prediction of protein subcellular localization using n-terminal targeting sequences, sequence motifs and amino acid composition. Bioinformatics (Oxford, England) 2006;22(10):1158–65. https://doi.org/10.1093/bioinformatics/btl002.
[38] Howard J, Ruder S. Universal language model fine-tuning for text classification. *ArXiv* 2018.
[39] Janin, Joël, Kim Henrick, John Moult, Lynn Ten Eyck, Michael J. E. Sternberg, Sandor Vajda, Ilya Vakser, and Shoshana J. Wodak. 2003. CAPRI: A Critical Assessment of PRedicted Interactions. Proteins: Struct Funct Bioinformatics 52 (1): 2–9. 10.1002/prot.10381.
[40] Ji, Yanrong, Zhihan Zhou, Han Liu, and Ramana V Davuluri. 2021. "DNABERT: Pre-Trained Bidirectional Encoder Representations from Transformers Model for DNA-Language in Genome." Edited by Dr Janet Kelso and Janet Kelso. Bioinformatics, February, btab083. 10.1093/bioinformatics/btab083.
[41] Jiang Y, Oron TR, Clark WT, Bankapur AR, D'Andrea D, Lepore R, et al. An expanded evaluation of protein function prediction methods shows an improvement in accuracy. Genome Biol 2016;17(1). https://doi.org/10.1186/s13059-016-1037-6.
[42] Joulin, Armand, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. "Bag of Tricks for Efficient Text Classification." ArXiv:1607.01759 [Cs], August. http://arxiv.org/abs/1607.01759.
[43] Keskar, Nitish Shirish, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. "CTRL: A Conditional Transformer Language Model for Controllable Generation." ArXiv:1909.05858 [Cs], September. http://arxiv.org/abs/1909.05858.
[44] Klein Guillaume, Kim Yoon, Deng Yuntian, Senellart Jean, Rush Alexander. In: Demonstrations System, editor. OpenNMT: Open-Source Toolkit for Neural Machine Translation. Vancouver, Canada: Association for Computational Linguistics; 2017. p. 67–72.
[45] Koumakis L. Deep learning models in genomics; are we there yet? Comput Struct Biotechnol J 2020;18:1466–73. https://doi.org/10.1016/j.csbj.2020.06.017.
[46] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. ImageNet Classification with Deep Convolutional Neural Networks 2012. https://doi.org/10.1145/3065386.
[47] Kryshtafovych A, Schwede T, Topf M, Fidelis K, Moult J. Critical assessment of methods of protein structure prediction (casp)—round xiii. Proteins Struct Funct Bioinf 2019;87(12):1011–20. https://doi.org/10.1002/prot.25823.
[48] Kudo, Taku. 2018. "Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates." ArXiv:1804.10959 [Cs], April. http://arxiv.org/abs/1804.10959.
[49] Lample, Guillaume, and François Charton. 2019. "Deep Learning for Symbolic Mathematics." ArXiv:1912.01412 [Cs], December. http://arxiv.org/abs/1912.01412.
[50] Lan Z, Chen M, Goodman S, Gimpel K, Sharma P, Soricut Rd. ALBERT: A lite BERT for self-supervised learning of language representations. *ArXiv* 2020.
[51] Leslie, Christina, Eleazar Eskin, and William Stafford Noble. 2002. "The Spectrum Kernel: A String Kernel for SVM Protein Classification." Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing 575 (January): 564–75.
[52] Leslie CS, Eskin E, Cohen A, Weston J, Noble WS. Mismatch string kernels for discriminative protein classification. Bioinformatics (Oxford, England) 2004;20(4):467–76. https://doi.org/10.1093/bioinformatics/btg431.
[53] Liang, Wang, and Zhao KaiYong. 2015. "Detecting 'Protein Words' through Unsupervised Word Segmentation." ArXiv:1404.6866 [Cs, q-Bio], October. http://arxiv.org/abs/1404.6866.
