Cell Systems

Synthesis

Learning the protein language: Evolution, structure, and function

Highlights

- Deep protein language models can learn information from protein sequence
- They capture the structure, function, and evolutionary fitness of sequence variants
- They can be enriched with prior knowledge and inform function predictions
- They can revolutionize protein biology by suggesting new ways to approach design

Authors

Tristan Bepler, Bonnie Berger

Correspondence

tbepler@nysbc.org (T.B.), bab@mit.edu (B.B.)

In brief

In this synthesis, Bepler and Berger discuss recent advances in protein language modeling and their applications to downstream protein property prediction problems. They consider how these models can be enriched with prior biological knowledge and introduce an approach for encoding protein structural knowledge into the learned representations.

Bepler & Berger, 2021, Cell Systems *12*, 654–669
June 16, 2021
https://doi.org/10.1016/j.cels.2021.05.017

CellPress

CellPress

Cell Systems

Synthesis

Learning the protein language: Evolution, structure, and function

Tristan Bepler[1,2,3,*] and Bonnie Berger[2,4,5,*]
[1]Simons Machine Learning Center, New York Structural Biology Center, New York, NY, USA
[2]Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA
[3]Computational and Systems Biology Program, Massachusetts Institute of Technology, Cambridge, MA, USA
[4]Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA, USA
[5]Lead contact
*Correspondence: tbepler@nysbc.org (T.B.), bab@mit.edu (B.B.)
https://doi.org/10.1016/j.cels.2021.05.017

SUMMARY

Language models have recently emerged as a powerful machine-learning approach for distilling information from massive protein sequence databases. From readily available sequence data alone, these models discover evolutionary, structural, and functional organization across protein space. Using language models, we can encode amino-acid sequences into distributed vector representations that capture their structural and functional properties, as well as evaluate the evolutionary fitness of sequence variants. We discuss recent advances in protein language modeling and their applications to downstream protein property prediction problems. We then consider how these models can be enriched with prior biological knowledge and introduce an approach for encoding protein structural knowledge into the learned representations. The knowledge distilled by these models allows us to improve downstream function prediction through transfer learning. Deep protein language models are revolutionizing protein biology. They suggest new ways to approach protein and therapeutic design. However, further developments are needed to encode strong biological priors into protein language models and to increase their accessibility to the broader community.

INTRODUCTION

Proteins are molecular machines that carry out the majority of the molecular function of cells. They are composed of linear sequences of amino acids which fold into complex ensembles of 3-dimensional structures, which can range from ordered to disordered and undergo conformational changes; biochemical and cellular functions emerge from protein sequence and structure. Understanding the sequence-structure-function relationship is the central problem of protein biology and is pivotal for understanding disease mechanisms and designing proteins and drugs for therapeutic and bioengineering applications.

The complexity of the sequence-structure-function relationship continues to challenge our computational modeling ability, in part because existing tools do not fully realize the potential of the increasing quantity of sequence, structure, and functional information stored in large databases. Until recently, computational methods for analyzing proteins have used either first principles-based structural simulations or statistical sequence modeling approaches that seek to identify sequence patterns that reflect evolutionary, and therefore functional, pressures (Marks, Hopf and Sander, 2012; Ekeberg et al., 2013; Wang et al., 2017; Liu et al., 2018; Yang et al., 2020) (Figure 1). Within these methods, structural analysis has been largely first principles driven while sequence analysis methods are primarily based on statistical sequence models, which make strong assumptions about evolutionary processes, but have become increasingly data driven with the growing amount of available natural sequence information.

Physics-based approaches use all atom energy functions (Hornak et al., 2006; Hess et al., 2008; Alford et al., 2017) or heuristics designed for proteins (Rohl et al., 2004) to estimate the energy of a given conformation and simulate natural motions. These methods are appealing, because they draw on our fundamental understanding of the physics of these systems and generate interpretable hypotheses. The Rosetta tool, which stitches together folded fragments associated with small constant-size contiguous subsequences, has been remarkably successful in its use of free energy estimation for protein folding and design (Leaver-Fay et al., 2011), and molecular dynamics software such as GROMACS are widely used for modeling dynamics and fine-grained structure prediction (Hess et al., 2008). Statistical sampling approaches have also been developed that seek to sample from accessible conformations based on coarse grained energy functions (Godzik, Kolinski and Skolnick, 1993; Srinivasan and Rose, 1995; Choi and Pappu, 2019). Rosetta has been especially successful for solving the design problem by using a mix of structural templates and free energy minimization to find sequences that match a target structure. However, despite Rosetta's successes, it and similar approaches assume simplified energy models, are extremely computationally expensive, require expert knowledge to set up correctly, and have limited accuracy.

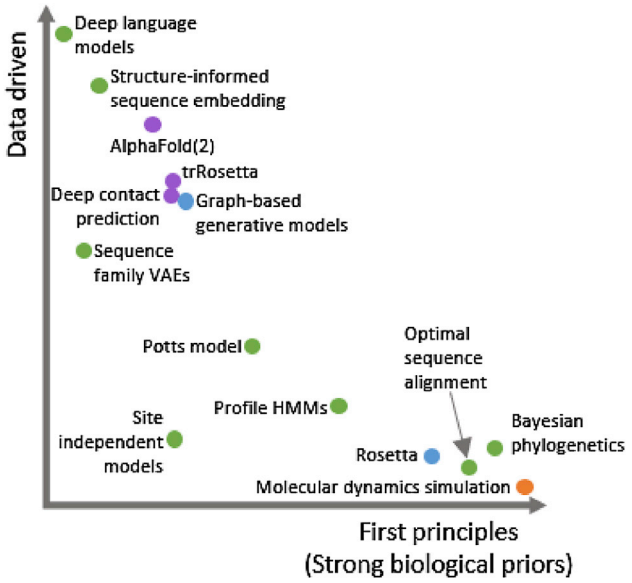


Figure 1. Two-dimensional schematic of some recent and classical methods in protein sequence and structure analysis, characterized by the extent to which the approach is motivated by first principles (strong biological priors) versus driven by big data

We color methods by types of input-output pairs. Green: sequence-sequence, purple: sequence-structure, blue: structure-sequence, orange: structure-structure. Classical methods tend to be more strongly first principles driven while newer methods are increasingly data driven. Existing methods tend to be either data driven or first principles-based with few methods existing in between. Note that, at this time, details of AlphaFold2 have not been made public, so placement in Figure 1 is a rough estimate. Some methods, especially Rosetta, can perform multiple functions.

At the other end of the spectrum, statistical sequence models have proven extremely useful for modeling the amino acid sequences of related sets of proteins. These methods allow us to discover constraints on amino acids imposed by evolutionary pressures and are widely used for homology search (Altschul and Koonin, 1998; Bateman et al., 2004; Rohl et al., 2004; Finn, Clements and Eddy, 2011; Remmert et al., 2011a) and for predicting residue-residue contacts in the 3D protein structure using covariation between amino acids at pairs of positions in the sequence (coevolution) (Göbel et al., 1994; Berger, 1995; Berger et al., 1995; Wolf, Kim and Berger, 1997; McDonnell et al., 2006; Trigg et al., 2011; Marks, Hopf and Sander, 2012; de Juan, Pazos and Valencia, 2013; Ekeberg et al., 2013). Advances in protein structure prediction have been driven by building increasingly large deep learning systems to predict residue-residue distances from sequence families (Liu et al., 2018; Xu and Wang, 2019) and fold proteins based on the predicted distance constraints which culminated recently in the success of AlphaFold2 at the Critical Assessment of protein Structure Prediction (CASP) 14 competition (Jumper et al., 2020). These methods rely on large datasets of protein sequences that are similar enough to be aligned with high confidence but contain enough divergence to confidently infer statistical couplings between positions. Accordingly, they are unable to learn patterns across large-scale databases of possibly unrelated proteins and have limited ability to draw on the increasing structure and function information available.

Language models have recently emerged as a powerful paradigm for generative modeling of sequences and as a means to learn "content-aware" data representations from large-scale sequence datasets. Statistical language models are probability distributions over sequences of tokens (e.g., words or characters in natural language processing, amino acids for proteins). Given a sequence of tokens, a language model assigns a probability to the whole sequence. In natural language processing (NLP), language models are widely used for machine translation, question-answering, and information retrieval among other applications. In biology, profile Hidden Markov Models (HMMs) are simple language models that are already widely used for homology modeling and search. Language models are able to capture complex dependencies between amino acids and can be trained on all protein sequences rather than being focused on individual families; in doing so, they have the potential to push the limits of statistical sequence modeling. In bringing these models to biology, we now not only have the ability to learn from naturally observed sequences, including across all of known sequence space (Alley et al., 2019; Bepler and Berger, 2019), but are also able to incorporate existing structural and functional knowledge through multi-task learning. (Box 1 provides a glossary of terms that might be less familiar.) Language models learn the probability of a sequence occurring and this can be directly applied to predict the fitness of sequence mutations (Riesselman, Ingraham and Marks, 2018; Hie et al., 2020a, 2021). They also learn summary representations, powerful features that can be used to better capture sequence relationships and link sequence to function via transfer learning (Alley et al., 2019; Bepler and Berger, 2019; Rao et al., 2019; Rives et al., 2019; Hie et al., 2020b; Luo et al., 2020). Finally, language models also offer the potential for controlled sequence generation by conditioning the language model on structural (Ingraham et al., 2019a) or functional (Madani et al., 2020) specifications.

Deep language models are an exciting breakthrough in protein sequence modeling, allowing us to discover aspects of structure and function from only the evolutionary relationships present in a corpus of sequences. However, the full potential of these models has not been realized as they continue to benefit from more parameters, more compute power, and more data. At the same time, these models can be enriched with strong biological priors through multi-task learning.

Here, we propose that methods incorporating both large datasets and strong domain knowledge will be key to unlocking the full potential of protein sequence modeling. Specifically, physical structure-based priors can be learned through structure supervision while also learning evolutionary relationships from hundreds of millions of natural protein sequences. Furthermore, the evolutionary and structural relationships encoded allow us to learn functional properties of proteins through transfer learning. In this synergy, we will discuss these developments and present new results toward enriching large-scale language models with structure-based priors through multi-task learning. First, we will discuss new developments in deep learning and language modeling and their application to protein sequence modeling with large datasets. Second, we will discuss how we can enrich these models with structure supervision. Third, we will discuss transfer learning and demonstrate that the evolutionary and structural information encoded in our deep language models

Box 1. Glossary

1-hot [embedding]. Vector representation of a discrete variable commonly used for discrete values that have no meaningful ordering. Each token is transformed into a V-dimensional zero vector, where V is the size of the vocabulary (the number of unique tokens, e.g., 20, 21, or 26 for amino acids depending on inclusion of missing and non-canonical amino acid tokens), except for the index representing the token, which is set to one.

autoregressive [language model]. Language models that factorize the probability of a sequence into a product of conditional probabilities in which the probability of each token is conditioned on the preceding tokens, $p(x_1 \ldots x_L) = \prod_{i=1}^{L} p(x_i | x_1 \ldots x_{i-1})$. Examples of autoregressive language models include k-mer (AKA n-gram) models, Hidden Markov Models, and typical autoregressive recurrent neural network or generative transformer language models. These models are called *autoregressive* because they model the probability of one token after another in order.

Bayesian methods. A statistical inference approach that uses Bayes rule to infer a posterior distribution over model parameters given by the observed data. Because these methods describe distributions over parameters or functions, they are especially useful in small data regimes or other settings when prediction uncertainties are desirable.

cloze task. A task in natural language processing, also known as the cloze test. The task is to fill in missing words given the context. For example, "The quick brown ____ jumps over the lazy dog."

conditional random field. Models the probability of a set (sequence in this case, i.e. linear chain CRF) of labels given a set of input variables by factorizing it into locally conditioned potentials conditioned on the input variables, $p(y_1 \ldots y_L \mid x_1 \ldots x_L) = p(y_1 \mid x_1 \ldots x_L) \prod_{i=2}^{L} p(y_i \mid y_{i-1}, x_1 \ldots x_L)$. This is often simplified such that each conditional only depends on the local input variable, i.e., $p(y_1 \ldots y_L \mid x_1 \ldots x_L) = p(y_1 \mid x_1) \prod_{i=2}^{L} p(y_i \mid y_{i-1}, x_i)$. Linear chain CRFs can be seen as the discriminative version of Hidden Markov Models.

contextual vector embedding. Vector embeddings that include information about the sequence context in which a token occurs. Encoding context into vector embeddings is important in NLP, because words can have different meanings in different contexts (i.e. many homonyms exist). For example, in the sentences, "she tied the ribbon into a bow" and "she drew back the string on her bow," the word bow refers to two different objects that can only be inferred from context. In the case of proteins, this problem is even worse, because there are only 20 (canonical) amino acids and so their "meaning" is highly context dependent. This is in contrast to typical vector embedding methods that learn a single vector embedding per token regardless of context.

distributional hypothesis. The observation that words that occur in similar contexts tend to have similar meanings. Applies also to proteins due to evolutionary pressure (Harris, 1954).

