nature machine intelligence

REVIEW ARTICLE

https://doi.org/10.1038/s42256-022-00499-z

Check for updates

Controllable protein design with language models

Noelia Ferruz[1,2] ✉ and Birte Höcker[1]

The twenty-first century is presenting humankind with unprecedented environmental and medical challenges. The ability to design novel proteins tailored for specific purposes would potentially transform our ability to respond to these issues in a timely manner. Recent advances in the field of artificial intelligence are now setting the stage to make this goal achievable. Protein sequences are inherently similar to natural languages: amino acids arrange in a multitude of combinations to form structures that carry function, the same way as letters form words and sentences carry meaning. Accordingly, it is not surprising that, throughout the history of natural language processing (NLP), many of its techniques have been applied to protein research problems. In the past few years we have witnessed revolutionary breakthroughs in the field of NLP. The implementation of transformer pre-trained models has enabled text generation with human-like capabilities, including texts with specific properties such as style or subject. Motivated by its considerable success in NLP tasks, we expect dedicated transformers to dominate custom protein sequence generation in the near future. Fine-tuning pre-trained models on protein families will enable the extension of their repertoires with novel sequences that could be highly divergent but still potentially functional. The combination of control tags such as cellular compartment or function will further enable the controllable design of novel protein functions. Moreover, recent model interpretability methods will allow us to open the 'black box' and thus enhance our understanding of folding principles. Early initiatives show the enormous potential of generative language models to design functional sequences. We believe that using generative text models to create novel proteins is a promising and largely unexplored field, and we discuss its foreseeable impact on protein design.

Proteins are the universal building blocks of life, having a vital role in essentially every cellular process. The custom design of specific, efficient and tailored proteins in a fast and cost-effective manner would have the potential to tackle many of the challenges that humankind faces today and will face in the future. For example, we would be able to design enzymes that metabolize plastic waste or hydrolyse polluting toxins, or create new vaccines in a timely fashion in the event of a pandemic. However, despite great advances, contemporary research is still far from designing proteins as proficient as those generated naturally[1].

Protein design seeks to create custom structures that perform a desired function. This enormous challenge has often been referred to as the inverse protein-folding problem: instead of finding the structure into which a sequence folds, the goal is to obtain an optimal sequence that adopts a certain fold. Mathematically, this problem is approached with optimization algorithms that search the global minimum of a sequence–structure landscape defined by an energy function. Despite the relative simplicity of the most widely used energy functions[2], the number of rotamers and possible combinations at each position promotes a combinatorial explosion, and, understandably, most protein design packages rely on heuristic algorithms. As a consequence of this complexity—and despite remarkable recent progress[3]—the design of de novo proteins usually takes considerable time and effort, and the overwhelming majority of functional proteins have materialized by pre-selecting naturally occurring scaffolds and subsequently optimizing their function in iterative rounds, as opposed to concomitantly designing the sequence and structure to perform a certain function[1].

Although the protein design problem has been approached with physicochemical functions that target their structures, one of the most extraordinary properties of proteins is that they entirely encode their structure and function in their amino-acid sequence, and they do so with extreme efficiency. The fact that sequences alone can capture the properties of proteins in the absence of biophysical constraints opens an unexplored door for protein research by exploiting natural language processing (NLP) methods.

The following sections summarize similarities and differences between natural languages and protein sequences and shows how NLP research has already influenced protein science. We will emphasize the most notable development in the field, namely, the transformer architecture. Subsequent sections will introduce how the unique generative capabilities of transformers are reshaping the protein design field. Finally, we will offer a perspective on how they might also dominate the exceptionally challenging cases of non-natural enzymatic reactions and tailored novel functions. We hope this Review reaches both the artificial intelligence and biology fields and encourages further collaborative efforts towards developing and adapting NLP techniques for protein design. A glossary of selected terms is provided in Box 1.

The language of proteins

Several characteristics evidence the similarities between human languages and protein sequences, with perhaps the most obvious being their hierarchical organization. Analogous to human languages, proteins are represented by a concatenation of strings: the 20 standard amino acids. Letters then assemble to form words, and amino acids combine to form secondary structural elements or conserved protein fragments[4]. Then, as words combine to form sentences that carry meaning, fragments can assemble into different protein structures that carry a function (Fig. 1a).

The origin and evolution of languages and proteins also show parallels. Languages grow and continuously adapt, with words emerging that better reflect our evolving society. Today, there are over 8,000 languages divided into more than 140 linguistic families, all of which originated from a common ancestral language spoken in central Africa 50,000–70,000 years ago[5]. Similarly, all organisms living on Earth have

[1]Department of Biochemistry, University of Bayreuth, Bayreuth, Germany. [2]Present address: Institute of Informatics and Applications, University of Girona, Girona, Spain. ✉e-mail: noelia.ferruz-capapey@uni-bayreuth.de

Box 1 | Glossary of selected terms

Autoencoding models: a model that is trained by predicting the input after masking or corrupting some of its tokens, such as a percentage of the words in a sentence or amino acids in a sequence.

Autoregressive models: a model where the current prediction depends only on past behaviour. A model that only depends on the last item of the time series is a Markov process.

Big Fantastic Database: a compendium of 2.5 billion protein sequences from several databases, including UniProt/TrEMBL + Swiss-Prot, clustered using MMseq2 and available at https://bfd.mmseqs.com/.

Corpus (pl. corpora): a collection of large and usually unstructured texts used for training models. Corpus sizes are usually measured in gigabytes or tokens.

Embedding: representation of words—or sentences—in the form of a real-valued vector that encodes the meaning of the word such that the words that are closer in the vector space are similar in meaning.

Energy function: a relationship between the energy of a system as a function of the position of their atoms.

Mutation: an alteration of the amino-acid sequence of a protein as a result of errors during DNA replication, mitosis or meoisis, or other damages to DNA.

Parameters: the internal variables of the model whose values are optimized during training. They are also termed weights.

Peptide: a stretch of amino acids connected by peptide bonds.

Perplexity: a way to evaluate language models based on the uncertainty of the model to predict the next word; in mathematical terms it is the exponentiated average negative log-likelihood of a sequence. It typically applies to language models.

Primary structure or sequence: the sequence of amino acids linked together to form a polypeptide chain.

Protein domain: a region of a full protein (tertiary structure) that is self-stabilizing and folds independently.

Quaternary structure: the arrangement of multiple folded protein chains.

Rosetta energy score: the energy of a biomolecule calculated with an internal energy function, or score developed in Rosetta, a software for macromolecular modelling. The energy function considers the atomic interactions of the three-dimensional (3D) structure.

Secondary structure: the 3D form of local segments of proteins, such as α-helices or β-sheets (Fig. 1a).

SMILES: abbreviation for 'simplified molecular-input line-entry system', a string notation describing a chemical molecular entity.

Tertiary structure or structure: 3D shape of a protein.

Tokenize: the process of breaking a text or sentence into individual linguistic units. It is usually the first step in NLP when modelling data.

a (last universal) common ancestor—LUCA—a microorganism that lived four billion years ago[6], which already contained most modern protein domains, which have developed through evolution.

In human languages, words bear relations and interact with adjacent words in the same way amino acids depend on their sequential surroundings. However, human languages also present long-distance dependencies, that is, dependencies between not strictly linearly adjacent words or morphemes such as subjects across sentences in long texts. This notion is reminiscent of protein structures, where amino acids far apart in the sequence could be interacting in the 3D structure, sometimes crossing domain boundaries. The associations also span other behaviours observed in proteins (Fig. 1b). The detrimental effect of adding or changing one letter in a sentence's meaning is equivalent to a loss of function caused by a single mutation[7]. The possibility of shuffling words while still preserving meaning is comparable to sequence permutation[8]. Finally, the formulation of a grammatically correct but meaningless sentence is analogous to designed protein structures with no apparent function or even dangerous functions, such as amyloid fibrils[9].

However, it is essential to note that protein and human languages also present differences that challenge applying NLP to protein research. We will mention a few examples. First, many human languages offer a clear discernible definition of words in written texts (with one prominent exception being Chinese), but the 'word boundaries' are less evident in proteins. One possibility could be to use the definitions of secondary structural elements (Fig. 1a) or conserved fragments[4]. In either case, the tokenization process would rely on the availability of tertiary structures and computationally more intensive calculations than word tokenization. A second impactful difference is the current lack of understanding of the protein language, similar to our current lack of knowledge of many extinct languages. Although we have the corpora to train the protein language—unlike the case for most extinct languages—the correct interpretation of the generated sequences will remain a challenge, requiring extensive experimental tests to decipher their functionality. Third, protein evolution is also obviously different from the evolution of languages, being subject to randomness and environmental pressure, and with a grammar that unavoidably will contain many irregularities. Although there are phonotactic constraints in language[10]—the succession of sound sequences that are possible—this aspect is more pronounced in proteins, the sequences of which must be compatible with folding into a 3D structure, biasing the patterns that protein language models must learn. Finally, although the amount of human languages with sizeable corpora is limited to a dozen languages, there are currently millions of species on Earth, and, rather than studying the proteins of a particular species, we are most often interested in the general properties of proteins. This fact questions the traditional approach to modelling natural languages, which takes a single sentence as input at a time. Indeed, protein structure prediction methods witnessed a substantial boost in performance when multiple sequence alignments (MSAs)[11] were introduced to predict physical contacts. Similarly, the introduction of MSA input in large neural models has led to the remarkable success of large neural models such as MSA Transformer[12] or AlphaFold[13]. Although reminiscent of the study of