[54] Littmann, Maria, Michael Heinzinger, Christian Dallago, Tobias Olenyi, and & Burkhard Rost. 2020. "Embeddings from Deep Learning Transfer GO Annotations beyond Homology." BioRxiv, September, 2020.09.04.282814. 10.1101/2020.09.04.282814.
[55] Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. "RoBERTa: A Robustly Optimized BERT Pretraining Approach." ArXiv:1907.11692 [Cs], July. http://arxiv.org/abs/1907.11692.
[56] Madani, Ali, Bryan McCann, Nikhil Naik, Nitish Shirish Keskar, Namrata Anand, Raphael R. Eguchi, Po-Ssu Huang, and Richard Socher. 2020. "ProGen: Language Modeling for Protein Generation." BioRxiv, January, 2020.03.07.982272. 10.1101/2020.03.07.982272.
[57] McCann, Bryan, James Bradbury, Caiming Xiong, and Richard Socher. 2018. "Learned in Translation: Contextualized Word Vectors." ArXiv:1708.00107 [Cs], June. http://arxiv.org/abs/1708.00107.
[58] Mignan A, Broccardo M. One neuron is more informative than a deep neural network for aftershock pattern forecasting. Nature 2019;574(7776):E1–3. https://doi.org/10.1038/s41586-019-1582-8.
[59] Mikolov T, Chen K, Corrado G, Dean J. Distributed representations of words and phrases and their compositionality. Nips 2013;1–9. https://doi.org/10.1162/jmlr.2003.3.4-5.951.
[60] Min, Seonwoo, Byunghan Lee, and Sungroh Yoon. 2016. "Deep Learning in Bioinformatics." Briefings Bioinf, July, bbw068. 10.1093/bib/bbw068.
[61] Murphy LR, Wallqvist A, Levy RM. Simplified amino acid alphabets for protein fold recognition and implications for folding. Protein Eng 2000;13(3):149–52. https://doi.org/10.1093/protein/13.3.149.
[62] Naamati G, Askenazi M, Linial M. ClanTox: A classifier of short animal toxins. Nucleic Acids Res 2009;37(Suppl. 2). https://doi.org/10.1093/nar/gkp299.
[63] Nematzadeh A, Meylan SC, Griffiths TL. Evaluating vector-space models of word representation, or, the unreasonable effectiveness of counting words near other words. CogSci 2017.
[64] Ofer D, Linial M, Ofer D, Linial M. NeuroPID: A predictor for identifying neuropeptide precursors from metazoan proteomes. Bioinformatics (Oxford, England) 2014;30(7):931–40. https://doi.org/10.1093/bioinformatics/btt725.
[65] Ofer, Dan. 2016. "Machine Learning for Protein Function." ArXiv:1603.02021 [q-Bio], March. http://arxiv.org/abs/1603.02021.
[66] Ofer, Dan, and Michal Linial. 2015. "ProFET: Feature Engineering Captures High-Level Protein Functions." Bioinformatics (Oxford, England), June. 10.1093/bioinformatics/btv345.
[67] Papanikolaou N, Pavlopoulos GA, Theodosiou T, Iliopoulos I. Protein–protein interaction predictions using text mining methods. *Methods* 2015;74:47–53. https://doi.org/10.1016/j.ymeth.2014.10.026.
[68] Paszke, Adam, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, et al. 2019. "PyTorch: An Imperative Style, High-Performance Deep Learning Library." ArXiv:1912.01703 [Cs, Stat], December. http://arxiv.org/abs/1912.01703.

[69] Pe'er Itsik, Felder Clifford E, Man Orna, Silman Israel, Sussman Joel L, Beckmann Jacques S. Proteomic Signatures: Amino Acid and Oligopeptide Compositions Differentiate among Phyla. Proteins 2004;54(1):20–40. https://doi.org/10.1002/prot.10559.
[70] Pennington, Jeffrey, Richard Socher, and Christopher Manning. 2014. "Glove: Global Vectors for Word Representation." In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1532–43. Doha, Qatar: Association for Computational Linguistics. 10.3115/v1/D14-1162.