Gaussian process. A class of models that describes distributions over functions conditioned on observations from those functions. Gaussian processes model outputs as being jointly normally distributed where the covariance between the outputs is a function of the input features. See Rasmussen and Williams for a comprehensive overview (Rasmussen and Williams, 2005)

generative model. A model of the data distribution, $p(X)$, joint data distribution, $p(X,Y)$, or conditional data distribution, $p(X|Y = y)$. Usually framed in contrast to discriminative models that model the probability of the target given an observation, $p(Y|X = x)$. Here, $X$is observable, for example the protein sequence, and $Y$is a target that is not observed, for example the protein structure or function. Conditional generative and discriminative models are related by Bayes' theorem. Language models are generative models.

hidden layer. Intermediate vector representations in a deep neural network. Deep neural networks are structured as layered data transformations before outputting a final prediction. The intermediate layers are referred to as "hidden" layers.

inductive bias. Describes the assumptions that a model uses to make predictions for data points it has not seen (Mitchell, 1980). That is, the inductive bias of a model is how that model generalizes to new data. Every machine learning model has inductive biases, implicitly or explicitly. For example, protein phenotype prediction based on homology assumes that phenotypes covary over evolutionary relatedness. In other words, it formally models the idea that proteins that are more evolutionarily related are likely to share the same function. In thinking about deep neural networks applied to proteins, it is important to understand the inductive biases these models assume, because it naturally relates to the true properties of the function we are trying to model. However, this is challenging, because we can only roughly describe the inductive biases of these models (Battaglia, Hamrick and Bapst, 2018).

language model. Probabilistic model of whole sequences. In the case of natural language, language models typically describe the probability of sentences or documents. In the case of proteins, they model the probability of amino acid sequences. Being simply probabilistic models, language models can take on many specific incarnations from column frequencies in multiple sequence alignments to Hidden Markov Models to Potts models (direct coupling analysis) to deep neural networks.

manifold embedding. A distance preserving, low dimensional embedding of the data. The goal of manifold embedding is to find points low dimensional vectors, $z_1 \ldots z_n$, such that the distances, $d(z_i, z_j)$, are as close as possible to the distances in the original data space, $d(x_i, x_j)$, given n high dimensional data vectors, $x_1 \ldots x_n$. t-SNE is a commonly used manifold embedding approach for visualization of high dimensional data.

masked language model. The training task used by BERT and other recent bidirectional language models. Instead of modeling the probability of a sequence autoregressively, masked language models seek to model the probability of each token given all other tokens. For computational convenience, this is achieved by randomly masking some percentage of the tokens in each

(Continued on next page)

Box 1. *Continued*

minibatch and training the model to recover those tokens. An auxiliary token is added to the vocabulary to indicate that this token has been masked.

multi-task learning. A machine learning paradigm in which multiple tasks are learned simultaneously. The idea is that similarities between tasks can lead to each task being learned better in combination rather than learning each individually. In the case of representation learning, multi-task learning can also be useful for learning representations that encode information relevant for all tasks. Multi-task learning allows us to use the signals encoded in other training signals as an inductive bias when learning the goal task.

representation learning. The problem of learning features, or intermediate data representations, better suited for solving a prediction problem on raw data. Deep learning systems are described as representation learning systems, because they learn a series of data transformations that make the goal task progressively easier to solve before outputting a prediction.

residue-residue contact prediction. The task of learning which amino acid residues are in contact in folded protein structures, where contact is assumed to be within a small number of angstroms, often with the goal of constraining the search space for protein structure prediction.

self-supervised learning. A relatively new term for methods for learning from data without labels. Generally used to describe methods that "automatically" create labels through data augmentation or generative modeling. Can be viewed as a subset of unsupervised learning focused on learning representations useful for transfer learning.

semantic priors. Prior semantic understanding of a word or token, e.g., protein structure or function.

semantics. The meaning of a word or token. In reference to proteins, we use semantics to mean the "functional" purpose of a residue, or combinations of residues.

structural classification of proteins (SCOP). A mostly manual curation of structural domains based on similarities of their sequences and structure. Similar databases include CATH (Sillitoe *et al.*, 2021).

structural similarity prediction. Given two protein sequences, predict how similar their respective structures would be according to some similarity measure.

supervised learning. A problem in machine learning. How we can learn a function to predict a target variable, usually denoted y, given an observed one, usually denoted x, from a set of known x, y pairs.

transfer learning. A problem in machine learning. How we can take knowledge learned from one task and apply it to solve another related task. When the tasks are different but related, representations learned on one task can be applied to the other. For example, representations learned from recognizing dogs could be transferred to recognizing cats. In the case of proteins and language models, we are interested in applying knowledge gained from learning to generate sequences to predicting function. Transfer learning could also be applied to applying representations learned from predicting structure to function or from predicting one function to another function among other applications.

unsupervised learning. A problem in machine learning that asks how we can learn patterns from unlabeled data. Clustering is a classic unsupervised learning problem. Unsupervised learning is often formulated as a generative modeling problem, where we view the data as being generated from some unobserved latent variable(s) that we infer jointly with the parameters of the model.

vector embedding. A term used to describe multidimensional real numbered representations of data that is usually discrete or high dimensional, word embeddings being a classic example. Sometimes referred to as "distributed vector embeddings" or "manifold embeddings" or simply just "embeddings." Low-dimensional vector representations of high dimensional data such as images or gene expression vectors as found by methods such as t-SNE are also vector embeddings. Usually, the goal in learning vector embeddings is to capture some semantic similarity between data as a function of similarity or distance in the vector embedding space.

can be used to improve protein function prediction. Finally, we will discuss future directions in protein machine learning and large-scale language modeling.

Protein language models distill information from massive protein sequence databases

Language models for protein sequence representation learning (Figure 2) have seen a surge of interest following the success of large-scale models in the field of natural language processing (NLP). These models draw on the idea that distributed vector representations of proteins can be extracted from generative models of protein sequences, learned from a large and diverse database of sequences across natural protein space, and thus can capture the semantics, or function, of a given sequence. Here, function refers to any and all properties related to what a protein does. These properties are often subject to evolutionary pressures because these functions must be maintained or enhanced in order for an organism to survive and reproduce. These pressures manifest in the distribution over amino acids present in natural protein sequences and, hence, are discoverable from large and diverse enough sets of naturally occurring sequences.

The ability to learn semantics emerges from the distributional hypothesis: tokens (e.g., words, amino acids) that occur in similar contexts tend to carry similar meanings. Language models only require sequences to be observed and are trained to model the probability distribution over amino acids using an *autoregressive* formulation (Figures 2A and 2B) or masked position prediction formulation (also called a cloze task in NLP, Figure 2C). In autoregressive language models, the probability of a sequence is factorized such that the probability of each token

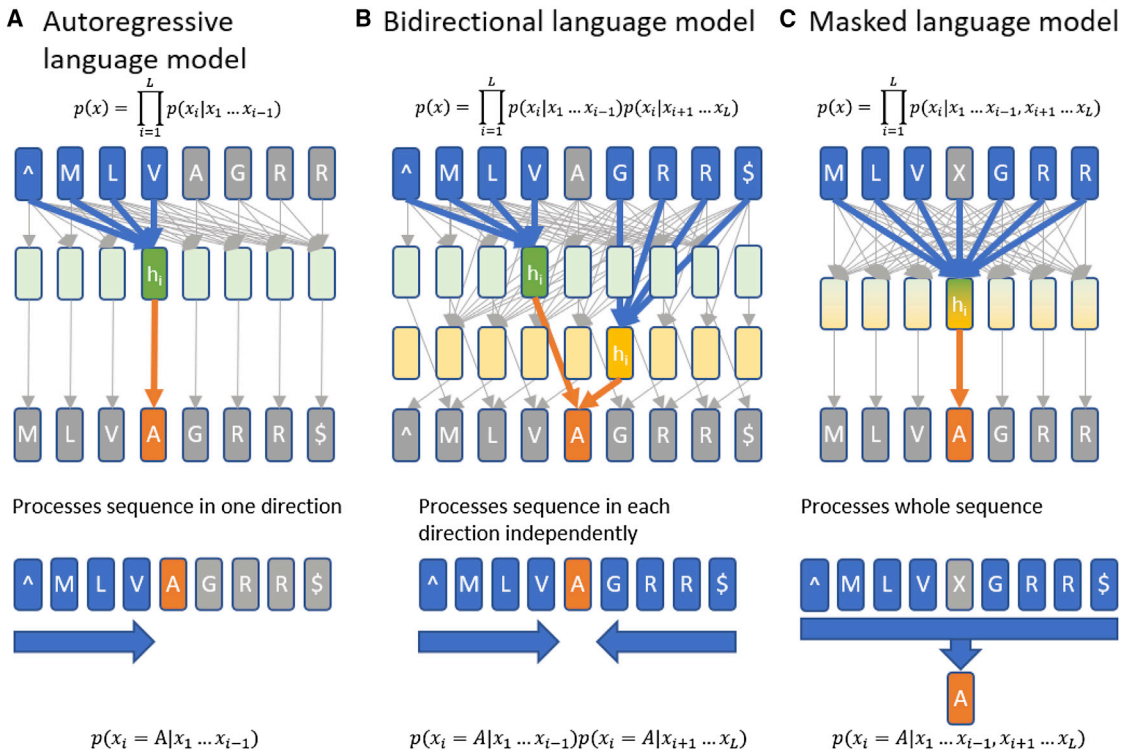


Figure 2. Diagram of model architectures and language modeling approaches

(A) Language models model the probability of sequences. Typically, this distribution is factorized over the sequence such that the probability of a token (e.g., amino acid) at position *i* (x_i) is conditioned on the previous tokens. In neural language models, this is achieved by first computing a hidden layer (h_i) given by the sequence up to position *i-1* and then calculating the probability distribution over token x_i given h_i. In this example sequence, "^" and "$" represent start and stop tokens respectively and the sequence has length *L*.

(B) Bidirectional language models instead model the probability of a token conditioned on the previous and following tokens independently. For each token x_i, we compute a hidden layer using separate forward and reverse direction models. These hidden layers are then used to calculate the probability distribution over tokens at position *i* conditioned on all other tokens in the sequence. This allows us to extract representations that capture complete sequence context.

(C) Masked language models model the probability of tokens at each position conditioned on all other tokens in the sequence by replacing the token at each position with an extra "mask" token ("X"). In these models, the hidden layer at each position is calculated from all tokens in the sequence which allows the model to capture conditional non-independence between tokens on either side of the masked token. This formulation lends itself well to transfer learning, because the representations can depend on the full context of each token.

is conditioned only on the preceding tokens. This factorization is exact and is useful when sampling from the distribution or evaluating the probabilities themselves is of primary interest. The drawback to this formulation is that the representations learned for each position depend only on preceding positions, potentially making them less useful as contextual representations. The masked position prediction formulation (also known as masked language modeling) addresses this problem by considering the probability distribution over each token at each position conditioned on all other tokens in the sequence. The masked language modeling approach does not allow calculating correctly normalized probabilities of whole sequences but is more appropriate when the learned representations are the outcomes of primary interest. The unprecedented recent success of language models in natural language processing, e.g.Google's BERT and OpenAI's GTP-3, is largely driven by their ability to learn from billions of text entries in enormous online corpora. Analogously, we have natural protein sequence databases with 100 s of millions of unique sequences that continue to grow rapidly.

Recent advances in NLP have been driven by innovations in neural network architectures, new training approaches, increasing compute power, and increasing accessibility of huge text corpuses. Several NLP methods have been proposed that draw on unsupervised, now often called self-supervised, learning (Devlin et al., 2018; Peters et al., 2018) to fit large-scale bidirectional long-short term recurrent neural networks (bidirectional LSTMs or biLSTMs) (Hochreiter and Schmidhuber, 1997; Graves, Fernández and Schmidhuber, 2005) or Transformers (Vaswani et al., 2017) and its recent variants. LSTMs are recurrent neural networks. These models process sequences one token at a time in order and therefore learn representations that capture information from a position and all previous positions. In order to include information from tokens before and after any given position, bidirectional LSTMs combine two separate LSTMs operating in the forward and backward directions in each layer (e.g., as in Figure 2B). Although these models can learn representations including whole sequence context, their ability to learn distant dependencies is limited in practice. To address this limitation, transformers learn representations by

explicitly calculating an attention vector over each position in the sequence. In the self-attention mechanism, the representation for each position is learned by "attending to" each position of the same sequence, well suited for masked language modeling (Figure 2C). In a self-attention module, the output representation of each element of a sequence is calculated as a weighted sum over transformations of the input representations at each position where the weighting itself is based on a learned transformation of the inputs. The attention mechanism is typically believed to allow transformers to learn dependencies between positions distant in the linear sequence more easily. Transformers are also useful as autoregressive language models.