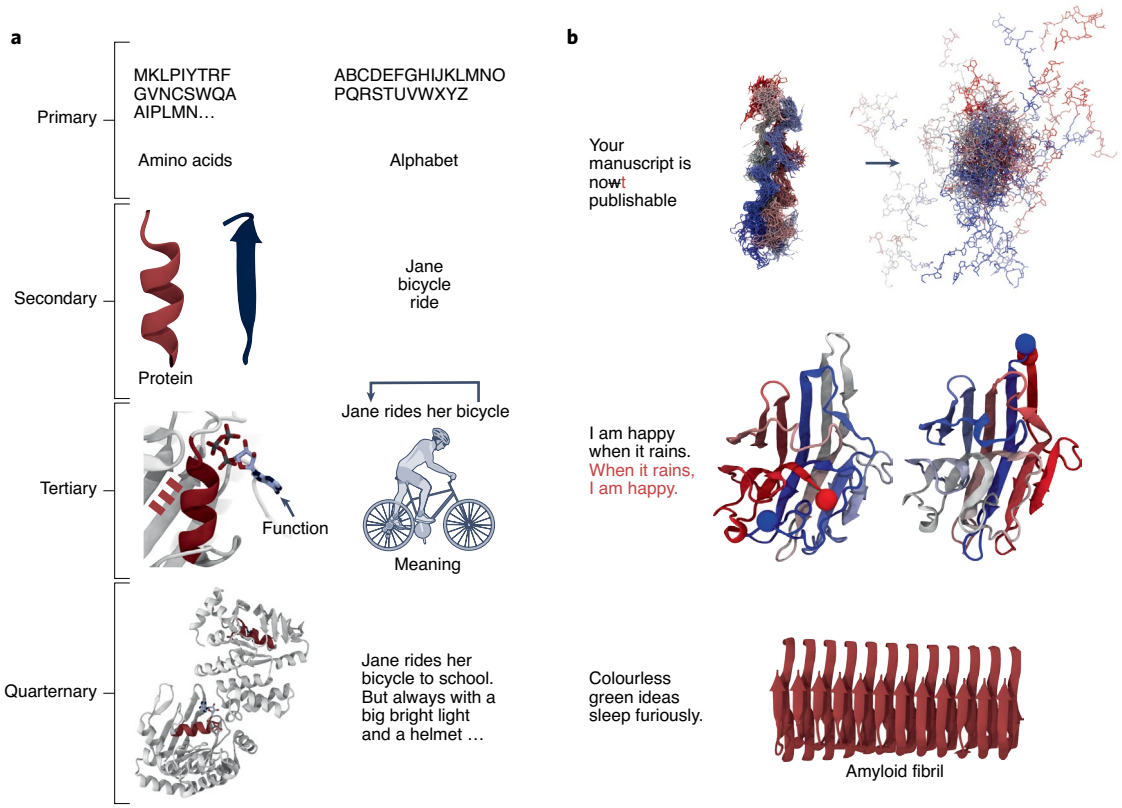

Fig. 1 | Similarities between proteins and languages. a, Protein sequences (primary structure) are represented by a concatenation of characters of their alphabet: the 20 standard amino acids. These amino acids form 3D secondary structural elements such as α-helices and β-sheets, which, like words assembling to form sentences that carry meaning, arrange to form tertiary structures that carry function. Protein domains further assemble to larger quaternary complexes, similar to sentences building text. **b**, The similarities between languages and proteins span other examples. Typos in sentences can be fatal, like missense mutations for protein functionalities. Sentences and sequences can be permuted, retaining their meaning and function, and grammatically correct sentences do not ensure a logical meaning like folded structures do not guarantee functionality.

synonyms[14], the concept of MSAs does not have a direct analogy in NLP methods.

Overall, dissimilarities between human languages and protein sequences pose considerable challenges to the application of NLP to protein design. Nevertheless, the apparent connections between the two fields provide a fresh perspective in the protein research field, despite these challenges. Considering the exceptional environmental and medical challenges of the twenty-first century that humankind is facing, we will require innovative new approaches that transcend disciplinary borders to tackle them. Current design approaches have made impressive advances, but cannot yet deliver solutions that keep pace with the urgency of these problems. Although these approaches will arguably continue to improve, the NLP-based viewpoint creates opportunities to gain complete control over the protein design process.

NLP has had an effect on protein research for decades

We are currently witnessing a revolutionary time in NLP. Software applications such as personal assistants (for example, Apple Siri, Amazon Alexa and Google Assistant), chatbots and translator machines such as Google Translate are reshaping how we interact with machines and go about our daily lives. NLP research has evolved from when analysing a single sentence could take minutes to today's search engines finding millions of websites within milliseconds[15].

Although not readily evident, the field of NLP has always impacted protein research by transferring techniques that arose as solutions to NLP problems to protein sequences. Figure 2a summarizes the parallels between the two fields. For decades, NLP problems were approached with shallow machine learning methods, such as support vector machines (SVMs) or hidden Markov models (HMMs), applied to solve text classification and labelling problems[16]. HMMs and SVMs have also been widely used in classification and labelling problems in proteins, such as fold recognition[17], sequence classification[18] and cell localization[19], and are still the state-of-the-art methods for sequence homology detection[20].

Since the 2010s, however, neural networks started to produce superior results in various NLP tasks (Fig. 2b). Although statistical NLP is now the most popular approach to modelling language tasks, in its beginning it suffered from the curse of dimensionality[15], which led to the motivation of learning representation of words and sentences in a lower dimensionality space[21]. Many advances in this field were based on Tomáš Mikolov's work, who developed the word2vec[22–24] and doc2vec[25] models. These techniques produce a distributed vector representation of a word or text and have been widely used in NLP synonym analyses and as input for larger neural models[15]. Soon, these concepts extended to produce protein sequence embeddings[26,27] (Fig. 2a). Following the popularization of word embeddings, the need arose for more complex functions that extract higher-level features from the input word vectors[15]. The popularity of CNNs among computer vision researchers made them the ideal candidates to accomplish this task, and soon CNNs showed their success in name-entity recognition (NER), part-of-speech (POS) tagging and semantic role labelling[28].

The applicability of CNNs thus soon extended to protein research to predict protein disorder[29], DNA binding sites[30] and fold classification[31]. CNNs, however, failed to model long-distance information,

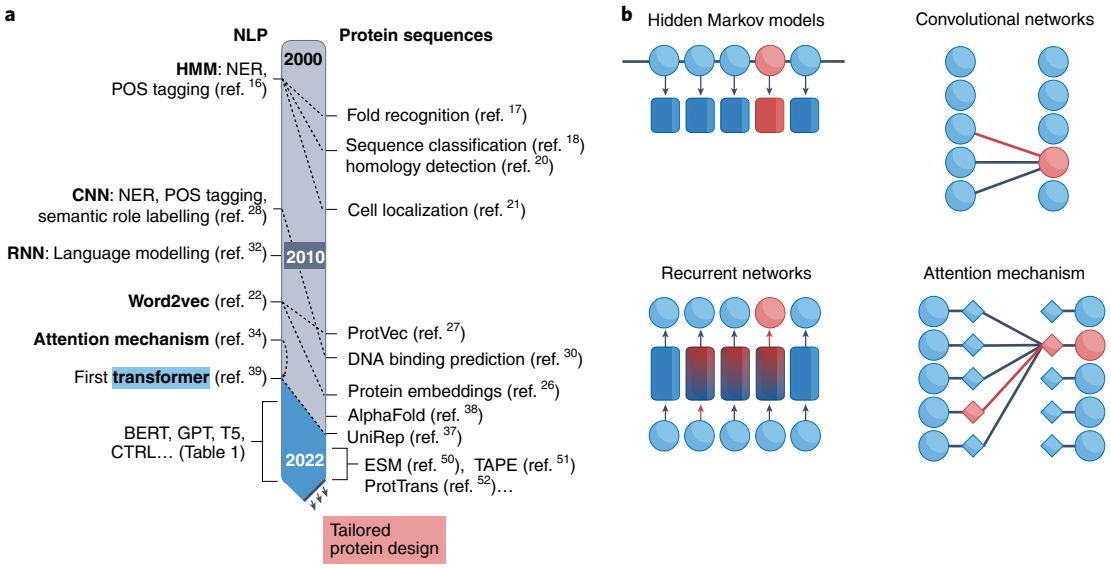

Fig. 2 | Overview of the most commonly used methods for NLP problems. a, Recent timeline of NLP methods and their application in protein research. Each breakthrough in NLP is mirrored years later in protein research applications. **b**, Graphical explanation of the most used methods for NLP. Although hidden Markov models (HMMs) are stochastic processes, convolutional networks (CNNs), recurrent networks (RNNs), and attention mechanisms are, or take part in, neural networks.

which is essential for global text comprehension, or, in the case of proteins, what would be long-range contacts. For this reason, NLP researchers switched to recurrent neural networks (RNNs)[32], in particular long short-term memory (LSTM). RNNs presented superior capabilities in learning long-term dependencies[33] and soon were used to create language models[34,35]. Inspired by their success, Alley et al. utilized a multiplicative LSMT architecture[36] for the protein language model UniRep, which predicted sequence stability with higher accuracy than previous methods[37]. Traditional LSTMs were soon superseded by attention mechanisms[34], influencing recent breakthroughs in protein research such as AlphaFold[38]. Based on the attention model, Google released the transformer[39], improving results in most NLP tasks at a much lower computational cost. The first transformer opened a new era in NLP and, since then, a myriad of adaptations have been implemented (Fig. 2a). It is worth mentioning the generative pre-trained transformer[40] (GPT) and its successors GPT-2[41] and GPT-3[42]. These pre-trained models have shown superior performance in most NLP tasks and, for the first time, were capable of generating human-like, long, coherent articles. These recent developments in the NLP field have a great potential to be adapted to protein research. The following sections will offer insight into how pre-trained language models could transform and dominate protein design in the years to come.