[71] Peters, Matthew E., Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. "Deep Contextualized Word Representations." ArXiv:1802.05365 [Cs], March. http://arxiv.org/abs/1802.05365.
[72] Peterson Eric L, Kondev Jané, Theriot Julie A, Phillips Rob. Reduced amino acid alphabets exhibit an improved sensitivity and selectivity in fold assignment. Bioinformatics (Oxford, England) 2009;25(11):1356–62. https://doi.org/10.1093/bioinformatics/btp164.
[73] Pierse Nuo Wang, Jingwen Lu. Aligning the pretraining and finetuning objectives of language models. ArXiv 2020.
[74] Ptitsyn OB. How does protein synthesis give rise to the 3D-structure? FEBS Lett 1991;285(2):176–81. https://doi.org/10.1016/0014-5793(91)80799-9.
[75] Qin Chunyuan, Luo Xiyang, Deng Chuan, Shu Kunxian, Zhu Weimin, Griss Johannes, et al. Deep Learning Embedder Method and Tool for Mass Spectra Similarity Search. Journal of Proteomics 2021;232(February):. https://doi.org/10.1016/j.jprot.2020.104070104070.
[76] Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. "Language Models Are Unsupervised Multitask Learners," 24.
[77] Raffel Colin, Shazeer Noam, Roberts Adam, Lee Katherine, Narang Sharan, Matena Michael, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. J Machine Learning Res 2020;21(140):1–67.
[78] Raiman Jonathan, Raiman Olivier. DeepType: Multilingual entity linking by neural type system evolution. *ArXiv* 2018.
[79] Rao, Roshan, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Xi Chen, John Canny, Pieter Abbeel, and Yun S. Song. 2019. "Evaluating Protein Transfer Learning with TAPE," June. https://arxiv.org/abs/1906.08230.
[80] Rao, Roshan M., Jason Liu, Robert Verkuil, Joshua Meier, John Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. 2021. "MSA Transformer." BioRxiv, February, 2021.02.12.430858. 10.1101/2021.02.12.430858.
[81] Razavian Sharif Ali, Azizpour Hossein, Sullivan Josephine, Carlsson Stefan, Royal KTH. CNN Features Off-the-Shelf: An Astounding Baseline for Recognition. In: CVPRW '14 Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops. p. 512–9. https://doi.org/10.1109/CVPRW.2014.131.
[82] Remmert Michael, Biegert Andreas, Hauser Andreas, Söding Johannes. HHblits: Lightning-fast iterative protein sequence searching by hmm-hmm alignment. Nat Methods 2011;9(2):173–5. https://doi.org/10.1038/nmeth.1818.
[83] Rives, Alexander, Siddharth Goyal, Joshua Meier, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. 2019. "Biological Structure and Function Emerge from Scaling Unsupervised Learning to 250 Million Protein Sequences." 10.1101/622803.
[84] Rocklin Gabriel J, Chidyausiku Tamuka M, Goreshnik Inna, Ford Alex, Houliston Scott, Lemak Alexander, et al. Global analysis of protein folding using massively parallel design, synthesis, and testing. Science 2017;357(6347):168–75. https://doi.org/10.1126/science.aan0693.
[85] Ruder Sebastian. NLP's imagenet moment has arrived. Gradient. 2018. , https://thegradient.pub/nlp-imagenet/.
[86] Sadka T, Linial M. Families of membranous proteins can be characterized by the amino acid composition of their transmembrane domains. Bioinformatics 2005;21(1):i378–86. https://doi.org/10.1093/bioinformatics/bti1035.
[87] Salton, Gerard, and Michael J. McGill. 1983. Introduction to Modern Information Retrieval. McGraw-Hill Computer Science Series. New York: McGraw-Hill.
[88] Savojardo, Castrense, Pier Luigi Martelli, Piero Fariselli, and Rita Casadio. 2018. "DeepSig: Deep Learning Improves Signal Peptide Detection in Proteins." Edited by Alfonso Valencia. Bioinformatics 34 (10): 1690–96. 10.1093/bioinformatics/btx818.