In natural language processing, Peters et al. recognized that the hidden layers (intermediate representations of stack neural networks) of biLSTMs encoded semantic meaning of words in context. This observation has been newly leveraged for biological sequence analysis (Alley et al., 2019; Bepler and Berger, 2019) to learn more semantically meaningful sequence representations. The success of deep transformers for machine translation inspired their application to contextual text embedding, that is learning contextual vector embeddings of words and sentences, giving rise to the now widely used Bidirectional Encoder Representations from Transformers (BERT) model in NLP (Devlin et al., 2018). BERT is a deep transformer trained as a masked language model on a large text corpus. As a result, it learns contextual representations of text that capture contextual meaning and improve the accuracy of downstream NLP systems. Transformers have also demonstrated impressive performance as autoregressive language models, for example with the Generative Pre-trained Transformer (GPT) family of models (Radford et al., 2018, 2019; Brown et al., 2020), which have made impressive strides in natural language generation. These works have inspired subsequent applications to protein sequences (Rao et al., 2019; Rives et al., 2019; Elnaggar et al., 2020; Vig et al., 2020).

Although transformers are powerful models, they require enormous numbers of parameters and train more slowly than typical recurrent neural networks. With massive scale datasets and compute and time budgets, transformers can achieve impressive results, but, generally, recurrent neural networks (e.g., biLSTMs) need less training data and less compute, so might be more suitable for problems where fewer sequences are available, such as training on individual protein families, or compute budgets are tight. Constructing language models that achieve high accuracy with better compute efficiency is an algorithmic challenge for the field. An advantage of general purpose pretrained protein models is that we only need to do the expensive training step once; the models can then be used to make predictions or can be applied to new problems via transfer learning (Bengio, 2012), as discussed below.

Using these and other tools, protein language models are able to synthesize the enormous quantity of known protein sequences by training on 100 s of millions of sequences stored in protein databases (e.g., UniProt, Pfam, NCBI (Bateman et al., 2004; Pruitt, Tatusova and Maglott, 2007; UniProt Consortium, 2019)). The distribution over sequences learned by language models captures the evolutionary fitness landscape of known proteins. When trained on tens of thousands of evolutionarily related proteins, the learned probability mass function describing the empirical distribution over naturally occurring sequences has shown promise for predicting the fitness of sequence variants (Riesselman, Ingraham and Marks, 2018; Hie et al., 2020a, 2021). Because these models learn from evolutionary data directly, they can make accurate predictions about protein function when function is reflected in the fitness of natural sequences. Riesselman et al. first demonstrated that language models fit on individual protein families are surprisingly accurate predictors of variant fitness measured in deep mutational scanning datasets (Riesselman, Ingraham and Marks, 2018). New work has since shown that the representations learned by language models are also powerful features for learning of variant fitness as a subsequent supervised learning task (Rives et al., 2019; Luo et al., 2020), building on earlier observations that language models can improve protein property prediction through transfer learning (Bepler and Berger, 2019). Recently, Hie et al. used language models to learn evolutionary fitness of viral envelope proteins and were able to predict mutations that could allow the SARS-CoV-2 spike protein to escape neutralizing antibodies (Hie et al., 2020a, 2021). As of publication, several variants predicted to have high escape potential have appeared in SARS-CoV-2 sequencing efforts around the world, but viral escape has not yet been experimentally verified (Walensky et al., 2021).

A few recent works have focused on increasing the scale of these models by adding more parameters and more learnable layers to improve sequence modeling. Interestingly, because so many sequences are available, these models continue to benefit from increased size (Rives et al., 2019). This parallels the general trend in natural language processing, where the number of parameters, rather than specific architectural choices, is the best indicator of model performance (Kaplan et al., 2020). However, ultimately, model size is limited by the computational resources available to train and apply these models. In NLP, models such as BERT and GPT-3 have become so large that only the best funded organizations with massive Graphics Processing Unit (GPU) compute clusters are realistically able to train and deploy them. This is demonstrated in some recent work on protein models where single transformer-based models were trained for days to weeks on hundreds of GPUs (Rives et al., 2019; Elnaggar et al., 2020; Vig et al., 2020), costing potentially 100 s of thousands of dollars for training. Increasing the scale of these models promises to continue to improve our ability to model proteins, but more resource efficient algorithms are needed to make these models more accessible to the broader scientific community.

So far, the language models we have discussed use natural protein sequence information. However, they do not learn from the protein structure and function knowledge that has been accumulated over the past decades of protein research. Incorporating such knowledge requires supervised approaches.

Supervision encodes biological meaning

Proteins are more than sequences of characters: they are physical chains of amino acids that fold into three-dimensional structures and carry out functions based on those structures. The sequence-structure-function relationship is the central pillar of protein biology and significant time and effort has been spent to elucidate this relationship for select proteins of interest. In particular, the increasing throughput and ease-of-use of protein

structure determination methods, (e.g., X-ray crystallography and cryo-EM (Cheng et al., 2015; Callaway, 2020)), has driven a rapid increase in the number of known protein structures available in databases such as the Protein Data Bank (PDB) (Berman et al., 2000). There are nearly 175,000 entries in PDB as of publication and this number is growing rapidly. 14,000 new structures were deposited in 2020 and the rate of new structure deposition is increasing. We pursue the intuition that incorporating such knowledge into our models via supervised learning can aid in predicting function from sequence, bypassing the need for solved structures.

Supervised learning is the problem of finding a mathematical function to predict a target variable given some observed variables. In the case of proteins, supervised learning is commonly used to predict protein structure from sequence, protein function from sequence, or for other sequence annotation problems (e.g., signal peptide or transmembrane region annotation). Beyond making predictions, supervised learning can be used to encode specific semantics into learned representations. This is common in computer vision where, for example, pre-training image recognition models on the large ImageNet dataset is used to prime the model with information from natural image categories (Russakovsky et al., 2015).

When we use supervised approaches, we encode semantic priors into our models. These priors are important for learning relationships that are not obvious from the raw data. For example, unrelated protein sequences can form the same structural fold and, therefore, are semantically similar. However, we cannot deduce this relationship from sequences alone. Supervision is required to learn that these sequences belong to the same semantic category. Although structure is more informative of function than sequence (Zhang and Kim, 2003; Shin et al., 2007) and structure is encoded by sequence, predicting structure remains hard, particularly due to the relative paucity of structural relative to sequence data. Significant strides have been made recently with massive computing resources (Jumper et al., 2020); yet there is still a long way to go before a complete sequence to structure mapping is possible. The degree to which such a map could or should be possible, even in principle, is unclear.

Evolutionary relationships between sequences are informative of structural and functional relationships, but only when the degree of sequence homology is sufficiently high. Above 30% sequence identity, structure and function are usually conserved between natural proteins (Rost, 1999). Often called the ''twilight zone'' of protein sequence homology, proteins with similar structures and functions still exist below this level, but they can no longer be detected from sequence similarity alone and it is unclear whether their functions are conserved. Although it is generally believed that proteins with similar sequences form similar structures, there are also interesting examples of highly similar protein sequences having radically different structures and functions (Kosloff and Kolodny, 2008; Wei et al., 2020) and of sequences that can form multiple folds (James and Tawfik, 2003). Evolutionary innovation requires that protein function can change with only a few mutations. Furthemore, it is important to note that although structure and function are related, they should not be directly conflated.

These phenomena suggest that there are aspects of protein biology that may not be discoverable by statistical sequence models alone. Supervision that represents known protein structure, function, and other prior knowledge may be necessary to encode distant sequence relationships into learned embeddings. By analogy, cars and boats are both means of transportation, but we would not expect a generative image model to infer this relationship from still images alone. However, we can teach these relationships through supervision.

On this premise, we hypothesize that incorporating structural supervision when training protein language models will improve the ability to predict function in downstream tasks through transfer learning. Eventually, such language models may become powerful enough that we can predict function directly without the need for solved structures. In the remainder of this Synthesis, we will explore this idea.

Multi-task language models capture the semantic organization of proteins

Here, we will demonstrate that training protein language models with self-supervision on a large amount of natural sequence data *and* with *structure* supervision on a smaller set of sequence, structure pairs enriches the learned representations and translates into improvements in downstream prediction problems (Figure 3). First, we generate a dataset that contains 76 million protein sequences from Uniref (Suzek et al., 2007) and an additional 28,000 protein sequences with structures from the Structural Classification of Proteins (SCOP) database, which classifies protein sequences into a hierarchy of structural motifs based on their sequence and structural similarities (e.g., family, superfamily, class) (Fox, Brenner and Chandonia, 2014; Chandonia, Fox and Brenner, 2017). Next, we train a bidirectional LSTM with three learning tasks simultaneously: 1) the masked language modeling task (Figures 2C and 3A), 2) residue-residue contact prediction (Figure 3B), and 3) structural similarity prediction (Figure 3C).

The fundamental idea behind this novel training scheme is to combine self-supervised and supervised learning approaches to overcome the shortcomings of each. Specifically, the masked language modeling objective (self-supervision) allows us to learn from millions of natural protein sequences from the Uniprot database. However, this does not include any prior semantic knowledge from protein structure and, therefore, has difficulty learning semantic similarity between divergent sequences. To address this, we consider two structural supervision tasks, residue-residue contact prediction and structural similarity prediction, trained with tens of thousands of protein structures classified by SCOP. In the residue-residue contact prediction task, we use the hidden layers of the language model to predict contacts between residues within the 3D structure using a learned bilinear projection layer (Figure 3B). In the structural similarity prediction task, we use the hidden layers of the language model to predict the number of shared structural levels in the SCOP hierarchy by aligning the proteins in vector embedding space and using this alignment score to predict structural similarity from the sequence embeddings. This task is critical for encoding structural relationships between unrelated sequences into the model. The parameters of the language model are shared across the self-supervised and two supervised tasks and the entire model is trained end-to-end. The set of proteins with known structure is much smaller than the full set of known proteins in Uniprot

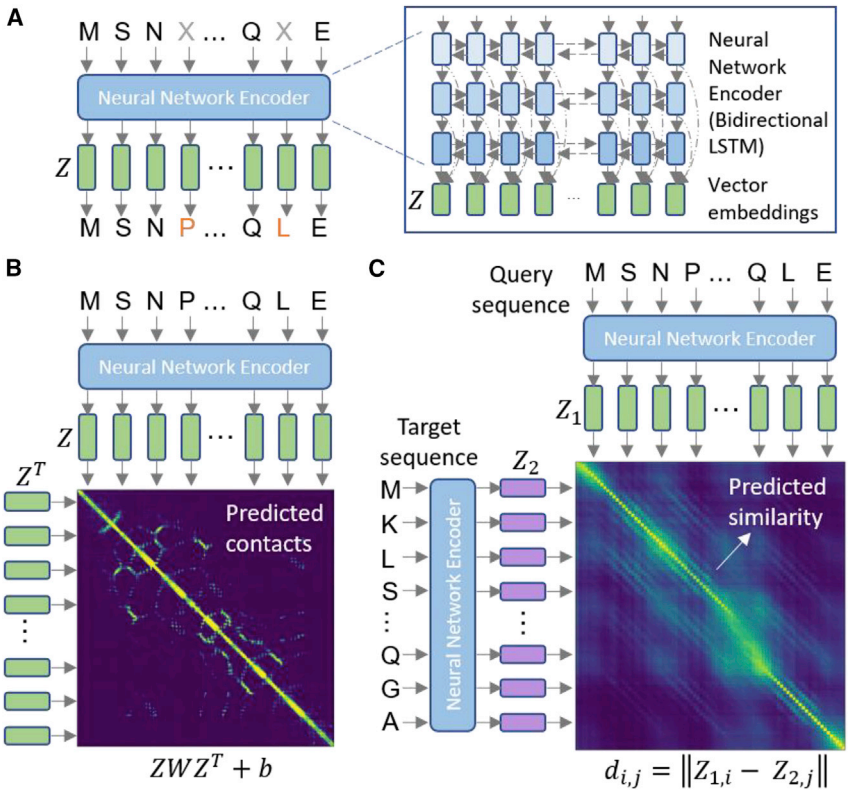


Figure 3. Our *multi-task* contextual embedding model learning framework

We train a neural network (NN) sequence encoder to solve three tasks simultaneously. The first task is masked language modeling on millions of natural protein sequences. We include two sources of structural supervision in a multi-task framework (MT-LSTM for Multi-Task LSTM) in order to encode structural semantics directly into the representations learned by our language model. We combine this with the masked language model objective to benefit from evolutionary and less available structure information (only 10 s of thousands of proteins).