Attention mechanism and transformers

Transformers are a current revolution in NLP. Their success derives from the evolution of a series of concepts built on top of each other, with the attention mechanism possibly being the most notable of these advances.

The attention mechanism originated as a solution to traditional sequence-to-sequence (seq2seq) models, which are widely used for tasks that process sequences from one domain to another, such as machine translation or text summarization. In seq2seq models, the input is stepwise-processed in a module termed encoder to produce a context vector passed to a decoder, which is responsible for generating an output (Fig. 3a). Traditionally, encoder and decoder architectures have usually been RNNs or LSTMs (Fig. 2b), and the context vector corresponded to the final output of the last encoder step (word) (Fig. 2b). Due to this inherently sequential nature, RNNs presented the major drawback of degrading performance and increasing training times with sequence length[15]. The introduction of the attention mechanism provided a solution to this problem by allowing the decoder to analyse the whole input and focus on specific parts of it, a notion similar to attention in the human mind. A simplified example in English–French translation would be focusing on the input word 'home', when outputting the word 'maison' (Fig. 3a).

Although attention mechanisms had been ubiquitously applied in many types of neural networks, they became particularly prominent in 2017, when researchers at Google published their seminal work 'Attention is all you need', which introduced an architecture that not only applied attention between the modules but also throughout them[39]. This new design permitted the encoder to focus on specific parts of the input sequence, producing a much better performance in many tasks[39]. The model was termed the Transformer and gave its name to all similar architectures that followed in subsequent years (Fig. 2a).

The transformer's encoder and decoder modules contain a stack of six submodules or layers (N) that process inputs from the previous layer in a parallel fashion, enabling much faster training times (Fig. 3b). The encoder submodules contain two layers: a self-attention layer, which applies the attention mechanism to the input sequence itself, and a feedforward layer that processes the inputs from the previous layer separately and identically, allowing parallelization. The decoder also comprises six submodules with these same two layers, but with another encoder–decoder attention layer in between that focuses on relevant parts of the input (Fig. 3b).

A summary of the internal workings of the dot-product attention mechanism from the original transformer goes as follows: because sentences are strings, the input needs to be converted to a vector of floats compatible with the internal mathematical operations of the model[26]. This step is termed 'embedding'. The transformer further adds the embeddings $\mathbf{e}$ to positional encoding vectors that offer information about the position of each word. The final vectors are passed to the first encoder submodule, which converts them into Query, Key and Value vectors by multiplying them with matrices obtained during training. The dot-product of the Query vector $\mathbf{q}$ against each word's Key vector $\mathbf{k}$ ($\mathbf{q}_1 \cdot \mathbf{k}_1$, $\mathbf{q}_1 \cdot \mathbf{k}_2$) produces

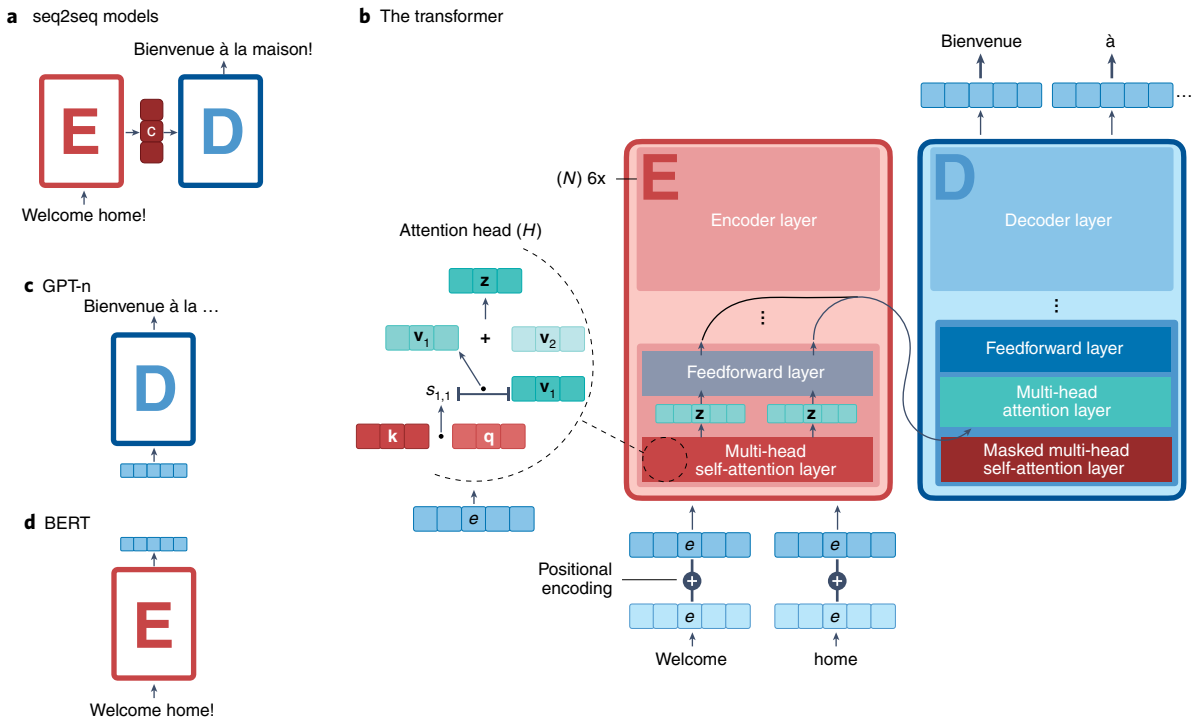

Fig. 3 | Schematic overview of most used transformers. a, The seq2seq models present encoder (E) and decoder (D) models processing sequential inputs that are encoded as context (c) vectors. **b**, The original transformer architecture consisted of encoder and decoder models with stacks of six layers each. An overview of the attention dot mechanism that takes place in the attention head is presented (see main text). **c**,**d**, The GPT-n transformers are based on the original transformer but contain only the decoder model (**c**), whereas BERT uses only the encoder (**d**).

a set of scores ($s_{1,1}$), which are later scaled to the 0–1 range. The score multiplies the corresponding Value vectors ($\mathbf{v}_1$), preserving their magnitude when the score is 1 and minimizing them on the contrary. These vectors are summed up into a final output vector ($\mathbf{z}_1$), which represents a contribution of all the other words in the sentence for each word. In the original implementation, this process was repeated in parallel in eight attention heads (H), expanding the model's capability to focus on different input parts by creating independent Query/Key/Value vectors. A summary of the hyperparameters of this model is presented in Table 1.

Motivated by the transformer architecture, OpenAI released GPT (Generative Pre-trained Transformer), the first of a series of highly performing pre-trained models[40]. The main idea behind GPT consisted of creating a general, task-agnostic language model by training it on a diverse corpus of unlabelled text followed by fine-tuning on labelled datasets to perform specific tasks, thus transferring knowledge from the first step. Despite its general nature, GPT showed significantly improved performance in comparison to state-of-the-art methods in 9 out of 12 tasks studied, with the added advantage that it only required training once[40]. GPT was pre-trained on the classic language modelling task, namely, predicting the next item of a sequence based on the previous ones—a task that makes it particularly powerful for language generation. Models trained on this objective are termed autoregressive and, in the case of transformers, usually their architecture corresponds to the stack of layers from the decoder module (Fig. 3c). However, GPT's generative capabilities did not become evident until the implementation of GPT-2, a model with ten times more parameters and training data[41] (Table 1). GPT-2 showed such an incredible performance at generating coherent text that the authors decided to withhold the model due to the risk of misuse, a decision that was met with controversy[43]. More recently, OpenAI publicized its third-generation GPT model, GPT-3, which contains 100 times more parameters than GPT-2 (Table 1) and is capable of performing well in a zero-shot fashion, even on tasks it has never been trained on, such as code writing[42].

Another prominent development for the NLP field came from the Google AI Language team, who pre-trained BERT (Bidirectional Encoder Representations from Transformers) to create a language model[44]. BERT is also inspired by the transformer architecture, but, given that in this case the interest lies in creating representations of text input, it only uses the encoder module (Fig. 3d). Models like BERT are called denoising autoencoders, which are pre-trained by corrupting the input tokens in some way and trying to reconstruct the original sentence. Although they can also be used for text generation[45], they are most often applied to produce vector representations that can be used for downstream tasks such as classification[46].