[89] Schnoes Alexandra M, Ream David C, Thorman Alexander W, Babbitt Patricia C, Friedberg Iddo. Biases in the experimental annotations of protein function and their effect on our understanding of protein function space. PLoS Comput Biol 2013;9(5). https://doi.org/10.1371/journal.pcbi.1003063.
[90] Schweiger Regev, Linial Michal. Cooperativity within proximal phosphorylation sites is revealed from large-scale proteomics data. Biology Direct 2010;5(January):6. https://doi.org/10.1186/1745-6150-5-6.
[91] Senior Andrew W, Evans Richard, Jumper John, Kirkpatrick James, Sifre Laurent, Green Tim, et al. Improved protein structure prediction using potentials from deep learning. Nature 2020;577(7792):706–10. https://doi.org/10.1038/s41586-019-1923-7.
[92] Sennrich, Rico, Barry Haddow, and Alexandra Birch. 2016. "Neural Machine Translation of Rare Words with Subword Units." In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 1715–25. Berlin, Germany: Association for Computational Linguistics. 10.18653/v1/P16-1162.
[93] Shannon CE. Prediction and entropy of printed english. Bell Syst Tech J 1951;30(1):50–64. https://doi.org/10.1002/j.1538-7305.1951.tb01366.x.
[94] Singer, Uriel, Kira Radinsky, and Eric Horvitz. 2020. "On Biases of Attention in Scientific Discovery." Edited by Jonathan Wren. Bioinformatics, December, btaa1036. 10.1093/bioinformatics/btaa1036.
[95] Smith, Noah A. 2019. "Contextual Word Representations: A Contextual Introduction," February. http://arxiv.org/abs/1902.06006.
[96] Solan Z, Horn D, Ruppin E, Edelman S. Proc Natl Acad Sci 2005:11629–116344. https://doi.org/10.1073/pnas.0409746102.
[97] Steinegger Martin, Söding Johannes. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. Nat Biotechnol 2017;35(11):1026–8. https://doi.org/10.1038/nbt.3988.
[98] Strait BJ, Dewey TG. The shannon information entropy of protein sequences. Biophys J 1996;71(1):148–55. https://doi.org/10.1016/S0006-3495(96)79210-X.
[99] Strodthoff Nils, Wagner Patrick, Wenzel Markus, Samek Wojciech. UDSMProt: universal deep sequence models for protein classification. Bioinformatics 2020;36(8):2401–9. https://doi.org/10.1093/bioinformatics/btaa003.
[100] Sunarso, Freddie, Srikumar Venugopal, and Federico Lauro. 2013. "Scalable Protein Sequence Similarity Search Using Locality-Sensitive Hashing and MapReduce." ArXiv:1310.0883 [Cs], October. http://arxiv.org/abs/1310.0883.
[101] Sutskever, Ilya, Oriol Vinyals, and Quoc V Le. 2014. "Sequence to Sequence Learning with Neural Networks." In Advances in Neural Information Processing Systems, 3104–12.
[102] Trifonov Edward N. The origin of the genetic code and of the earliest oligopeptides. Res Microbiol 2009;160(7):481–6. https://doi.org/10.1016/j.resmic.2009.05.004.
[103] Varshavsky, Roy, Menachem Fromer, Amit Man, and Michal Linial. 2007. "When Less Is More : Improving Classification of Protein Families with a Minimal Set of Global Features," 12–24.
[104] Vaswani Ashish, Shazeer Noam, Parmar Niki, Uszkoreit Jakob, Jones Llion, Gomez Aidan N, et al. Attention is all you need. Adv Neural Inf Process Syst 2017;30:5998–6008.
[105] Vig, Jesse, Ali Madani, Lav R. Varshney, Caiming Xiong, Richard Socher, and Nazneen Fatema Rajani. 2020. "BERTology Meets Biology: Interpreting Attention in Protein Language Models," June. http://arxiv.org/abs/2006.15222.