(A) The masked language model objective allows us to learn contextual embeddings from hundreds of millions of sequences. Our training framework is agnostic to the NN architecture, but we specifically use a three layer bidirectional LSTM with skip connections (inset box) in this work in order to capture long range dependencies but train quickly. We can train language models using only this objective (DLM-LSTM), but can also enrich the model with structural supervision.

(B) The first structure task is predicting contacts between residues in protein structures using a bilinear projection of the learned embeddings. In this task, The hidden layer representations of the language model are then used to predict residue-residue contacts using a bilinear projection. That is, we model the log likelihood ratio of a contact between the *i*-th and *j*-th residues in the protein sequence, by $z_i W z_j + b$where matrix W and scalar b are learned parameters.

(C) The second source of structural supervision is structural similarity, defined by the Structural Classification of Proteins (SCOP) hierarchy (Hubbard et al., 1997). We predict the ordinal levels of similarity between pairs of proteins by aligning the sequences in embedding space. Here, we embed the query and target sequences using the language model (Z_1 and Z_2) and then predict the structural homology by calculating the pairwise distances between the query and target embeddings ($d_{i,j}$) and aligning the sequences based on these distances.

and, therefore, by combining these tasks in a multi-task learning approach we can learn language models and sequence representations that are enriched with strong biological priors from known protein structures. We refer to this model as the multi-task (MT)-LSTM.

Next, we demonstrate how the trained language model can be used for protein sequence analysis and compare this with conventional approaches. Given the trained MT-LSTM, we apply it to new protein sequences to embed them into the learned semantic representation space (Figure 4A). Sequences are fed through the model and the hidden layer vectors are combined to form vector embeddings of each position of the sequences. Given a sequence of length L, this yields L D-dimensional vectors, where D is the dimension of the vector embeddings. This allows us to map the semantic space of each residue within a sequence, but we can also map the semantic space of whole sequences by summarizing them into fixed size vector embeddings via a reduction operation. Practically, this is useful for coarse sequence comparisons including clustering and manifold embedding for visualization of large protein datasets, revealing evolutionary, structural, and functional relationships between sequences in the dataset (Figure 4B). In this figure, we visualize proteins in the SCOP dataset, colored by structural class, after embedding with our MT-LSTM. For comparison, we also show results of embedding using a bidirectional LSTM trained only with the masked language modeling objective (DLM-LSTM), which is not enriched with the structure-based priors. We observe that even though the DLM-LSTM model was trained using only sequence information, protein sequences still organize roughly by structure in embedding space. However, this organization is improved when we include structure supervision in the language model training (Figure 4B).

The semantic organization of our learned embedding space enables a direct application: we can search protein sequence databases for semantically related proteins by comparing proteins based on their vector embeddings (Bepler and Berger, 2019). Because we embed sequences into a semantic representation space, we can find structurally related proteins even though their sequences are not closely related (Figure 4C, Table S1). To demonstrate this, we take pairs of proteins in the SCOP database, not seen by our multi-task model during training, and calculate the similarity between these pairs of sequences using direct sequence homology-based methods (Needleman-Wunsch alignment, HMM-sequence alignment, and HMM-HMM alignment (Needleman and Wunsch, 1970; Eddy, 2011; Remmert et al., 2011b)), a popular structure-based method (TMalign (Zhang and Skolnick, 2005)), and an alignment between the sequences in our learned embedding space. We then evaluate these methods based on their ability to correctly find pairs of proteins that are similar at the class, fold, superfamily, and

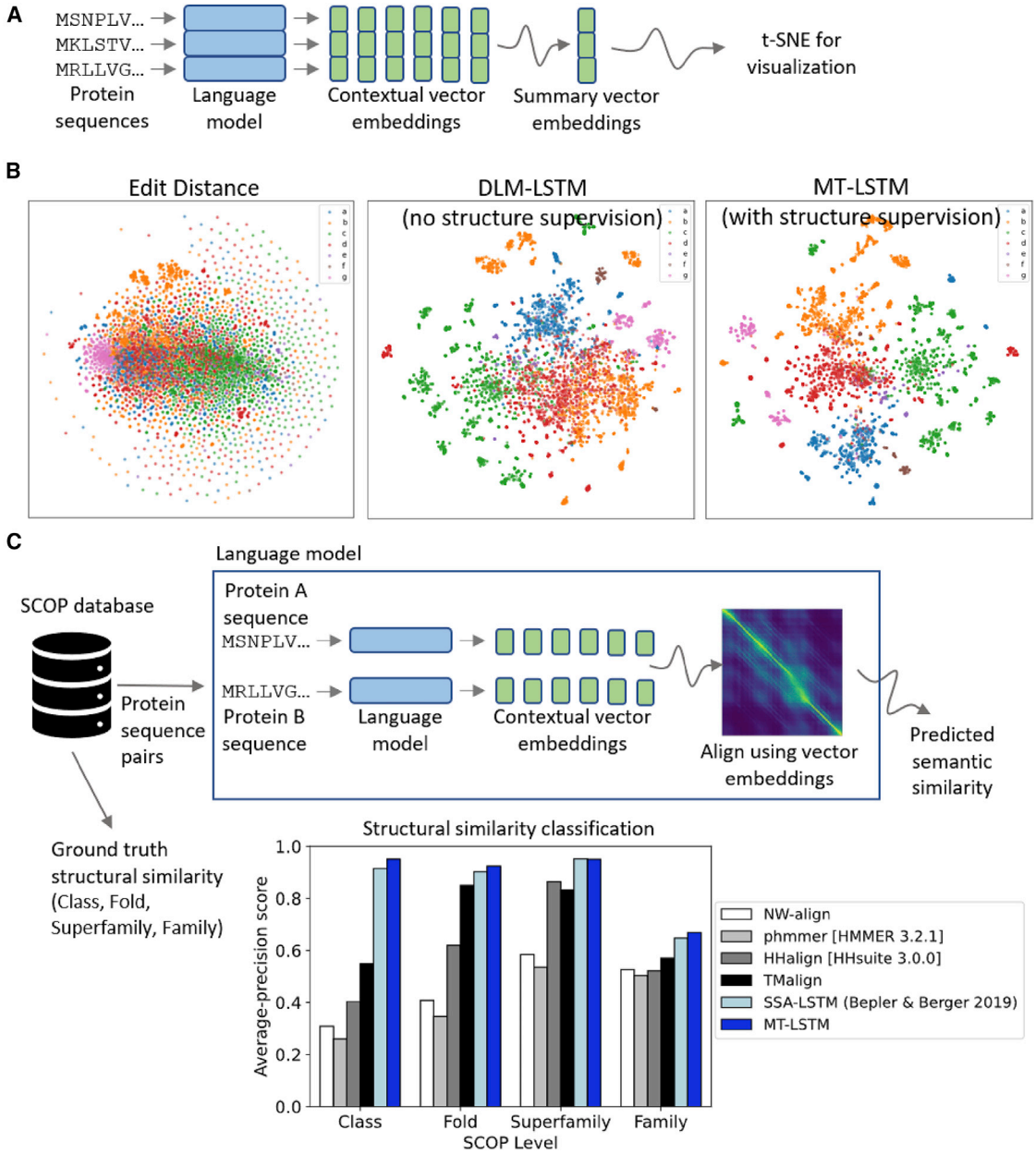


Figure 4. Language models capture the semantic organization of proteins

(A) Given a trained language model, we embed sequences by processing them with the neural network and taking the hidden layer representations for each position of the sequence. This gives an LxD matrix containing a D-dimensional vector embedding for each position of a length L sequence. We can reduce this to a D-dimensional vector "summarizing" the entire sequence by a pooling operation. Specifically, we use averaging here. These representations allow us to directly visualize large protein datasets with manifold embedding techniques.

(B) Manifold embedding of SCOP protein sequences reveals that our language models learn protein sequence representations that capture structural semantics of proteins. We embed thousands of protein sequences from the SCOP database and show t-SNE plots of the embedded proteins colored by SCOP structural class. The masked language (unsupervised) model (DLM-LSTM) learns embeddings that separate protein sequences by structural class, whereas the multi-task language model (MT-LSTM) with structural supervision learns an even better organized embedding space. In contrast, manifold embedding of sequences directly (edit distance) produces an unintelligible mash and does not resolve structural groupings of proteins.

(C) In order to quantitatively evaluate the quality of the learned semantic embeddings, we calculate the correspondence between semantic similarity predicted by our language model representations and ground truth structural similarities between proteins in the SCOP database. Given two proteins, we calculate the semantic similarity between them by embedding these proteins using our MT-LSTM, align the proteins using the embeddings, and calculate an alignment score. We compute the average-precision score for retrieving pairs of proteins similar at different structural levels in the SCOP hierarchy based on this predicted semantic similarity and find that our semantic similarity score dramatically outperforms other direct sequence comparison methods for predicting protein similarity.

(legend continued on next page)

family level, based on their SCOP classification. We find that our learned semantic embeddings dramatically outperform the sequence comparison methods and even outperform structure comparison with TMalign when predicting structural similarity. Interestingly, we observe that the structural supervision component is critical for learning well organized embeddings at a fine-grained level, because the DLM-LSTM representations alone do not perform well at this task (Table S1). Furthermore, the multi-task learning approach outperforms a two-step learning approach presented previously (SSA-LSTM) (Bepler and Berger, 2019).

With the success of our self-supervised and supervised language models, we sought to investigate whether protein language models could improve function prediction through transfer learning.

Transfer learning improves downstream applications

A key challenge in biology is that many problems are small data problems. Quantitative protein characterization assays are rarely high throughput and methods are needed that can generalize given only 10 s to 100 s of experimental measurements. Furthermore, we are often interested in extrapolating from data collected over a small region of protein sequence space to other sequences, often with little to no homology. Learned protein representations improve predictive ability for downstream prediction problems through transfer learning (Figure 5A). Transfer learning is the problem of applying knowledge learned from solving some prior tasks to a different task of interest. In other words, learning to solve task A can help learn to solve task B; analogously, learning how to wax cars helps to learn karate moves (Karate Kid, 1984). This is especially useful for tasks with little available training data, such as protein function prediction, because models can be pre-trained on other tasks with plentiful training data to improve performance through transfer learning.

Application of protein language models to downstream tasks through transfer learning was first demonstrated by Bepler and Berger (2019). They showed that transfer learning was useful for structural similarity prediction, secondary structure prediction, residue-residue contact prediction, and transmembrane region prediction, by fitting task specific models on top of a pre-trained bidirectional language model. The key insight was that the sequence representations (vector embeddings) learned by the language model were powerful features for solving other prediction problems. Since then, various language model-based protein embedding methods have been applied to these and other protein prediction problems through transfer learning, including protein phenotype prediction (Alley et al., 2019; Rao et al., 2019; Rives et al., 2019; Luo et al., 2020), residue-residue contact prediction (Rives et al., 2019; Rao et al., 2020), fold recognition (Rao et al., 2019), protein-protein (Zhou et al., 2020; Sledzieski et al., 2021) and protein-drug interaction prediction (Hie et al., 2020b; Truong and Truong, 2020). Recent works have shown that increasing language model scale leads to continued improvements in downstream applications, such as residue-residue contact prediction (Rao et al., 2020). We also find that increasing model size improves transfer learning performance.