In addition to these two representative examples of the encoder and decoder-only architectures of transformers, thousands of transformers have been published in the past three years. Many have become available in the HuggingFace repository[47]. We provide a summary in Table 1.

Protein sequences are ideal candidates for transformers

The amount of digital data generated is growing exponentially. In 2020 alone, we accumulated over 40 zettabytes (ZB) of data, and current estimates set the levels to be more than 80 ZB by 2025[48]. Indeed, the enormous success of the last generation of transformers arises in part because of the ever-growing corpora on which they are trained (Table 1), which, in turn, permits creating larger and more powerful models. Figure 4a shows the transformers presented in Table 1 by their release date and number of parameters (on a logarithmic scale).

This data explosion is, however, not specific to web data. The size of biological databases is also growing considerably, a trend most noticeable for protein sequences. Figure 4b illustrates the data acquisition trends in the past 20 years for sequence and structural

Table 1 | Summary of transformer models

Model	*H*	*N*	*d*	Training set	No. of parameters	Computational time (available information)	Architecture/training objective	Ref.
Transformer	8	6	512	WMT English–German (4.5M sentence pairs) WMT English–French (36M sentence pairs)	~50M[a]	3.5 days 8 NVIDIA P100	Encoder–decoder	39
GPT	12	12	768	BooksCorpus (800M words)	110M	1 month, 8 GPUs	Autoregressive	40
BERT	**16**	**24**	**1,024**	**BooksCorpus, English Wikipedia (2,500M words). Total: 3.3B words or 16 GB**	**340M**	**4 days, 16 TPU Pods (64 TPU chips)**	**Denoising autoencoder**	44
Transformer-XL	16	18	1,024	Several datasets	257M	–	Autoregressive	87
XLM	12	12	2,048	BERT dataset	665M	64 Volta GPUs	Several learning objectives	88
GPT-2	**25**	**48**	**1,600**	**WebText (40 GB or 10B Tokens)**	**1.6B**	**32 TPUv3 (128 chips)**	**Autoregressive**	41
XLNet	**16**	**24**	**1,024**	**BERT dataset + 114 GB additional. Total: 130 GB.**	**340M**	**512 TPU chips, 2.5 days.**	**Autoregressive**	89
RoBERTa	**16**	**24**	**1,024**	**BERT dataset + 144 GB additional. Total: 160 GB**	**255M**	**1,024 V100 GPUs for 1 day (4–5 times more than BERT)**	**Denoising autoencoder**	90
CTRL*	**16**	**48**	**1,280**	**Wikipedia, Project Gutenberg, Amazon Reviews and Reddit. Total: 140 GB**	**1.6B**	**–**	**Autoregressive**	62
Megatron-LM	72	32	3,072	Wikipedia, OpenWebText, RealNews, + CC-Stories.	8.3B	53 min, 512 GPUs	Autoregressive	91
Albert	**64**	**12**	**4,096**	**BERT dataset**	**223M**	**512 TPUv3 chips, 32 h**	**Denoising autoencoder**	92
DistilBERT	6	12	768	BERT dataset	65M	8 V100, 3.5 days (four times less than BERT)	Denoising autoencoder	93
Turing-NLG	78	28	4,256	The Pile[94]	17B	–	Autoregressive	95
Electra	**16**	**24**	**1,024**	**XLnet dataset**	**335M**	**–**	**Generator (autoregressive) and discriminator (Electra: predicting masked tokens)**	96
GPT-3	96	96	12,288	Total: 499B Tokens from Common Crawl, WebText2, Books1, Books2, Wikipedia	175B	36 years, 8 V100 GPUs	Autoregressive	42
T5	**128**	**24**	**1,024**	**The Colossal Clean Crawled Corpus (C4, 750 GB)**	**11B**	**4.68 days, 2,048 A100 GPUs (for T5-3B)**	**Encoder–decoder**	97
Switch	NA, mixture of experts			The Colossal Clean Crawled Corpus (C4, 750 GB)	1.6T	2,048 TPUs, time not specified	Denoising autoencoder	98
Wu Dao 2.0	NA, mixture of experts			1.2 TB Chinese text data in Wu Dao Corpora 2.5 TB Chinese graphic data 1.2 TB English text data in the Pile dataset	1.75T	–	–	–
Megatron-Turing-NLG	105	128	20,480	The Pile	530B	>3 months, 560 DGX 8xA100 servers	Autoregressive	99

All cases report the largest transformer of their series. The reported transformers are ordered by their release date. Bold rows correspond to models that have been applied to protein sequence datasets. Data not known is depicted with a dash. GPU, graphics processing unit; *N*, number of transformer layers; *H*, number of attention heads; *d*, dimension of the model; M, millions; B, billions; T, trillions; TPU, tensor processing unit. [a]Estimated data.

databases, revealing that the characterization of protein sequences is growing at a much faster rate than their counterpart structures. The UniParc database, a comprehensive and non-redundant dataset containing most of the publicly available protein sequences, consists of 485.7 million entries (release 2022_01). UniRef50, a database of cluster representatives filtered at the 50% identity level, contains over 51 million sequences and has more than doubled its number of entries in the past four years.

Although the recent development of high-performing methods for structure prediction, such as AlphaFold[13,38], has enabled

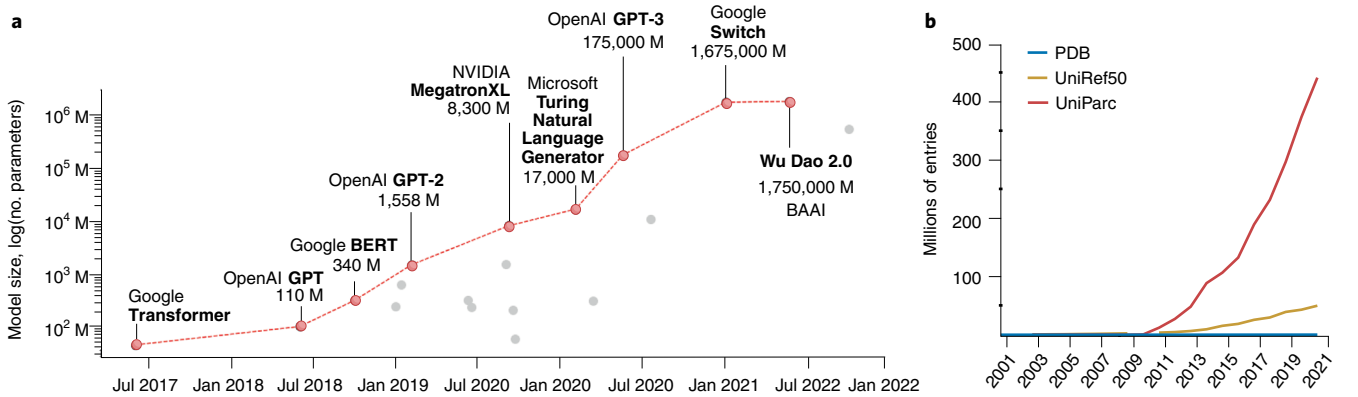

Fig. 4 | Model size and database growth over time. a, Overview of the release date and the number of parameters for the transformers in Table 1. The *y* axis is shown in a logarithmic scale. The largest transformers released at the time are depicted in red, the rest in grey. The numbers of parameters are always shown in millions (M). **b**, Deposited entries in the protein databases PDB, UniParc and UniRef50 in 2001–2021.

scientists to equate the growth of structures with sequences, it does not solve the time-consuming problem of functional annotation. We are thus dealing with a field where the ratio of unlabelled-to-labelled data is increasing extensively (a phenomenon termed the sequence–structure gap), reminiscent of the exponential accumulation of unlabelled corpora on the internet. Given the success of semi-supervised methods such as transformers harnessing scrapped web data to create language models, we can speculate that transformers could similarly exploit the vast protein space and stimulate a similar revolution in the protein research field.

Transformers for protein design

The recent revolutionary developments in NLP are already influencing some pioneering protein research. Inspired by advances from the pre-transformer era, several studies have applied the concept of language models to protein sequences. Yu et al. applied n-gram modelling to generate a probabilistic protein language model[49]. Similar to Radford et al.'s work, where the authors applied multiplicative LSTMs[36] (mLSTM) to learn a language model[35], Alley et al. trained an mLSTM to implement UniRep, a model able to output vector representations of protein sequences[37]. Radford et al. found a single neuron that directly impacted sentiment analysis, and, similarly, Alley et al. identified a single unit that was sensitive to the secondary structure of the input, despite UniRep being trained on sequence data alone. In a similar fashion to BERT, UniRep's vector representations were used in downstream tasks, in this case allowing the prediction of protein sequence stability or function, among others, with state-of-the-art accuracy[37].

The first transformer-based protein language models, ESM[50] and TAPE[51], date to 2019. ESM-1b, an effort led by Facebook AI, is an encoder transformer trained on 250 million protein sequences that has the same architecture and training objective as BERT (Fig. 3d), but, in this case, 33 encoder layers were pre-trained on the UniParc database (Fig. 3d) to produce vector representations that encode protein sequences. ESM-1b's representations, analogous to BERT sentence representations capturing language grammar, encode the internal organization of proteins from the level of biochemical properties of amino acids to that of evolutionary relationships among proteins. Coupling to downstream deep learning models enabled up-to-date predictions of mutational effects and improved predictions for long-range contacts[50].