[106] Wang Alex, Singh Amanpreet, Michael Julian, Hill Felix, Levy Omer, Bowman Samuel R. Glue: A multi-task benchmark and analysis platform for natural language understanding. ArXiv Preprint ArXiv:1804.07461. 2018.
[107] Wang Yanbin, You Zhu-Hong, Yang Shan, Li Xiao, Jiang Tong-Hai, Zhou Xi. A high efficient biological language model for predicting protein-protein interactions. Cells 2019;8(2):122. https://doi.org/10.3390/cells8020122.
[108] Weathers Edward A, Paulaitis Michael E, Woolf Thomas B, Hoh Jan H. Reduced amino acid alphabet is sufficient to accurately recognize intrinsically disordered protein. FEBS Lett 2004;576(3):348–52. https://doi.org/10.1016/j.febslet.2004.09.036.
[109] Wen Bo, Zeng Wen-Feng, Liao Yuxing, Shi Zhiao, Savage Sara R, Jiang Wen, et al. Deep learning in proteomics. Proteomics 2020;20(21–22). https://doi.org/10.1002/pmic.201900335.
[110] Wu Zachary, Yang Kevin K, Liszka Michael J, Lee Alycia, Batzilla Alina, Wernick David, et al. Signal peptides generated by attention-based neural networks. ACS Synth Biol 2020;9(8):2154–61. https://doi.org/10.1021/acssynbio.0c00219.
[111] Yamada, Ikuya, and Hiroyuki Shindo. 2019. "Neural Attentive Bag-of-Entities Model for Text Classification." In Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL), 563–73. Hong Kong, China: Association for Computational Linguistics. 10.18653/v1/K19-1052.
[112] Yan Renhong, Zhang Yuanyuan, Yaning Li Lu, Xia Yingying Guo, Zhou Qiang. Structural basis for the recognition of sars-cov-2 by full-length human ACE2. Science 2020;367(6485):1444–8. https://doi.org/10.1126/science.abb2762.
[113] Yang, Kevin K, Zachary Wu, Claire N Bedbrook, and Frances H Arnold. 2018. "Learned Protein Embeddings for Machine Learning." Edited by Jonathan Wren. Bioinformatics 34 (15): 2642–48. 10.1093/bioinformatics/bty178.
[114] Yang Zhilin, Dai Zihang, Yang Yiming, Carbonell Jaime, Salakhutdinov Ruslan, Quoc V Le. XLNet: Generalized autoregressive pretraining for language understanding. Advanc Neural Inform Process Sys 2019;32(June).
[115] Yao, Liang, Chengsheng Mao, and Yuan Luo. 2019. "KG-BERT: BERT for Knowledge Graph Completion." ArXiv:1909.03193 [Cs], September. http://arxiv.org/abs/1909.03193.
[116] Yu Fisher, Koltun Vladlen, Funkhouser Thomas. Dilated residual networks. *ArXiv* 2017.
[117] Yu, Lijia, Deepak Kumar Tanwar, Emanuel Diego S. Penha, Yuri I. Wolf, Eugene V. Koonin, and Malay Kumar Basu. 2019. "Grammar of Protein Domain Architectures." Proceedings of the National Academy of Sciences 116 (9): 3636–45. 10.1073/pnas.1814684116.
[118] Yuille, Alan L., and Chenxi Liu. 2020. "Deep Nets: What Have They Ever Done for Vision?" ArXiv:1805.04025 [Cs], November. http://arxiv.org/abs/1805.04025.
[119] Zaheer, Manzil, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, et al. 2020. "Big Bird: Transformers for Longer Sequences." ArXiv:2007.14062 [Cs, Stat], July. http://arxiv.org/abs/2007.14062.
[120] Zhang Chiyuan, Bengio Samy, Hardt Moritz, Recht Benjamin, Vinyals Oriol. Understanding deep learning requires rethinking generalization. *ArXiv* 2017.