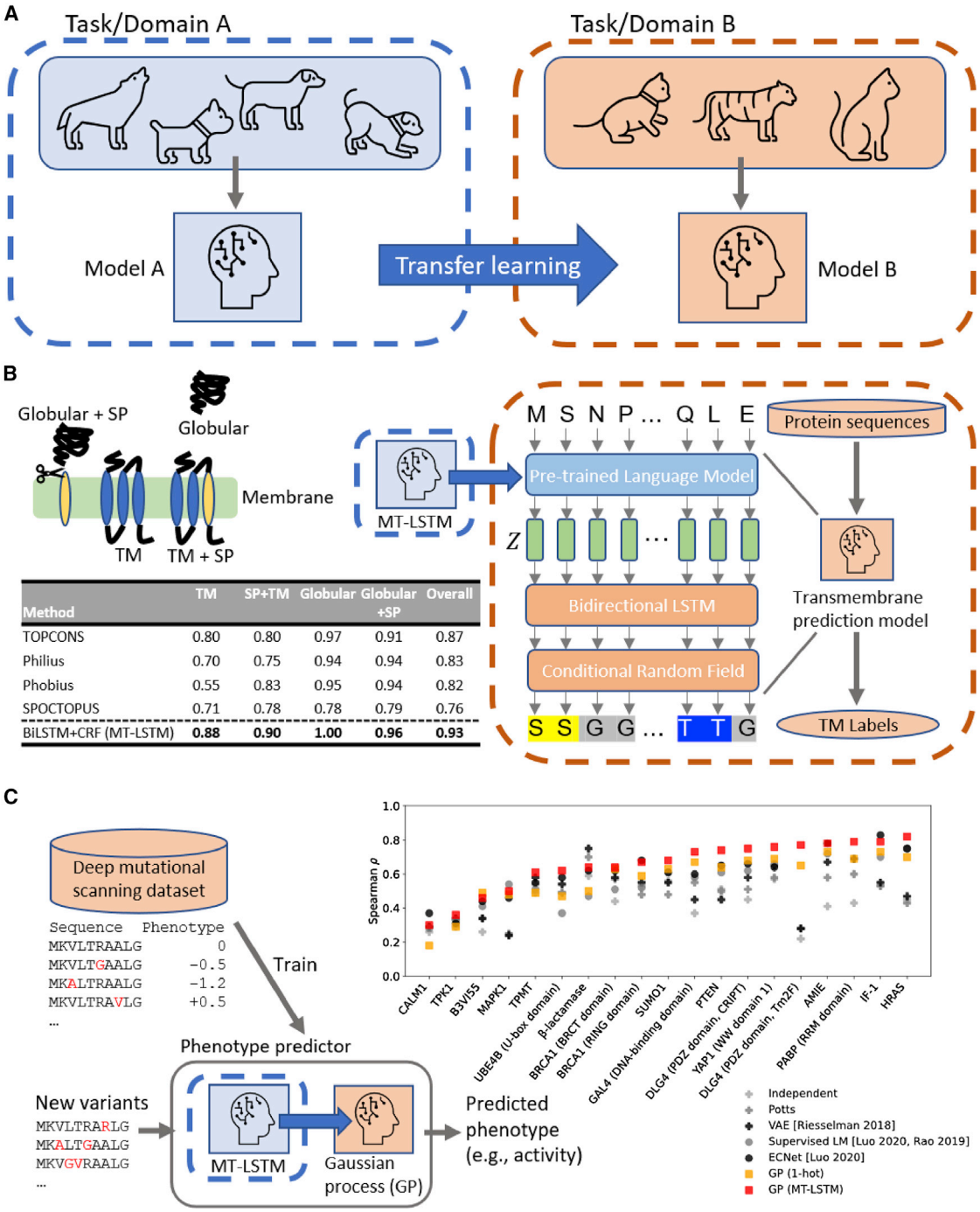
Here, we demonstrate two use cases where transfer learning from our MT-LSTM improves performance on downstream tasks. First, we consider the problem of transmembrane prediction. This is a sequence labeling task in which we are provided with the amino acid sequence of a protein and wish to decode, for each position of the protein, whether that position is in a transmembrane (i.e., membrane spanning) region of the protein or not. This problem is complicated by the presence of signal peptides, which are often confused as transmembrane regions.

In order to compare different sequence representations for this problem, we train a small one layer bidirectional LSTM with a conditional random field (BiLSTM+CRF) decoder on a well-defined transmembrane protein benchmark dataset (Tsirigos et al., 2015a). Methods are compared by 10-fold cross validation. We find that the BiLSTM+CRFs with our new embeddings (DLM-LSTM and MT-LSTM) outperform existing transmembrane predictors and a BiLSTM+CRF using our previous smaller embedding model (SSA-LSTM). Furthermore, representations learned by our MT-LSTM model significantly outperform (paired t test, $p = 0.044$) the embeddings learned by our DLM-LSTM model on this application (Figure 5B).

Second, we demonstrate that we are able to accurately predict functional implications of small changes in protein sequence through transfer learning. An ideal model would be sensitive down to the single amino acid level and would group mutations with similar functional outcomes closely in semantic space. Recently, Luo et al. presented a method for combining language model-based representations with local evolutionary context-based representations (ECNet) and demonstrated that these representations were powerful for sequence-to-phenotype mapping on a panel of deep mutational scanning datasets (Luo et al., 2020). In this problem, we observe a relatively small set (100 s-1000s) of sequence-phenotype measurement pairs and our goal is to predict phenotypes for unmeasured variants. Observing that these are small data problems, we reasoned that this is an ideal setting for Bayesian methods and that transfer learning will be important for achieving good performance. To this end, we propose a framework in which sequence variants are first embedded using our MT-LSTM and then phenotype predictions are made using Gaussian process (GP) regression using our embeddings as features. We find that we can predict the phenotypes of unobserved sequence variants across datasets better than existing methods (Figure 5C). Our MT-LSTM embedding powered GP achieves an average Spearman correlation of 0.65 with the measured phenotypes significantly outperforming (paired t test, $p = 0.006$) the next best method, ECNet, which reaches 0.60 average Spearman correlation.

Semi-supervised learning (van Engelen and Hoos, 2020), few-shot learning (Wang et al., 2020a), meta-learning (Vanschoren, 2018; Hospedales et al., 2020), and other methods for rapid adaptation to new problems and domains will be key future developments for pushing the limit of data efficient learning. Methods that capture uncertainty (e.g., Gaussian processes

Furthermore, our *entirely sequence-based method* even outperforms structural comparison with TMalign when predicting structural similarity in the SCOP database. Furthermore, we contrast our end-to-end MT-LSTM model with an earlier two-step language model (SSA-LSTM) and find that training end-to-end in a unified multi-task framework improves structural similarity classification.



Method	TM	SP+TM	Globular	Globular +SP	Overall
TOPCONS	0.80	0.80	0.97	0.91	0.87
Philius	0.70	0.75	0.94	0.94	0.83
Phobius	0.55	0.83	0.95	0.94	0.82
SPOCTOPUS	0.71	0.78	0.78	0.79	0.76
BiLSTM+CRF (MT-LSTM)	**0.88**	**0.90**	**1.00**	**0.96**	**0.93**

Figure 5. Protein language models with transfer learning improve function prediction

(A) Transfer learning is the problem of applying knowledge gained from learning to solve some task, A, to another related task, B. For example, applying knowledge from recognizing dogs to recognizing cats. Usually, transfer learning is used to improve performance on tasks with little available data by transferring knowledge from other tasks with large amounts of available data. In the case of proteins, we are interested in applying knowledge from evolutionary sequence modeling and structure modeling to protein function prediction tasks.

(B) Transfer learning improves transmembrane prediction. Our transmembrane prediction model consists of two components. First, the protein sequence is embedded using our pre-trained language model (MT-LSTM) by taking the hidden layers of the language model at each position. Then, these representations are fed into a small single layer bidirectional LSTM (BiLSTM) and the output of this is fed into a conditional random field (CRF) to predict the transmembrane label at each position. We evaluate the model by 10-fold cross validation on proteins split into four categories: transmembrane only (TM), signal peptide and transmembrane (TM+SP), globular only (Globular), and globular with signal peptide (Globular+SP). A protein is considered correctly predicted if 1) the presence or absence of signal peptide is correctly predicted and 2) the number of locations of transmembrane regions is correctly predicted. The table reports the fraction of correctly predicted proteins in each category for our model (BiLSTM+CRF) and widely used transmembrane prediction methods. A BiLSTM+CRF model trained

(legend continued on next page)

and other Bayesian methods) will continue to be important, particularly for guiding experimental design. Some recent works have explored Gaussian process-based methods for guiding protein design with simple protein sequence representations (Romero, Krause and Arnold, 2013; Bedbrook et al., 2017; Yang et al., 2018). Hie et al., presented a GP-based method for guiding experimental drug design informed by deep protein embeddings (Hie et al., 2020b). Other works have explored combining neural network and GP models (Ding et al., 2019; Patacchiola et al., 2019); while still others considered non-GP-based uncertainty aware prediction methods for antibody design and major histocompatibility complex (MHC) peptide display prediction (Zeng and Gifford, 2019; Liu et al., 2020). Methods for combining multiple predictors and for incorporating strong priors into protein design can also help to alleviate problems that arise in the low data regime (Brookes, Park and Listgarten, 2019). Transfer learning and massive protein language models will play a key role in future protein property prediction and machine learning driven protein and drug design efforts.

Conclusions and perspectives: Strong biological priors are key to improving protein language models

Future developments in protein language modeling and representation learning will need to model properties that are unique to proteins. Biological sequences are not natural language, and we should develop new language models that capture the fundamental nature of biological sequences. While demonstrably useful, existing methods based on recurrent neural networks and Transformers still do not fundamentally encode key protein properties in the model architecture and the inductive biases of these models are only roughly understood (Box 1).

Proteins are objects that exist in physical space. Similarly, we understand many of the fundamental evolutionary processes that give rise to the diversity of protein sequences observed today. These two elements, physics and evolution, are the key properties of proteins and our models might benefit from being structured explicitly to incorporate evolutionary and physics-based inductive biases. Early attempts at capturing physical properties of proteins as part of machine learning models have already demonstrated that conditioning on structure improves generative models of sequence (Ingraham et al., 2019b) and significant work has been done in the opposite direction of machine learning-based structure prediction methods that explicitly incorporate constraints on protein geometries (Liu et al., 2018; AlQuraishi, 2019; Ingraham et al., 2019b; Xu, 2019; Jumper et al., 2020; Yang et al., 2020). However, new methods are needed to fuse these directions with physics-based approaches and to start to fully merge sequence- and structure-based models.

At the same time, current protein language models make heavily simplified phylogenetic assumptions. By treating each sequence as an independent draw from some prior distribution over sequences, current methods assume that all protein sequences arise independently in a star phylogeny. Conventionally, this problem is crudely addressed by filtering sequences based on percent identity. However, significant effort has been dedicated to understanding protein sequences as emerging from tree-structured evolutionary processes over time or coalescent processes in reverse time (Rosenberg and Nordborg, 2002; Nascimento, Reis and Yang, 2017). Methods for inferring these latent phylogenetic trees continue to be of substantial interest (Huelsenbeck and Ronquist, 2001; Lartillot, Lepage and Blanquart, 2009; Bouckaert et al., 2019), but are frustrated by long run times and poor scalability to large datasets. In the future, deep generative models of proteins might seek to merge these disciplines to model proteins as being generated from evolutionary processes other than star phylogenies.

Other practical considerations continue to frustrate our ability to develop new protein language models and rapidly iterate on experiments. High compute costs and murky design guidelines mean that developing new models is often an expensive, time consuming, and *ad hoc* process. It is not clear at what dataset sizes and levels of sequence diversity one model will outperform another or how many parameters a model should include. At the upper limit of large natural protein databases, larger models continue to yield improved performance. However, for individual protein families or other application specific protein datasets, the gold standard is to select model architectures and number of parameters via brute force hyperparameter search methods. Fine-tuning pre-trained models can help with this problem but does not fully resolve it. Sequence length also remains a challenge for these models. Transformers scale quadratically with sequence length, which means that in practical implementations long sequences need to either be excluded or truncated. New linear complexity attention mechanisms may help to alleviate this limitation (Choromanski et al., 2020; Wang et al., 2020b). This problem is less extreme for recurrent neural networks, which scale linearly with sequence length, but very long sequences are still impractical for RNNs to handle and long-range sequence dependencies are unlikely to be learned well by these models.

Language models capture complex relationships between residues in protein sequences by condensing information from enormous protein sequence databases. They are a powerful new development for understanding and making predictions about biological sequences. Increasing model size, compute power, and dataset size will only continue to improve performance of protein language models. Already, these methods are transforming computational protein biology today due to their ease

using 1-hot embeddings of the protein sequence instead of our language model representations performs poorly, highlighting the importance of transfer learning for this task (Table S2).

(C) Transfer learning improves sequence-to-phenotype prediction. Deep mutational scanning measures function for thousands of protein sequence variants. We consider 19 mutational scanning datasets spanning a variety of proteins and phenotypes. For each dataset, we learn the sequence-to-phenotype mapping by fitting a Gaussian process regression model on top of representations given by our pre-trained language model. We compare three unsupervised approaches (+), prior works in supervised learning (o), and our Gaussian process regression approaches with (ÿ, GP (MT-LSTM)) and without (GP (1-hot)) transfer learning by 5-fold cross validation. Spearman rank correlation coefficients between predicted and ground truth functional measurements are plotted. Our GP with transfer learning outperforms all other methods, having an average correlation of 0.65 across datasets. The benefits of transfer learning are highlighted by the improvement over the 1-hot representations which only reach 0.57 average correlation across datasets. Transfer learning improves performance on 18 out of 19 datasets.

of use and widespread applicability. Furthermore, augmenting language models with protein specific properties such as structure and function offers one already successful route toward even richer representations and novel biology. However, it remains unclear how best to encode prior biological knowledge into the inductive bias of these models. We hope this Synthesis propels the community to work toward developing purpose-built protein language models with natural inductive biases suited for the physical nature of proteins and how they evolve.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- RESOURCE AVAILABILITY
  - Lead contact
  - Materials availability
  - Data and code availability
- METHOD DETAILS
  - Bidirectional LSTM encoder with skip connections
  - Masked language modeling module
  - Residue-residue contact prediction module
  - Structure similarity prediction module
  - Multi-task loss
  - Training datasets
  - Hyperparameters and training details
  - Protein structural similarity prediction evaluation
  - Transmembrane region prediction training and evaluation
  - Sequence-to-phenotype prediction and evaluation

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.cels.2021.05.017.