TAPE (Tasks Assessing Protein Embeddings), on the other hand, is a set of semi-supervised models and datasets that constituted the first attempt to evaluate the performance of protein embeddings across different biological tasks[51]. Five architectures, including a 38-million-parameter transformer, were benchmarked on five tasks that ranged from structure prediction to protein engineering. The authors observed that self-supervision pre-training improved performance in almost all cases, but the performance for each architecture varied substantially depending on the tasks[51].

More recently, a collaboration of scientists from Munich, Nvidia and Google AI led to ProtTrans, an impressive adaptation of six previously released transformer-based architectures (Transformer-XL, BERT, Albert, XLnet, T5 and Electra; Table 1) to the protein domain that is completely accessible to the community (https://github.com/agemagician/ProtTrans). This study utilized the largest training dataset so far, containing over 390 billion amino acids taken from UniParc and the Big Fantastic Database. Their work showed that protein embeddings—the vector representations that the transformers output—are capable of accurately predicting per-residue secondary structure and subcellular localization[52].

These early studies demonstrated the potential of learned protein representations for downstream applications, including classification or regression tasks. Recently, several works were published that use pre-trained models to generate protein sequences. Although not explicitly employing language models, we mention two that shift away from the traditional protein design paradigm—based on searching energy function minima—to a neural network approach. First, inspired by the generative capabilities of DeepDream, Google's CNN that is capable of generating psychedelic images[53,54], Anishchenko et al. applied trRosetta in a stepwise process to generate idealized protein structures in a high-throughput fashion[55]. Second, Huang et al. recently generated SCUBA, a novel adaptive-kernel neighbour counting-neural network (NC-NN) approach that produced de novo structures with novel topologies[56].

Regarding the particular case of language models, several recent works are harnessing transformers for protein design. Castro et al. implemented ReLSO, an autoencoder trained to jointly generate sequences and predict fitness given an input labelled dataset[57]. Moffat et al. implemented DARK, a 110-million-decoder-only transformer capable of designing novel structures[58], and Ferruz et al. released ProtGPT2, a 738 million transformer model based on the GPT-2 architecture, which generates de novo sequences in unexplored regions of the protein space[59].

Although these last models have been focusing on zero-shot de novo sequence design, they also hold the potential to perform other tasks. On the one hand, because their last layer's activations are effectively input representations, the DARK or ProtGPT2 architectures could also be coupled to other models to predict function loss or mutational effects, to cite a few examples (Fig. 5c). On the other hand, generative models can control the direction of the text by fine-tuning them on specific types of data[60], so protein language

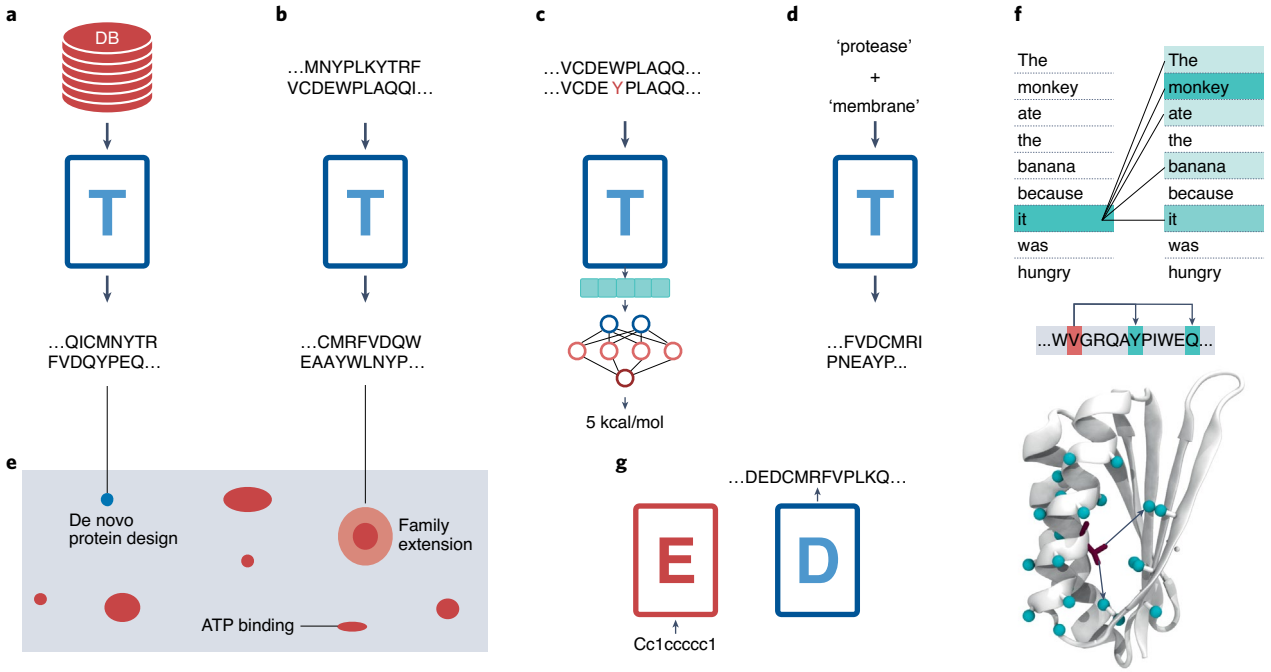

Fig. 5 | Overview of possibilities in the protein engineering field with transformer models. a, After training a transformer (T) with a protein sequence database, it is possible to generate de novo protein sequences (shown in **e**). **b**, Fine-tuning the pre-trained model on a protein family would generate novel sequences compatible with the family. **c**, The last layer's vector representations can be used in a variety of downstream tasks by training them with coupled models, for example, to predict protein stability. **d**, Conditional transformers will be capable of generating sequences with certain properties, such as 'protease' or 'membrane'-bound. **e**, A schematic of protein space, adapted from ref. [100]. **f**, Visualization of the attention mechanism has opened the door to understand transformer models, which, along with other techniques, could be used to understand protein design principles, such as required interactions. **g**, Machine translation models, such as from the original transformer, could enable receptor and enzyme design.

models also have the potential to design a particular family or fold after fine-tuning on a user-defined set of sequences. These advances show the promising future of autoregressive transformers in protein design while opening a new door for protein engineering tasks.

Tailored protein design

The next important step in NLP and its application to custom protein design is the inclusion of functional tags during training. Recently, Gligorijević et al. implemented a denoising autoencoder transformer in which a certain input sequence is translated to an output sequence of superior quality and a certain function[61]. Nevertheless, perhaps one of the most important works towards controllable text generation was the development of the Conditional TRansformer Language (CTRL), an autoregressive model including conditional tags capable of controllably generating text without relying on input sequences[62] (Table 1). These tags, called control codes, allow users to more specifically influence genre, topic or style, among others—an enormous step towards goal-oriented text generation. Shortly after CTRL implementation, the authors adapted this model to a dataset of 281 million protein sequences[63]. The model, named ProGen, contains as conditional tags UniparKB Keywords, a vocabulary of ten categories including 'biological process', 'cellular component' or 'molecular function'. In total, the conditional tags comprised more than 1,100 terms. ProGen presented 'perplexities' representative of high-quality English language models, even on protein families not present in the training set. Generation of random sequences and their Rosetta energy evaluation revealed that the sequences had better scores than random ones. The authors analysed ProGen's capabilities to complete a truncated kinase domain and showed that all completed proteins remained close to the Rosetta score of the native protein. As the last test for generative capabilities, several protein G-binding domain variants passed through ProGen, and selection of the top 100 variants with the lowest perplexity values provided better fitness scores than random mutations. In a later application of this work, the authors applied ProGen to the generation of lysozymes after fine-tuning on five different protein families. Experimental validation showed that the generated sequences possess enzymatic activities in the range of natural lysozymes, and X-ray characterization of one of the variants showed that it recapitulated the native 3D structure[64].

These studies highlight a promising new area of research: the controllable generation of protein sequences with conditional transformers. The inclusion of conditional tags in transformer-based protein language models will not only enable the generation of novel sequences as in these previous works, but could potentially also provide control over the properties of these proteins. We will mention here a few possibilities.

First, we envision directly generating sequences that have a property included in the training set, such as binding ATP, folding into an all-beta structure or being membrane-bound (Fig. 5d). Second, it would be important to investigate property tags that appear in several regions across protein space, such as 'membrane protein' or 'ATP binding' (Fig. 5e). The output sequences could perhaps render so far unknown solutions for these properties—proteins in unexplored regions of the sequence space (Fig. 5e)—and provide the means to understand their structural requisites for these functional characteristics. Finally, conditional transformers would enable the tailored design of proteins with novel functions. Analogous to how the combination of control tags, such as style + topic ('poetry' + 'politics'), provides specific text, the fusion of protein properties could create novel functions, such as 'hydrolase' + 'PET binding' or 'membrane bound' + 'protease'.

The ability to generate tailored sequences by prompting a language model would be a transformative milestone in protein research, but its implementation is not without challenges.