ACKNOWLEDGMENTS

The authors are grateful to Grace Yeo and Brian Hie for helpful suggestions. T.B. is supported by the Simons Foundation International, Ltd. (SF349247). B.B. is partially supported by NIH grant R35 GM141861.

AUTHOR CONTRIBUTIONS

All authors conceived and guided the project and methodology. T.B. wrote the software and performed the computational experiments. All authors interpreted the results and wrote the manuscript.

DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: January 16, 2021
Revised: May 20, 2021
Accepted: May 20, 2021
Published: June 16, 2021

REFERENCES

Alford, R.F., Leaver-Fay, A., Jeliazkov, J.R., O'Meara, M.J., DiMaio, F.P., Park, H., Shapovalov, M.V., Renfrew, P.D., Mulligan, V.K., Kappel, K., et al. (2017). The Rosetta All-Atom Energy Function for Macromolecular Modeling and Design. J. Chem. Theory Comput. *13*, 3031–3048.

Alley, E.C., Khimulya, G., Biswas, S., AlQuraishi, M., and Church, G.M. (2019). Unified rational protein engineering with sequence-based deep representation learning. Nat. Methods *16*, 1315–1322.

AlQuraishi, M. (2019). End-to-End Differentiable Learning of Protein Structure. Cell Syst. *8*, 292–301.e3.

Altschul, S.F., and Koonin, E.V. (1998). Iterated profile searches with PSI-BLAST–a tool for discovery in protein databases. Trends Biochem. Sci. *23*, 444–447.

Araya, C.L., Fowler, D.M., Chen, W., Muniez, I., Kelly, J.W., and Fields, S. (2012). A fundamental protein property, thermodynamic stability, revealed solely from large-scale measurements of protein function. Proc. Natl. Acad. Sci. USA *109*, 16858–16863.

Bandaru, P., Shah, N.H., Bhattacharyya, M., Barton, J.P., Kondo, Y., Cofsky, J.C., Gee, C.L., Chakraborty, A.K., Kortemme, T., Ranganathan, R., and Kuriyan, J. (2017). Deconstruction of the Ras switching cycle through saturation mutagenesis. eLife *6*, e27810. https://doi.org/10.7554/eLife.27810.

Bateman, A., Coin, L., Durbin, R., Finn, R.D., Hollich, V., Griffiths-Jones, S., Khanna, A., Marshall, M., Moxon, S., Sonnhammer, E.L., et al. (2004). The Pfam protein families database. Nucleic Acids Res. *32*, D138–D141.

Battaglia, P.W., Hamrick, J.B., and Bapst, V. (2018). Relational inductive biases, deep learning, and graph networks. arXiv, 1806.01261 https://arxiv.org/abs/1806.01261.

Bedbrook, C.N., Yang, K.K., Rice, A.J., Gradinaru, V., and Arnold, F.H. (2017). Machine learning to design integral membrane channelrhodopsins for efficient eukaryotic expression and plasma membrane localization. PLoS Comput. Biol. *13*, e1005786.

Bengio, Y. (2012) Deep learning of representations for unsupervised and transfer learning. In Proceedings of ICML workshop on unsupervised and transfer learning. jmlr.org, pp. 17–36.

Bepler, T., and Berger, B. (2019). Learning protein sequence embeddings using information from structure. In International Conference on Learning Representations. 1902.08661, https://arxiv.org/abs/1902.08661.

Berger, B. (1995). Algorithms for protein structural motif recognition. J. Comput. Biol. *2*, 125–138.

Berger, B., Wilson, D.B., Wolf, E., Tonchev, T., Milla, M., and Kim, P.S. (1995). Predicting coiled coils by use of pairwise residue correlations. Proc. Natl. Acad. Sci. USA *92*, 8259–8263.

Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., and Bourne, P.E. (2000). The Protein Data Bank. Nucleic Acids Res. *28*, 235–242.

Bouckaert, R., Vaughan, T.G., Barido-Sottani, J., Duchêne, S., Fourment, M., Gavryushkina, A., Heled, J., Jones, G., Kühnert, D., De Maio, N., et al. (2019). BEAST 2.5: An advanced software platform for Bayesian evolutionary analysis. PLoS Comput. Biol. *15*, e1006650.

Brenan, L., Andreev, A., Cohen, O., Pantel, S., Kamburov, A., Cacchiarelli, D., Persky, N.S., Zhu, C., Bagul, M., Goetz, E.M., et al. (2016). Phenotypic Characterization of a Comprehensive Set of MAPK1/ERK2 Missense Mutants. Cell Rep. *17*, 1171–1183.

Brookes, D., Park, H., and Listgarten, J. (2019) 'Conditioning by adaptive sampling for robust design', in Chaudhuri, K. and Salakhutdinov, R. (eds) *Proceedings of the 36th International Conference on Machine Learning*. PMLR (Proceedings of Machine Learning Research), pp. 773–782.

Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language Models are Few-Shot Learners. arXiv, 2005.14165 http://arxiv.org/abs/2005.14165.

Callaway, E. (2020). Revolutionary cryo-EM is taking over structural biology. Nature *578*, 201.

Chandonia, J.-M., Fox, N.K., and Brenner, S.E. (2017). SCOPe: Manual Curation and Artifact Removal in the Structural Classification of Proteins - extended Database. J. Mol. Biol. *429*, 348–355.

Cheng, Y., Grigorieff, N., Penczek, P.A., and Walz, T. (2015). A primer to single-particle cryo-electron microscopy. Cell *161*, 438–449.

Choi, J.-M., and Pappu, R.V. (2019). Improvements to the ABSINTH Force Field for Proteins Based on Experimentally Derived Amino Acid Specific Backbone Conformational Statistics. J. Chem. Theory Comput. *15*, 1367–1382.

Choromanski, K.M., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., Hawkins, P., Davis, J.Q., Mohiuddin, A., Kaiser, L., et al. (2020). Rethinking Attention with Performers. In International Conference on Learning Representations. https://openreview.net/pdf?id=Ua6zuk0WRH (Accessed: 20 May 2021).

de Juan, D., Pazos, F., and Valencia, A. (2013). Emerging methods in protein co-evolution. Nat. Rev. Genet. *14*, 249–261.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv, 1810.04805 http://arxiv.org/abs/1810.04805.

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

Eddy, S.R. (2011). Accelerated Profile HMM Searches. PLoS Comput. Biol. *7*, e1002195.

Ekeberg, M., Lövkvist, C., Lan, Y., Weigt, M., and Aurell, E. (2013). Improved contact prediction in proteins: using pseudolikelihoods to infer Potts models. Phys. Rev. E Stat. Nonlin. Soft Matter Phys. *87*, 012707.

Elnaggar, A., Heinzinger, M., Dallago, C., Rihawi, G., Wang, Y., Jones, L., Gibbs, T., Feher, T., Angerer, C., Steinegger, M., et al. (2020). ProtTrans: Towards Cracking the Language of Life's Code Through Self-Supervised Deep Learning and High Performance Computing. arXiv, 2007.06225 http://arxiv.org/abs/2007.06225.

Findlay, G.M., Daza, R.M., Martin, B., Zhang, M.D., Leith, A.P., Gasperini, M., Janizek, J.D., Huang, X., Starita, L.M., and Shendure, J. (2018). Accurate classification of BRCA1 variants with saturation genome editing. Nature *562*, 217–222.

Finn, R.D., Clements, J., and Eddy, S.R. (2011). HMMER web server: interactive sequence similarity searching. Nucleic Acids Res. *39*, 29–37.

Fox, N.K., Brenner, S.E., and Chandonia, J.-M. (2014). SCOPe: Structural Classification of Proteins–extended, integrating SCOP and ASTRAL data and classification of new structures. Nucleic Acids Res. *42*, D304–D309.

Gardner, J., et al. (2018). GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration. In Advances in Neural Information Processing Systems, S. Bengio, et al., eds. (Curran Associates, Inc), pp. 7576–7586.

Göbel, U., Sander, C., Schneider, R., and Valencia, A. (1994). Correlated mutations and residue contacts in proteins. Proteins *18*, 309–317.

Godzik, A., Kolinski, A., and Skolnick, J. (1993). De novo and inverse folding predictions of protein structure and dynamics. J. Comput. Aided Mol. Des. *7*, 397–438.

Graves, A., Fernández, S., and Schmidhuber, J. (2005). Bidirectional LSTM Networks for Improved Phoneme Classification and Recognition. In Artificial Neural Networks: Formal Models and Their Applications – ICANN 2005 (Springer Berlin Heidelberg), pp. 799–804.

Harris, Z.S. (1954). Distributional Structure. Word World *10*, 146–162.

Hess, B., Kutzner, C., van der Spoel, D., and Lindahl, E. (2008). GROMACS 4:cAlgorithms for Highly Efficient, Load-Balanced, and Scalable Molecular Simulation. J. Chem. Theory Comput. *4*, 435–447.

Hie, B., Zhong, E., Bryson, B., and Berger, B. (2020a). Learning mutational semantics. Advances in Neural Information Processing Systems *33*. https://proceedings.neurips.cc/paper/2020/hash/6754e06e46dfa419d5afe3c9781cecad-Abstract.html.

Hie, B., Bryson, B.D., and Berger, B. (2020b). Leveraging Uncertainty in Machine Learning Accelerates Biological Discovery and Design. Cell Syst. *11*, 461–477.e9.

Hie, B., Zhong, E.D., Berger, B., and Bryson, B. (2021). Learning the language of viral evolution and escape. Science *371*, 284–288.

Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. Neural Comput. *9*, 1735–1780.

Hornak, V., Abel, R., Okur, A., Strockbine, B., Roitberg, A., and Simmerling, C. (2006). Comparison of multiple Amber force fields and development of improved protein backbone parameters. Proteins *65*, 712–725.

Hospedales, T., Antoniou, A., Micaelli, P., and Storkey, A. (2020). Meta-Learning in Neural Networks: A Survey. arXiv, 2004.05439 http://arxiv.org/abs/2004.05439.

Hubbard, T.J., Murzin, A.G., Brenner, S.E., and Chothia, C. (1997). SCOP: a structural classification of proteins database. Nucleic Acids Res. *25*, 236–239.

Huelsenbeck, J.P., and Ronquist, F. (2001). MRBAYES: Bayesian inference of phylogenetic trees. Bioinformatics *17*, 754–755.

Ingraham, J., Garg, V.K., Barzilay, R., and Jaakkola, T. (2019a). Generative Models for Graph-Based Protein Design. In Advances in Neural Information Processing Systems, H. Wallach, et al., eds. (Curran Associates, Inc.), pp. 15820–15831.

Ingraham, J., Riesselman, A., Sander, C., and Marks, D. (2019b). Learning protein structure with a differentiable simulator. In International Conference on Learning Representations https://openreview.net/forum?id=Byg3y3C9Km.

Jacquier, H., Birgy, A., Le Nagard, H., Mechulam, Y., Schmitt, E., Glodt, J., Bercot, B., Petit, E., Poulain, J., Barnaud, G., et al. (2013). Capturing the mutational landscape of the beta-lactamase TEM-1. Proc. Natl. Acad. Sci. USA *110*, 13067–13072.

James, L.C., and Tawfik, D.S. (2003). Conformational diversity and protein evolution–a 60-year-old hypothesis revisited. Trends Biochem. Sci. *28*, 361–368.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Tunyasuvunakool, K., Ronneberger, O., Bates, R., Zidek, A., Brigland, A., et al.. https://predictioncenter.org/casp14/doc/CASP14_Abstracts.pdf.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T.B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. (2020). Scaling Laws for Neural Language Models. arXiv, 2001.08361 http://arxiv.org/abs/2001.08361.

Kingma, D.P., and Ba, J. (2015). Adam: A Method for Stochastic Optimization. In International Conference on Learning Representations. 1412.6980, http://arxiv.org/abs/1412.6980.

Kitzman, J.O., Starita, L.M., Lo, R.S., Fields, S., and Shendure, J. (2015). Massively parallel single-amino-acid mutagenesis. Nat. Methods *12*, 203–206, 4 p following 206.

Klesmith, J.R., Bacik, J.-P., Wrenbeck, E.E., Michalczyk, R., and Whitehead, T.A. (2017). Trade-offs between enzyme fitness and solubility illuminated by deep mutational scanning. PNAS *114*, 2265–2270.

Kosloff, M., and Kolodny, R. (2008). Sequence-similar, structure-dissimilar protein pairs in the PDB. Proteins *71*, 891–902.

Lartillot, N., Lepage, T., and Blanquart, S. (2009). PhyloBayes 3: a Bayesian software package for phylogenetic reconstruction and molecular dating. Bioinformatics *25*, 2286–2288.

Leaver-Fay, A., Tyka, M., Lewis, S.M., Lange, O.F., Thompson, J., Jacak, R., Kaufman, K., Renfrew, P.D., Smith, C.A., et al. (2011). Chapter nineteen - Rosetta3: An Object-Oriented Software Suite for the Simulation and Design of Macromolecules. In Methods in Enzymology, M.L. Johnson and L. Brand, eds. (Academic Press), pp. 545–574.

Liu, Y., Palmedo, P., Ye, Q., Berger, B., and Peng, J. (2018). Enhancing Evolutionary Couplings with Deep Convolutional Neural Networks. Cell Syst. *6*, 65–74.e3.

Liu, G., Zeng, H., Mueller, J., Carter, B., Wang, Z., Schilz, J., Horny, G., Birnbaum, M.E., Ewert, S., and Gifford, D.K. (2020). Antibody complementarity determining region design using high-capacity machine learning. Bioinformatics *36*, 2126–2133.

Luo, Y., Vo, L., Ding, H., Su, Y., Liu, Y., Qian, W.W., Zhao, H., and Peng, J. (2020). Evolutionary Context-Integrated Deep Sequence Modeling for Protein Engineering. In Research in Computational Molecular Biology (Springer International Publishing), pp. 261–263.

Madani, A., McCann, B., Naik, N., Keskar, N.S., Anand, N., Eguchi, R.R., Huang, P.-S., and Sochler, R. (2020). ProGen: Language Modeling for Protein Generation. arXiv, 2004.03497 http://arxiv.org/abs/2004.03497.

Marks, D.S., Hopf, T.A., and Sander, C. (2012). Protein structure prediction from sequence variation. Nat. Biotechnol. *30*, 1072–1080.

Matreyek, K.A., Starita, L.M., Stephany, J.J., Martin, B., Chiasson, M.A., Gray, V.E., Kircher, M., Khechaduri, A., Dines, J.N., Hause, R.J., et al. (2018). Multiplex assessment of protein variant abundance by massively parallel sequencing. Nat. Genet. *50*, 874–882.

McDonnell, A.V., Jiang, T., Keating, A.E., and Berger, B. (2006). Paircoil2: improved prediction of coiled coils from sequence. Bioinformatics *22*, 356–358.

McLaughlin, R.N., Jr., Poelwijk, F.J., Raman, A., Gosal, W.S., and Ranganathan, R. (2012). The spatial architecture of protein function and adaptation. Nature *491*, 138–142.

Melamed, D., Young, D.L., Gamble, C.E., Miller, C.R., and Fields, S. (2013). Deep mutational scanning of an RRM domain of the Saccharomyces cerevisiae poly(A)-binding protein. RNA *19*, 1537–1551.

Mitchell, T. M. 1980. The need for biases in learning generalizations. New Jersey: Department of Computer Science, Laboratory for Computer Science Research, Rutgers Univ.; 1980 May.

Nascimento, F.F., Reis, M.D., and Yang, Z. (2017). A biologist's guide to Bayesian phylogenetic analysis. Nat. Ecol. Evol. *1*, 1446–1454.

Needleman, S.B., and Wunsch, C.D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. J. Mol. Biol. *48*, 443–453.

Patacchiola, M., Turner, J., Crowley, E.J., O'Boyle, M., and Storkey, A. (2019). Bayesian Meta-Learning for the Few-Shot Setting via Deep Kernels. arXiv, 1910.05199 https://arxiv.org/abs/1910.05199.

Peters, M.E., Neumann M, and Iyyer M. (2018). Deep contextualized word representations. arXiv, 1802.05365 http://arxiv.org/abs/1802.05365.

Pruitt, K.D., Tatusova, T., and Maglott, D.R. (2007). NCBI reference sequences (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. Nucleic Acids Res. *35*, D61–D65.

Paszke, A., Gross, S., Chintala, S., et al.. https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018) Improving language understanding by generative pre-training. cs.ubc.ca. https://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf (Accessed: 14 January 2021).

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI blog *1*. https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Rao, R., Bhattacharya, N., Thomas, N., Duan, Y., Chen, X., Canny, J., Abbeel, P., and Song, Y.S. (2019). Evaluating Protein Transfer Learning with TAPE. Adv. Neural Inf. Process. Syst. *32*, 9689–9701.

Rao, R., Meier, J., Sercu, T., Ovchinnikov, S., and Rives, A. (2020). Transformer protein language models are unsupervised structure learners. bioRxiv. https://doi.org/10.1101/2020.12.15.422761.

Rasmussen, C.E., and Williams, C.K.I. (2005). Gaussian Processes for Machine Learning. (MIT Press).

Remmert, M., Biegert, A., Hauser, A., and Söding, J. (2011a). HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. Nat. Methods *9*, 173–175.

Remmert, M., Biegert, A., Hauser, A., and Söding, J. (2011b). HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. Nat. Methods *9*, 173–175.

Remmert, M., Biegert, A., Hauser, A., and Söding, J. (2011c). HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. Nat. Methods *9*, 173–175.

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

Rives, A., Goyal, S., Meier, J., Guo, D., Ott, M., Zitnick, C.L., Ma, J., and Fergus, R. (2019). Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. bioRxiv https://www.biorxiv.org/content/10.1101/622803v1.

Rohl, C.A., Strauss, C.E., Misura, K.M., and Baker, D. (2004). Protein structure prediction using Rosetta. Methods Enzymol. *383*, 66–93.

Romero, P.A., Krause, A., and Arnold, F.H. (2013). Navigating the protein fitness landscape with Gaussian processes. Proc. Natl. Acad. Sci. USA *110*, E193–E201.

Rosenberg, N.A., and Nordborg, M. (2002). Genealogical trees, coalescent theory and the analysis of genetic polymorphisms. Nat. Rev. Genet. *3*, 380–390.

Rost, B. (1999). Twilight zone of protein sequence alignments. Protein Eng. *12*, 85–94.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L., et al. (2015). ImageNet Large Scale Visual Recognition Challenge. Int. J. Comput. Vis. *115*, 211–252.

Shin, D.H., Hou, J., Chandonia, J.M., Das, D., Choi, I.G., Kim, R., and Kim, S.H. (2007). Structure-based inference of molecular functions of proteins of unknown function from Berkeley Structural Genomics Center. J. Struct. Funct. Genomics *8*, 99–105.

Sillitoe, I., Bordin, N., Dawson, N., Waman, V.P., Ashford, P., Scholes, H.M., Pang, C.S.M., Woodridge, L., Rauer, C., Sen, N., et al. (2021). CATH: increased structural coverage of functional space. Nucleic Acids Res. *49* (D1), D266–D273.

Sledzieski, S., Singh, R., Cowen, L., and Berger, B. (2021). Sequence-based prediction of protein-protein interactions: a structure-aware interpretable deep learning model. In Research in Computational Molecular Biology (RECOMB). https://doi.org/10.1101/2021.01.22.427866.

Srinivasan, R., and Rose, G.D. (1995). LINUS: a hierarchic procedure to predict the fold of a protein. Proteins *22*, 81–99. https://doi.org/10.1002/prot.340220202.

Starita, L.M., Pruneda, J.N., Lo, R.S., Fowler, D.M., Kim, H.J., Hiatt, J.B., Shendure, J., Brzovic, P.S., Fields, S., and Klevit, R.E. (2013). Activity-enhancing mutations in an E3 ubiquitin ligase identified by high-throughput mutagenesis. Proc. Natl. Acad. Sci. USA *110*, E1263–E1272.

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

Trigg, J., Gutwin, K., Keating, A.E., and Berger, B. (2011). Multicoil2: predicting coiled coils and their oligomerization states from sequence in the twilight zone. PLoS ONE *6*, e23519.

Truong, T.F., and Truong, T.F., Jr. (2020). Interpretable deep learning framework for binding affinity prediction. Massachusetts Institute of Technology https://dspace.mit.edu/handle/1721.1/127527?show=full.

Tsirigos, K.D., Peters, C., Shu, N., Käll, L., and Elofsson, A. (2015a). The TOPCONS web server for consensus prediction of membrane protein topology and signal peptides. Nucleic Acids Res. *43* (W1), W401-7.

Tsirigos, K.D., Peters, C., Shu, N., Käll, L., and Elofsson, A. (2015b). The TOPCONS web server for consensus prediction of membrane protein topology and signal peptides. Nucleic Acids Res. *43* (W1), W401-7.

UniProt Consortium (2019). UniProt: a worldwide hub of protein knowledge. Nucleic Acids Res. *47* (D1), D506–D515.

van Engelen, J.E., and Hoos, H.H. (2020). A survey on semi-supervised learning. Mach. Learn. *109*, 373–440.

Vanschoren, J. (2018). Meta-Learning: A Survey. arXiv, 1810.03548 http://arxiv.org/abs/1810.03548.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., and Polosukhin, I. (2017). Attention is All you Need. In Advances in Neural Information Processing Systems, I. Guyon, et al., eds. (Curran Associates, Inc.), pp. 5998–6008.

Vig, J., Madani, A., Varshney, L.R., Xiong, C., Socher, R., and Rajani, N.F. (2020). BERTology Meets Biology: Interpreting Attention in Protein Language Models. arXiv, 2006.15222 http://arxiv.org/abs/2006.15222.

Wang, S., Sun, S., Li, Z., Zhang, R., and Xu, J. (2017). Accurate De Novo Prediction of Protein Contact Map by Ultra-Deep Learning Model. PLoS Comput. Biol. *13*, e1005324.

Walensky, R.P., Walke, H.T., and Fauci, A.S. (2021). SARS-CoV-2 Variants of Concern in the United States—Challenges and Opportunities. JAMA *325* (11), 1037–1038. https://doi.org/10.1001/jama.2021.2294.

Wang, S., Li, B.Z., Khabsa, M., Fang, H., and Ma, H. (2020a). Linformer: Self-Attention with Linear Complexity. arXiv, 2006.04768 http://arxiv.org/abs/2006.04768.

Wang, Y., Yao, Q., Kwok, J., and Ni, L.M. (2020b). Generalizing from a Few Examples: A Survey on Few-shot Learning. ACM Comput. Surv. *53*, 1–34.

Wei, K.Y., Moschidi, D., Bick, M.J., Nerli, S., McShan, A.C., Carter, L.P., Huang, P.S., Fletcher, D.A., Sgourakis, N.G., Boyken, S.E., and Baker, D. (2020). Computational design of closely related proteins that adopt two well-defined but structurally divergent folds. Proc. Natl. Acad. Sci. USA *117*, 7208–7215.

Weile, J., Sun, S., Cote, A.G., Knapp, J., Verby, M., Mellor, J.C., Wu, Y., Pons, C., Wong, C., van Lieshout, N., et al. (2017). A framework for exhaustively mapping functional missense variants. Mol. Syst. Biol. *13*, 957.

Wolf, E., Kim, P.S., and Berger, B. (1997). MultiCoil: a program for predicting two- and three-stranded coiled coils. Protein Sci. *6*, 1179–1189.

Wrenbeck, E.E., Azouz, L.R., and Whitehead, T.A. (2017). Single-mutation fitness landscapes for an enzyme on multiple substrates reveal specificity is globally encoded. Nat. Commun. *8*, 15695.

Xu, J. (2019). Distance-based protein folding powered by deep learning. Proc. Natl. Acad. Sci. USA *116*, 16856–16865.

Xu, J., and Wang, S. (2019). Analysis of distance-based protein structure prediction by deep learning in CASP13. Proteins *87*, 1069–1081.

Yang, K.K., Wu, Z., Bedbrook, C.N., and Arnold, F.H. (2018). Learned protein embeddings for machine learning. Bioinformatics *34*, 4138.

Yang, J., Anishchenko, I., Park, H., Peng, Z., Ovchinnikov, S., and Baker, D. (2020). Improved protein structure prediction using predicted interresidue orientations. Proc. Natl. Acad. Sci. USA *117*, 1496–1503.

Zeng, H., and Gifford, D.K. (2019). Quantification of Uncertainty in Peptide-MHC Binding Prediction Improves High-Affinity Peptide Selection for Therapeutic Design. Cell Syst. *9*, 159–166.e3.

Zhang, C., and Kim, S.-H. (2003). Overview of structural genomics: from structure to function. Curr. Opin. Chem. Biol. *7*, 28–32.

Zhang, Y., and Skolnick, J. (2005). TM-align: a protein structure alignment algorithm based on the TM-score. Nucleic Acids Res. *33*, 2302–2309.

Zhou, G., Chen, M., Ju, C.J.T., Wang, Z., Jiang, J.Y., and Wang, W. (2020). Mutation effect estimation on protein-protein interactions using deep contextualized representation learning. NAR Genom. Bioinform. *2*, a015. https://doi.org/10.1093/nargab/lqaa015.