The process of supervised sequence labelling relies on the quality of annotated sequences. Recently, Rembeza et al. analysed the BRENDA database and found that nearly 78% of sequences in the EC 1.1.3.15 enzyme class are misannotated, and around 18% of all sequences are classified to an enzyme class with few similarities to the class representatives[65]. Besides, annotation is a considerably time-consuming process: the widely used protein family database Pfam has grown by less than 5% over the past five years, and at least one-third of microbial proteins has not been annotated through alignment to functionally characterized sequences[66]. However, recent works in sequence annotation might open a new door to faster automatable annotation processes. Bileschi et al. recently used neural networks to predict functional annotations, extending the Pfam database by >9.5%[67]. Although it might be a lengthy and challenging process, efforts in this direction may lead to the annotation of a substantial part of protein space in the foreseeable future, thereby facilitating the implementation of conditional transformers.

Enzyme, receptor and biosensor design

In 2018, IBM Research released IBM RXN for Chemistry, a cloud-based app that relates organic chemistry to human language[68]. The app hosts the Molecular Transformer, a model that can predict the most likely outcome of a chemical reaction using an encoder–decoder architecture (Fig. 3a). In this case, the model encoder processes chemical species as input (reactants + reagents), whereas the decoder outputs the most likely reaction products[68]. Subsequently, the authors reversed the network: instead of predicting the outcome of a possible chemical reaction, the problem consisted of determining the reactants needed to create a given target molecule, a process termed retrosynthesis[69]. Following a similar approach, Grechishnikova implemented an encoder–decoder architecture for de novo drug generation[70]. In this case, the encoder processes protein sequences while the decoder generates SMILES of ligands that are potentially compatible with binding the input sequence.

These two examples show how models based on the original transformer are powerful tools for generating outputs conditioned on an entry input. In particular, Grechishnikova's approach is interesting for the protein design realm; by reversing the translation machine we might be capable of generating sequences compatible with the encoder input SMILES (Fig. 5f). Such a model could have tremendous applications for engineering of receptor proteins, including the prediction of sequences for the recognition and binding of specific ligands, a big step forward for receptor and biosensor design. Given the recent approach by IBM to encode vector representations of chemical reactions[71], we could envision another model that takes chemical reactions as input and produces protein sequences as output. Such a model would provide an innovative route for enzyme design, including engineering enzymes capable of catalysing reactions not found in nature. This approach could potentially support biological strategies, for example, to reverse environmental pollution.

Explainable protein design

The design of proteins with customizable properties is a long-standing goal in biochemistry. On a more fundamental level, there is also interest in understanding the principles that relate sequences to protein structures, which would enable the rational design of funnel-shaped protein-folding energy landscapes[72]. For this reason, there is a growing interest in providing interpretations for the underlying mathematical workings of deep learning models in a way that is understandable to the human mind. Explainable artificial intelligence (XAI) would help us understand why models reach a particular answer and lead scientists to new ideas and approaches. Research in the drug discovery field is already benefitting from the application of XAI techniques, for example, to identify ligand pharmacophores that drive a molecule's activity[73].

Traditionally, the most widely used NLP techniques, such as HMMs or SVMs (Fig. 2), were inherently explainable and are therefore attributed the term 'white box'. The recent explosion of deep learning methods reaching high performance across NLP tasks has brought the challenge of developing new techniques to explain these models. Substantial progress has been made on XAI techniques for 'black box' models, among which the five main techniques are feature importance, surrogate model, example-driven, provenance-based and declarative induction. For a recent review covering these techniques in the NLP domain, see ref. [74].

For the particular case of transformers, the use of the attention mechanism throughout their architectures provides advantages for explaining their internal representations. The attention mechanism itself corresponds to an importance score over the input features, which allows visualizing the raw scores as a saliency heatmap. Figure 5f exemplifies the self-attention for a sentence where a particular attention layer has attributed several attention scores between the word 'it' and others. In an analogous fashion, protein sequences would correspond to a representation of attention scores among the amino acids (Fig. 5f). Recently, efforts have been made to bring XAI for transformers into user-friendly interfaces. For example, exBERT (https://exbert.net)[75] enables visualization of internal representations for any transformer trained on any corpus. It is possible to visualize self-attention user-defined sentences for all the different attention layers, select specific words and visualize the network part-of-speech prediction at each layer, or search them over the training corpus showing the highest-similarity matches. An adaptation of exBERT to a protein-trained transformer would enable interactive visualization of relationships among amino acids in a protein and, similar to POS tags, their predicted properties. Similarly, searching protein fragments in the training corpus and finding the highest-similarity matches could illuminate new relationships between proteins. Although this field is still in its infancy, the possibility of visualizing the internal workings of transformers could bring great opportunities to better understand protein folding and design.

Is the future of protein design in the hand of big companies?

The landscape of transformer models published in recent years is dominated by big companies (Fig. 4a). Training GPT-3 with 175 billion parameters—the second-largest model so far—was estimated to have cost US$12 million and required over 10,000 days of GPU time[76]. Other models have been trained by accessing considerable TPU resources. Training such deep learning models is a commodity that might be accessible to large companies such as OpenAI or Google, but is potentially beyond the reach of start-ups and many academic research groups. Their economic accessibility is a concern, and the carbon footprint associated with training such AI models is drawing growing attention[77]. Although there is increasing awareness of these possible problems associated with AI, the truth is that models perform considerably better with increasing size[78], and model sizes will most predictably only continue to grow: the amount of computing used in the largest AI training runs has been increasing exponentially this year at a 3.4-month doubling time rate[79].

This has obvious repercussions for protein research and academic groups. Seven out of nine protein-based transformer models published so far (Table 1) correspond to efforts led by or including large companies. Although this might sound like a troubling prospect for academic groups and the overall future of this rapidly evolving field, this does not necessarily create an imbalance.

First, large transformer models have the advantage of only requiring training once and can then be used for a wide variety of downstream tasks, suggesting that the research community would still benefit after public release. Examples of this are efforts including AlphaFold and ProtTrans, but, unfortunately, public release is

not always provided. Moreover, although the protein-based published transformers are linked to big companies' efforts, in all cases they involved collaborations with academic groups, a trend that, if extended in the future, might bring new opportunities to academia and create a more collaborative research community, with science, and ultimately society, benefitting from the funding opportunities brought by large companies. Finally, although large language models tend to perform better, there have also been efforts to implement equally performing models with lesser computational resources, such as DistilBERT, which retains 97% of the performance of BERT while reducing its size by 40%, and Switch, which shows an up to seven times increase in pre-training speed (with the same computational resources) compared with T5. These last examples are reminiscent of the analysis of long-timescale molecular dynamics, which initially was only accessible to companies with costly, specialized hardware like ANTON[80], but soon became accessible to the whole research community with the use of in-house GPU clusters and elegant algorithmic solutions[81,82] In this sense, it is important to again emphasize the differences between human languages and protein sequences. Although directly re-applying NLP language models is already showing enormous success, tailoring models to the specific properties of proteins, such as biases due to their necessity to form a 3D structure, may provide increased performance with a decrease in computational cost. One example in this direction is the MSA Transformer[12], which uses a row and column attention across the MSA and takes as input an MSA requiring only 100 million parameters, to offer similar performance as transformers that are six times its size. Another example is AlphaFold 2, which introduced triangle self-attention blocks that better leverage the relationships between sequence and 3D structure. Although training these methods is still costly, they highlight that further engineering of NLP models to the specific properties of protein sequences might provide affordable models with superior performance.

Conclusion

The recent developments in the NLP field and its potential applications to protein sequences are opening exciting new doors for protein research and the design of customizable proteins. Transformer-based language models have served a variety of tasks, including translating natural language, or even writing code to train machine learning models. Moreover, these new models have been capable of generating text so similarly to humans that, since their inception, they have been surrounded by controversy, often not being released due to concerns about potential misuse in the form of fake news or unethical medical advice[43]. Regardless of these discourses, these examples clearly show the incredible potential of transformers. Given the similarities between language and protein sequences, the protein research field will undoubtedly benefit from this transformational new technology[83].

We envision six direct applications from transferring current NLP methods to the protein research domain, as summarized in the previous sections and illustrated in Fig. 5. Ordered by how readily applicable current NLP transformers are to protein sequences, we could (1) generate sequences in unobserved regions of protein space, (2) fine-tune sequences of natural protein families to extend their repertoires, (3) utilize their encoded vector representations as input for other downstream models for protein engineering tasks, (4) generate conditional sequences with specific functional properties, (5) design completely novel and purpose-driven receptors and enzymes using encoder–decoder transformers and (6) gain a more complete understanding of sequence–structure–function relationships, including the rules that govern protein folding by interpreting these language models. Undoubtedly, these advances are not without their challenges, with both the size of the models and the difficulties in functional annotation being two of the most noteworthy. Besides, as pointed out in earlier studies[51,84,85], benchmarks will be paramount to compare model performance, which is particularly challenging in the case of sequence generation. Most generative models thus far have been assessed in the context of their secondary structural content, globularity or similarity to natural sequences[58,59]. Nevertheless, a proper assessment of the generated sequences will eventually require the implementation of high-throughput experimental characterizations. As performed in previous work[86], an assessment of the expressability of these sequences will be essential. Besides, it will ultimately be crucial to assess that these sequences' associated functions, such as their catalytic activities, surpass current protein engineering strategies—possibly in iterative rounds where experimental feedback improves the models. Despite these difficulties, we believe that transformer-based protein language models will revolutionize the field of protein design and provide novel solutions for many current and future societal challenges. We hope that our ideas reach both the artificial intelligence and biochemistry communities and encourage the application of NLP methods to protein research.