STAR★METHODS

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Bonnie Berger (bab@mit.edu).

Materials availability

This study did not generate new materials.

Data and code availability

- This paper did not generate new data.
- Source code and model parameters are available at https://github.com/tbepler/prose.
- Any additional information required to reanalyze the data reported in this paper is available from the lead contact upon request.

METHOD DETAILS

Bidirectional LSTM encoder with skip connections

We structure the sequence encoder of our DLM- and MT-LSTM models as a three-layered bidirectional LSTM with skip connections from each layer to the final output. Our LSTMs have 1024 hidden units in each direction of each layer. We feed a 1-hot encoding of the amino acid sequence as the input to the first layer. Given a sequence input, x, of length L, this sequence is 1-hot encoded into a matrix, O, of size Lx21 where entry $o_{i,j} = 1$ if $x_i = j$ (that is, amino acid x_i has index j) and $o_{i,j} = 0$ otherwise. We then calculate $H^{(1)} = f^{(1)}(O)$, $H^{(2)} = f^{(2)}(H^{(1)})$, $H^{(3)} = f^{(3)}(H^{(2)})$, and $Z = [H^{(1)}\ H^{(2)}\ H^{(3)}]$ where $H^{(a)}$ is the hidden units of the ath layer and $f^{(a)}$ is the ath BiLSTM layer. The final output of the encoder, Z, is the concatenation of the hidden units of each layer along the embedding dimension.

Masked language modeling module

We use a masked language modeling objective for training on sequences only. During training, we randomly replace 10% of the amino acids in a sequence with either an auxiliary mask token or a uniformly random draw from the amino acids and train our model to predict the original amino acids at those positions. Given an input sequence, x, we randomly mask this sequence to create a new sequence, x'. This sequence is fed into our encoder to give a sequence of vector representations, Z. We decode these vectors into a distribution over amino acids at each position, p, using a linear layer. The parameters of this layer are learned jointly with the parameters of the encoder network. We calculate the masked language modeling loss as the negative log likelihood of the true amino acid at each of the masked positions, $L_{masked} = -\frac{1}{n}\sum_i log\ p_{i,x_i}$ where there are n masked positions indexed by i.

Residue-residue contact prediction module

We predict intra-residue contacts using a bilinear projection of the sequence embeddings. Given a sequence, x, with embeddings, Z, calculated using our encoder network, the bilinear projection calculates $ZWZ^T + b$, where W and b are learnable parameters of dimension DxD and 1 respectively where D is the dimension of an embedding vector. These parameters are fit together with the parameters of the encoder network. This produces an LxL matrix, where L is the length of x. We interpret the i,jth entry in this matrix as the log-likelihood ratio between the probability that the ith and jth residues are within 8 Å in the 3D protein structure and the probability that they are not. We then calculate the contact loss, $L_{contact}$, as the negative log-likelihood of the true contacts given the predicted contact probabilities.

Structure similarity prediction module

Our structure similarity prediction module follows previously described methods (Bepler and Berger, 2019). Given two input sequences, X and X' with lengths N and M, that have been encoded into vector representations, Z and Z', we calculate reduced dimension projections, A = ZB and A' = Z'B, where B is a DxK matrix that is trained together with the encoder network parameters. K is a hyperparameter and is set to 100. Given A and A', we calculate the inter-residue semantic distances between the two sequences as the Manhattan distance between the embedding at position i in the first sequence and embedding at position j in the second sequence, $d_{i,j} = ||A_i - A'_j||_1$. Given these distances, we calculate a soft alignment between the positions of sequences X and X'. The alignment weight between two positions, i and j, is defined as $c_{ij} = \alpha_{ij} + \beta_{ij} - \alpha_{ij}\beta_{ij}$ where $\alpha_{ij} = \frac{k_{ij}}{\sum_{l=1}^{N} k_{ij}}$ and $\beta_{ij} = \frac{k_{ij}}{\sum_{l=1}^{M} k_{ij}}$ and $k_{ij} = e^{-d_{ij}}$. With the inter-residue semantic distances and the alignment weights, we then define a global similarity between the two sequences as the negative semantic distance between the positions averaged over the alignment, $s = -\frac{1}{C}\sum_{ij} c_{ij} d_{ij}$ where $c = \sum_{ij} c_{ij}$.

With this global similarity based on the sequence embeddings in hand, we need to compare it against a ground truth similarity to calculate the gradient of our loss signal and update the parameters. Because we want our semantic similarity to reflect structural similarity, we retrieve ground truth labels, t, from the SCOP database by assigning increasing levels of similarity to proteins based on the number of levels in the SCOP hierarchy that they share. In other words, we assign a ground truth label of 0 to proteins not in the same class, 1 to proteins in the same class but not the same fold, 2 to proteins in the same fold but not the same superfamily, 3 to proteins in the same superfamily but not in the same family, and finally 4 to proteins in the same family. We relate our semantic similarity to these levels of structural similarity through ordinal regression. We calculate the probability that two sequences are similar at a level t or higher as $p(y \geq t) = \theta_t s + b_t$ where θ_t and b_t are additional learnable parameters for $t \geq 1$. We impose the constraint that $\theta_t \geq 0$ in order to ensure that increasing similarity between the embeddings corresponds to increasing numbers of shared levels in the SCOP hierarchy. Given these distributions, we calculate the probability that two proteins are similar at exactly level t as $p(y = t) = p(y \geq t)(1 - p(y \geq t + 1))$. That is, the probability that two sequences are similar at exactly level t is equal to the probability they are similar at at least level t times the probability they are not similar at a level above t.

We then define the structural similarity prediction loss to be the negative log-likelihood of the observed similarity labels under this model, $L_{similarity} = -\log p(y = t)$.

Multi-task loss

We define the combined multi-task loss as a weighted sum of the language modeling, contact prediction, and similarity prediction losses, $L_{MT} = \lambda_{masked} L_{masked} + \lambda_{contact} L_{contact} + \lambda_{similarity} L_{similarity}$.

Training datasets

We train our masked language models on a large corpus of protein sequences, UniRef90 (Suzek et al., 2007), retrieved in July 2018. This dataset contains 76,215,872 protein sequences filtered to 90% sequence identity. For structural supervision, we use the SCOPe ASTRAL protein dataset previously presented by Bepler & Berger (Fox, Brenner and Chandonia, 2014; Chandonia, Fox and Brenner, 2017; Bepler and Berger, 2019). This dataset contains 28,010 protein sequences with known structures and SCOP classifications from the SCOPe ASTRAL 2.06 release. These sequences are split into 22,408 training sequences and 5,602 testing sequences.

Hyperparameters and training details

We train two language models with different settings of the weights in the loss term. The first model, DLM-LSTM, uses only the masked language modeling objective so is trained with $\lambda_{masked} = 1$, $\lambda_{contact} = 0$, and $\lambda_{similarity} = 0$.The second model, MT-LSTM, uses the full multi-task objective with weights $\lambda_{masked} = 0.5$, $\lambda_{contact} = 0.9$, and $\lambda_{similarity} = 0.1$.The DLM-LSTM model was trained for 1,000,000 parameter updates using a minibatch size of 100 using the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 0.0001. The MT-LSTM model was also trained for 1,000,000 parameter updates using Adam with a learning rate of 0.0001, but, due to GPU RAM restrictions, we had to train the MT-LSTM model with smaller minibatch sizes of 64 for the masked language model objective and 16 for the structure-based objectives. Following Bepler & Berger (Bepler and Berger, 2019), we sampled pairs of proteins for the structural similarity prediction task with an exponential smoothing parameter, $\tau = 0.5$, in order to oversample the relatively rare highly similar protein pairs in the dataset. During training, we applied a mild regularization on the structure tasks by randomly resampling positions from a uniform distribution over amino acids with probability 0.05.

Models were implemented using PyTorch (Paszke et al., 2017) and trained on a single NVIDIA V100 GPU with 32GB of RAM. Training time was roughly 13 days for the DLM-LSTM model and 51 days for the MT-LSTM model.

Protein structural similarity prediction evaluation

We evaluate protein structural similarity methods on the SCOPe ASTRAL test set described above (Training datasets). All methods are evaluated on 100,000 randomly sampled protein pairs in this dataset. For each prediction method, we calculate the predicted similarity between each pair using only the sequence of each protein with the exception of TMalign which operates on the protein structures. Because TMscore is not symmetric, we calculate TMscore for both comparison directions and average them together for each protein pair. We found this outperformed other methods of combining the two scores. For HHalign, we first constructed profile HMMs for each protein by iteratively searching for homologs in the uniprot30 database provided by the authors using HHblits (Remmert et al., 2011c). We then calculate the similarities between each pair of proteins by aligning their HMMs with HHalign. For protein language model embedding methods, we calculate the predicted similarity as described above (Structural similarity prediction module).

We compare the predicted structural similarity scores against the ground truth scores defined by SCOP across a variety of metrics. Accuracy is the fraction of protein pairs for which the similarity level is predicted exactly correctly. We also calculate the Pearson correlation coefficient (r) and Spearman rank correlation coefficient (ρ) between the predicted and ground truth similarities. Finally, we calculate the average-precision score for retrieving pairs of proteins at or above each level of similarity. That is, we report the average-precision score for each method where the positive set is proteins in the same class, in the same fold, in the same superfamily, or in the same family.

Transmembrane region prediction training and evaluation

We follow the procedure for transmembrane prediction and evaluation previously described by Tsirigos et al. and the model described by Bepler & Berger (Tsirigos et al., 2015a; Bepler and Berger, 2019). The TOPCONS2 dataset contains protein sequences and transmembrane annotations for four categories of proteins: 1) proteins with transmembrane regions (TM), 2) proteins with transmembrane regions and a signal peptide (TM+SP), 3) proteins without transmembrane regions or a signal peptide (globular), and 4) proteins without transmembrane regions but with a signal peptide (globular+SP). Altogether, the dataset contains 5154 proteins broken down into 286 TM, 627 TM+SP, 2927 globular, and 1314 globular+SP proteins.

In order to compare different protein representations for transmembrane prediction, we fit a single layer BiLSTM followed by a conditional random field (CRF) decoder using either 1-hot encodings of the amino acid sequence or embeddings generated by the SSA-LSTM, DLM-LSTM, or MT-LSTM models. The BiLSTM has 150 hidden units in each direction and the CRF decodes the outputs of the BiLSTM to one of four states: signal peptide, cytosolic region, transmembrane region, or extracellular region. In the CRF, we use the hidden state grammar and transitions defined by Tsirigos et al. (Tsirigos et al., 2015b) and only fit the input potentials. The models are trained for 10 epochs over the data with a batch size of 1 using the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 0.0003.

We compare methods by 10-fold cross validation. We calculate prediction performance over proteins in the held-out set by decoding the most likely sequence of labels using the Viterbi algorithm and then scoring a protein as correctly predicted if 1) the protein is globular and we predict no transmembrane or signal peptide regions, 2) the protein is globular+SP and we predict that the protein starts with a signal peptide and has no transmembrane regions, 3) the protein is TM and we predict the correct number of transmembrane regions with at least 50% overlap to the ground truth regions and no signal peptide, and 4) the protein is TM+SP and is the same as TM except that we also predict that the protein starts with a signal peptide.

Sequence-to-phenotype prediction and evaluation

We retrieve the set of deep mutational scanning datasets aggregated by Riesselman et al. (Riesselman, Ingraham, and Marks, 2018) and follow the supervised learning procedure used by Luo et al. (Luo et al., 2020). These datasets contain phenotypic measurements of sequence variants across a variety of proteins and measured phenotypes. Phenotypes include enzyme function (Bandaru et al., 2017; Wrenbeck, Azouz and Whitehead, 2017), growth (Melamed et al., 2013; Kitzman et al., 2015; Brenan et al., 2016; Klesmith et al., 2017; Weile et al., 2017; Findlay et al., 2018), stability (Matreyek et al., 2018), peptide binding (Araya et al., 2012; McLaughlin et al., 2012), ligase activity (Starita et al., 2013), and MIC (Jacquier et al., 2013).

For each dataset, we featurize the amino acid sequences of each variant as either a 1-hot encoding or by embedding the sequence with our MT-LSTM model. We then apply dimensionality reduction to these vectors using PCA down to the minimum of 1000 PCs or the number of data points in the dataset in order to improve the runtime of the learning algorithm. We then fit a Gaussian process (GP) regression model using the RBF kernel and fit the kernel hyperparameters by maximum likelihood. We implement our GP models in GPyTorch (Gardner et al., 2018). To compare methods, we follow Luo et al. and perform 5-fold cross validation on each deep mutational scanning dataset (Luo et al., 2020) and calculate the Spearman rank correlation coefficient between our predicted phenotypes and the ground truth phenotypes on the heldout data for each fold.