Received: 9 September 2021; Accepted: 6 May 2022;
Published online: 22 June 2022

References

1. Lechner, H., Ferruz, N. & Höcker, B. Strategies for designing non-natural enzymes and binders. *Curr. Opin. Chem. Biol.* **47**, 67–76 (2018).
2. Gainza, P., Nisonoff, H. M. & Donald, B. R. Algorithms for protein design. *Curr. Opin. Struct. Biol.* **39**, 16–26 (2016).
3. Kuhlman, B. & Bradley, P. Advances in protein structure prediction and design. *Nat. Rev. Mol. Cell Biol.* **20**, 681–697 (2019).
4. Ferruz, N. et al. Identification and analysis of natural building blocks for evolution-guided fragment-based protein design. *J. Mol. Biol.* **432**, 3898–3914 (2020).
5. W, E. et al. Molecular evolution of FOXP2, a gene involved in speech and language. *Nature* **418**, 869–872 (2002).
6. Theobald, D. L. A formal test of the theory of universal common ancestry. *Nature* **465**, 219–222 (2010).
7. Arena, S. et al. Emergence of multiple EGFR extracellular mutations during cetuximab treatment in colorectal cancer. *Clin. Cancer Res.* **21**, 2157–2166 (2015).
8. Lindqvist, Y. & Schneider, G. Circular permutations of natural protein sequences: structural evidence. *Curr. Opin. Struct. Biol.* **7**, 422–427 (1997).
9. Huang, P. S. et al. De novo design of a four-fold symmetric TIM-barrel protein with atomic-level accuracy. *Nat. Chem. Biol.* **12**, 29–34 (2016).
10. Freeman, M. R., Blumenfeld, H. K. & Marian, V. Phonotactic constraints are activated across languages in bilinguals. *Front. Psychol.* **7**, 702 (2016).
11. Göbel, U., Sander, C., Schneider, R. & Valencia, A. Correlated mutations and residue contacts in proteins. *Proteins Struct. Funct. Bioinformatics* **18**, 309–317 (1994).
12. Rao, R. M. et al. MSA Transformer. In *Proc. 38th International Conference on Machine Learning* Vol. 139, 8844–8856 https://proceedings.mlr.press/v139/rao21a.html (MLR, 2021).
13. Jumper, J. et al. Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583–589 (2021).
14. Nguyen, K. A., Im Walde, S. S. & Vu, N. T. Distinguishing antonyms and synonyms in a pattern-based neural network. In *Proc. 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017* Vol. 1, 76–85 (Association for Computational Linguistics, 2017).
15. Young, T., Hazarika, D., Poria, S., and Cambria, E. Recent Trends in Deep Learning Based Natural Language Processing" in IEEE Computational Intelligence Magazine, Vol. 13, no. 3, 55-75, (2018).
16. Zhou, G. & Su, J. Named entity recognition using an HMM-based chunk tagger. In *Proc. 40th Annual Meeting of the Association for Computational Linguistics, ACL '02* 473–480 https://doi.org/10.3115/1073083.1073163 (Association for Computational Linguistics, 2001).
17. Karchin, R., Cline, M., Mandel-Gutfreund, Y. & Karplus, K. Hidden Markov models that use predicted local structure for fold recognition: alphabets of backbone geometry. *Proteins Struct. Funct. Genet.* **51**, 504–514 (2003).
18. Yakhnenko, O., Silvescu, A. & Honavar, V. Discriminatively trained Markov model for sequence classification. In *Proc. IEEE International Conference on Data Mining*, *ICDM* 498–505 https://doi.org/10.1109/ICDM.2005.52 (IEEE, 2005).
19. Nguyen Ba, A. N., Pogoutse, A., Provart, N. & Moses, A. M. NLStradamus: a simple hidden Markov model for nuclear localization signal prediction. *BMC Bioinformatics* **10**, 202 (2009).

20. Söding, J. Protein homology detection by HMM-HMM comparison. *Bioinformatics* **21**, 951–960 (2005).
21. Bengio, Y. et al. A neural probabilistic language model. *J. Mach. Learn. Res.* **3**, 1137–1155 (2003).
22. Mikolov, T., Chen, K., Corrado, G. & Dean, J. Efficient estimation of word representations in vector space. In *Proc. 1st International Conference on Learning Representations*, *ICLR 2013* (ICLR, 2013).
23. Mikolov, T., Chen, K., Corrado, G. & Dean, J. Distributed representations of words and phrases and their compositionality. In *Proc. 26th International Conference on Neural Information Processing Systems* Vol. 2, 3111–3119 (ACM, 2013).
24. Mikolov, T., Yih, W.-T. & Zweig, G. *Linguistic Regularities in Continuous Space Word Representations* http://research.microsoft.com/en- (Microsoft, 2013).
25. Young, T., Hazarika, D., Poria, S. & Cambria, E. Recent trends in deep learning based natural language processing. *IEEE Comput. Intell. Mag.* **13**, 55–75 (2017).
26. Yang, K. K., Wu, Z., Bedbrook, C. N. & Arnold, F. H. Learned protein embeddings for machine learning. *Bioinformatics* **34**, 2642–2648 (2018).
27. Asgari, E. & Mofrad, M. R. K. Continuous distributed representation of biological sequences for deep proteomics and genomics. *PLoS ONE* **10**, e0141287 (2015).
28. Collobert, R. & Weston, J. A unified architecture for natural language processing. In *Proc. 25th International Conference on Machine learning, ICML '08* 160–167 https://doi.org/10.1145/1390156.1390177 (ACM, 2008).
29. Wang, S., Weng, S., Ma, J. & Tang, Q. DeepCNF-D: predicting protein order/disorder regions by weighted deep convolutional neural fields. *Int. J. Mol. Sci.* **16**, 17315–17330 (2015).
30. Zeng, H., Edwards, M. D., Liu, G. & Gifford, D. K. Convolutional neural network architectures for predicting DNA-protein binding. *Bioinformatics* **32**, i121–i127 (2016).
31. Hou, J., Adhikari, B. & Cheng, J. DeepSF: deep convolutional neural network for mapping protein sequences to folds. *Bioinformatics* **34**, 1295–1303 (2018).
32. Mikolov, T. et al. Recurrent neural network based language model. In *Proc. 11th Annual Conference of the International Speech Communication Association* 1048–1048 (ISCA, 2010).
33. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K. & Dyer, C. Neural architectures for named entity recognition. In *Proc. 2016 Conference of the North American Chapter of the Association for Computational Linguistics*: *Human Language Technologies* 260–270 (Association for Computational Linguistics, 2016).
34. Bahdanau, D., Cho, K. H. & Bengio, Y. Neural machine translation by jointly learning to align and translate. In *Proc. 3rd International Conference on Learning Representations*, *ICLR 2015* (ICLR, 2015).
35. Radford, A., Jozefowicz, R. & Sutskever, I. Learning to generate reviews and discovering sentiment. Preprint at https://arxiv.org/abs/1704.01444 (2017).
36. Krause, B., Murray, I., Renals, S. & Lu, L. Multiplicative LSTM for sequence modelling. In *Proc. 5th International Conference on Learning Representations, ICLR 2017* https://doi.org/10.48550/arxiv.1609.07959 (ICLR, 2016).
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. Senior, A. W. et al. Improved protein structure prediction using potentials from deep learning. *Nature* **577**, 706–710 (2020).
39. Vaswani, A. et al. Transformer: attention is all you need. In *Advances in Neural Information Processing Systems* Vol. 2017, 5999–6009 (NIPS, 2017).
40. Radford, A. & Narasimhan, K. Improving language understanding by generative pre-training; https://openai.com/blog/language-unsupervised/ (2018).
41. Radford, A. et al. *Language Models are Unsupervised Multitask Learners* (GitHub); https://github.com/codelucas/newspaper
42. Brown, T. B. et al. Language models are few-shot learners. Preprint at https://arxiv.org/abs/2005.14165 (2020).
43. Mak, A. When is technology too dangerous to release to the public? *Slate* https://slate.com/technology/2019/02/openai-gpt2-text-generating-algorithm-ai-dangerous.html (22 February 2019).
44. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proc. 2019 Conference of the North American Chapter of the Association for Computational Linguistics*: *Human Language Technologies* Vol. 1 4171–4186 (Association for Computational Linguistics, 2018).
45. Wang, A. & Cho, K. BERT has a mouth, and it must speak: BERT as a Markov random field language model. In *Proc. Workshop on Methods for Optimizing and Evaluating Neural Language Generation* 30–36 (ACL, 2019).
46. Sun, C., Qiu, X., Xu, Y. & Huang, X. in *Lecture Notes in Computer Science* Vol. 11856, 194–206 (Springer, 2019).
47. Wolf, T. et al. ransformers: state-of-the-art natural language processing. in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. 38-45 (2020).
48. *Total Data Volume Worldwide 2010–2025* (Statista); https://www.statista.com/statistics/871513/worldwide-data-created/
49. Yu, L. et al. Grammar of protein domain architectures. *Proc. Natl Acad. Sci. USA* **116**, 3636–3645 (2019).
50. 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).
51. Rao, R. et al. Evaluating protein transfer learning with TAPE. *Adv. Neural Inf. Process. Syst* **32**, 9689–9701 (2019).
52. Elnaggar, A. et al. ProtTrans: towards cracking the language of life's code through self-supervised learning. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1-1 (2019).
53. Ferruz, N. & Höcker, B. Dreaming ideal protein structures. *Nat. Biotechnol.* https://doi.org/10.1038/s41587-021-01196-9 (2022).
54. Mordvintsev, A. DeepDream—a code example for visualizing neural networks. *Google Research Blog* https://web.archive.org/web/20150708233542/http://googleresearch.blogspot.co.uk/2015/07/deepdream-code-example-for-visualizing.html
55. Anishchenko, I. et al. De novo protein design by deep network hallucination. *Nature* **600**, 547–552 (2021).
56. Huang, B. et al. A backbone-centred energy function of neural networks for protein design. *Nature* **602**, 523–528 (2022).
57. Castro, E. et al. Guided generative protein design using regularized transformers. Preprint at https://arxiv.org/abs/2201.09948 (2022).
58. Moffat, L., Kandathil, S. M. & Jones, D. T. Design in the DARK: learning deep generative models for de novo protein design. Preprint at https://www.biorxiv.org/content/10.1101/2022.01.27.478087v1 (2022).
59. Ferruz, N., Schmidt, S. & Höcker, B. A deep unsupervised language model for protein design. Preprint at https://www.biorxiv.org/content/10.1101/2022.03.09.483666v1 (2022).
60. Lee, J. S. & Hsiang, J. Patent claim generation by fine-tuning OpenAI GPT-2. *World Pat. Inf.* **62**, 101983 (2020).
61. Gligorijević, V. et al. Function-guided protein design by deep manifold sampling. in *Neural Information Processing Systems* (NeurIPS, 2021).
62. Keskar, N. S., McCann, B., Varshney, L. R., Xiong, C. & Socher, R. CTRL: a conditional transformer language model for controllable generation. Preprint at https://arxiv.org/abs/1909.05858 (2019).
63. Madani, A. et al. ProGen: language modeling for protein generation. Preprint at https://www.biorxiv.org/content/10.1101/2020.03.07.982272v2 (2020).
64. Madani, A. et al. Deep neural language modeling enables functional protein generation across families. Preprint at https://www.biorxiv.org/content/10.1101/2021.07.18.452833v1 (2021).
65. Rembeza, E. & Engqvist, M. K. M. Experimental and computational investigation of enzyme functional annotations uncovers misannotation in the EC 1.1.3.15 enzyme class. *PLoS Comput. Biol.* **17**, e1009446 (2021).
66. Chang, Y. C. et al. COMBREX-DB: an experiment centered database of protein function: knowledge, predictions and knowledge gaps. *Nucleic Acids Res.* **44**, D330–D335 (2016).
67. 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).
68. Schwaller, P., Gaudin, T., Lányi, D., Bekas, C. & Laino, T. 'Found in Translation': predicting outcomes of complex organic chemistry reactions using neural sequence-to-sequence models. *Chem. Sci.* **9**, 6091–6098 (2018).
69. Schwaller, P. et al. Predicting retrosynthetic pathways using transformer-based models and a hyper-graph exploration strategy. *Chem. Sci.* **11**, 3316–3325 (2020).
70. Grechishnikova, D. Transformer neural network for protein-specific de novo drug generation as a machine translation problem. *Sci. Rep.* **11**, 321 (2021).
71. Schwaller, P. et al. Mapping the space of chemical reactions using attention-based neural networks. *Nat. Mach. Intell.* **3**, 144–152 (2021).
72. Koga, N. et al. Principles for designing ideal protein structures. *Nature* **491**, 222–227 (2012).
73. Jiménez-Luna, J., Grisoni, F. & Schneider, G. Drug discovery with explainable artificial intelligence. *Nat. Mach. Intell.* **2**, 573–584 (2020).
74. Danilevsky, M. et al. A survey of the state of explainable AI for natural language processing. In *Proc. 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing* 447–459 (Association for Computational Linguistics, 2020).
75. Hoover, B., Strobelt, H. & Gehrmann, S. exBERT: a visual analysis tool to explore learned representations in transformer models. In *Proc 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations* 187–196 (Association for Computational Linguistics, 2019); https://doi.org/10.18653/v1/2020.acl-demos.22

76. OpenAI's massive GPT-3 model is impressive, but size isn't everything. *VentureBeat* https://venturebeat.com/2020/06/01/ai-Junemachine-learning-openai-gpt-3-size-isnt-everything/ (1 June 2020).
77. Dhar, P. The carbon impact of artificial intelligence. *Nat. Mach. Intell.* **2**, 423–425 (2020).
78. Li, Z. et al. Train large, then compress: rethinking model size for efficient training and inference of Transformers. In *Proc. 37th International Conference on Machine Learning ICML 2020* 5914–5924 (ICML, 2020).
79. AI and Compute; https://openai.com/blog/ai-and-compute/
80. Shaw, D. E. et al. Anton, a special-purpose machine for molecular dynamics simulation. *Commun. ACM* **51**, 91–97 (2008).
81. Buch, I., Giorgino, T. & De Fabritiis, G. Complete reconstruction of an enzyme-inhibitor binding process by molecular dynamics simulations. *Proc. Natl Acad. Sci. USA* **108**, 10184–10189 (2011).
82. Ferruz, N., Harvey, M. J., Mestres, J. & De Fabritiis, G. Insights from fragment hit binding assays by molecular simulations. *J. Chem. Inf. Model.* **55**, 2200–2205 (2015).
83. Yang, K. K., Wu, Z. & Arnold, F. H. Machine-learning-guided directed evolution for protein engineering. *Nat. Methods* **16**, 687–694 (2019).
84. Chu, S. K. S. & Siegel, J. Predicting single-point mutational effect on protein stability. In *Proc. 35th Conference on Neural Information Processing Systems* (NIPS, 2021).
85. Hsu, C., Nisonoff, H., Fannjiang, C. & Listgarten, J. Combining evolutionary and assay-labelled data for protein fitness prediction. Preprint at https://www.biorxiv.org/content/10.1101/2021.03.28.437402v1 (2021).
86. Baran, D. et al. Principles for computational design of binding antibodies. *Proc. Natl Acad. Sci. USA* **114**, 10900–10905 (2017).
87. Dai, Z. et al. Transformer-XL: attentive language models beyond a fixed-length context. In *Proc. 57th Annual Meeting for the Association for Computational Linguistics* 2978–2988 (ACL, 2019).
88. Lample, G. & Conneau, A. Cross-lingual language model pretraining. *Adv. Neural Inf. Process. Syst.* **32**, 7057–7067 (2019).
89. Yang, Z. et al. XLNet: Generalized autoregressive pretraining for language understanding. In *Proc. 33rd International Conference on Neural Information Processing Systems* Vol. 517, 5753–5763 (ACM, 2019).
90. Liu, Y. et al. RoBERTa: a robustly optimized BERT pretraining approach. Preprint at https://arxiv.org/abs/1907.11692 (2019).
91. Shoeybi, M. et al. Megatron-LM: training multi-billion parameter language models using model parallelism. Preprint at https://arxiv.org/abs/1909.08053 (2019).
92. Lan, Z. et al. ALBERT: a lite BERT for self-supervised learning of language representations. Preprint at https://arxiv.org/abs/1909.11942 (2019).
93. Sanh, V., Debut, L., Chaumond, J. & Wolf, T. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. Preprint at https://arxiv.org/abs/1910.01108 (2019).
94. Gao, L. et al. The Pile: an 800-GB dataset of diverse text for language modeling. Preprint at https://arxiv.org/abs/2101.00027 (2020).
95. Rasley, J., Rajbhandari, S., Ruwase, O. & He, Y. DeepSpeed: system optimizations enable training deep learning models with over 100 billion parameters. In *Proc. 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 3505–3506 https://doi.org/10.1145/3394486.3406703 (ACM, 2020).
96. Clark, K., Luong, M.-T., Brain, G., Le Google Brain, Q. V. & Manning, C. D. ELECTRA: pre-training text encoders as discriminators rather than generators. Preprint at https://arxiv.org/abs/2003.10555 (2020).
97. Raffel, C. et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **21**, 1–67 (2020).
98. Fedus, W., Brain, G., Zoph, B. & Shazeer, N. Switch transformers: scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.* **23**, 1–39 (2022).
99. Smith, S. et al. Using DeepSpeed and Megatron to train Megatron-Turing NLG 530B, a large-scale generative language model. Preprint at https://arxiv.org/abs/2201.11990 (2022).
100. Huang, P. S., Boyken, S. E. & Baker, D. The coming of age of de novo protein design. *Nature* **537**, 320–327 (2016).

Acknowledgements

N.F. acknowledges support from an AGAUR Beatriu de Pinós MSCA-COFUND Fellowship (project 2020-BP-00130).

Competing interests

The authors declare no competing interests.

Additional information

Correspondence should be addressed to Noelia Ferruz.

Peer review information *Nature Machine Intelligence* thanks the anonymous reviewers 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.