Science
AAAS

Supplementary Materials for

Evolutionary-scale prediction of atomic-level protein structure with a language model

Zeming Lin *et al*.

Corresponding author: Alexander Rives, arives@meta.com

Science **379**, 1123 (2023)
DOI: 10.1126/science.ade2574

The PDF file includes:

A Materials and Methods

A.1 Data

A.1.1 Sequence dataset used to train ESM-2

UniRef50, September 2021 version, is used for the training of ESM models. The training dataset was partitioned by randomly selecting 0.5% (≈ 250,000) sequences to form the validation set. The training set has sequences removed via the procedure described in Meier et al. (61). MMseqs search (`-min-seq-id 0.5 -alignment-mode 3 -max-seqs 300 -s 7 -c 0.8 -cov-mode 0`) is run using the train set as query database and the validation set as target database. All train sequences which match a validation sequence with 50% sequence identity under this search are removed from the train set. The 50% identity threshold is chosen because the purpose of the validation set is primarily to detect overfitting (as is common in the machine learning community), rather than to test generalization. Generalization performance is tested through performance on downstream tasks (such as structure prediction on the CASP14 and CAMEO test sets).

De-novo designed proteins are filtered out from the pretraining dataset via two filters. First, any sequence in UniRef50 and UniRef90 that was annotated as "artificial sequence" by a taxonomy search on the UniProt website, when `2021_04` was the most recent release (1,027 proteins), was removed. Second, jackhmmer was used to remove all hits around a manually curated set of 81 de-novo proteins. jackhmmer was run with `−num-iter 1 -max` flags, with each of the 81 de-novo proteins as a query and UniRef100 as a search database. All proteins returned by jackhmmer were removed from both UniRef50 and UniRef90 via their UniRef IDs (58,462 proteins). This filtering is performed to enable future work evaluating the generalization of language models to de-novo sequences.

To increase the amount of data and its diversity, a minibatch of UniRef50 sequences is sampled for each training update. Each sequence is then replaced with a sequence sampled uniformly from the corresponding UniRef90 cluster. This allowed ESM-2 models to train on over 60M protein sequences.

A.1.2 Structure training sets for ESMFold

For training ESMFold, we follow the training procedure outlined in Jumper et al. (12). We find all PDB chains until 2020-05-01 with resolution less than or equal to 9Å and length greater than 20. All proteins where over 20% of the sequence is the same residue is not considered. MMseqs easy-cluster with default parameters is used to cluster resulting sequences at 40% sequence identity. Only individual chains are used during training, even when the chain is part of a protein complex. This results in 25,450 clusters covering a total of 325,498 chains.

At training time, each cluster is sampled evenly, and then a random protein is sampled from each cluster. Rejection sampling is applied to train on longer proteins more frequently, where protein chains are accepted with probability $\frac{1}{512}\max(\min(N_{\text{res}}, 512), 256)$.

As described in Hsu et al. (62), we generated a set of 13,477,259 structure predictions with AlphaFold2 using MSAs generated via the process in Rao et al. (44). The dataset is then filtered to select only sequences with mean pLDDT >70. Because of the way the dataset is constructed, only 1.5% of the dataset is removed with this filter. Additionally, loss is not calculated for residues with pLDDT <70. We found that this is necessary to obtain increased performance using predicted structures. Predicted structures are sampled 75% of the time, and real structures 25% of the time during training. Data processing is done with Biotite (63).

A.1.3 Structure validation and test sets

During method development (e.g. hyperparameter selection), we used a temporally held out validation set obtained from the Continuous Automated Model Evaluation (CAMEO) server (41) by filtering from August 2021 to January 2022.

We report results by testing 3D structure prediction models on two test sets, both chosen to be temporally held out from our supervised training set. The first test set is from CAMEO, consisting of all 194 test proteins from April 01, 2022 through June 25, 2022. Our second test set consists of 51 targets from the CASP14 competition (42). For both test sets, metrics are computed on all modeled residues in the PDB file. The full CASP14 target list is:

T1024, T1025, T1026, T1027, T1028, T1029, T1030, T1031, T1032, T1033, T1034, T1035, T1036s1, T1037, T1038, T1039, T1040, T1041, T1042, T1043, T1044, T1045s1, T1045s2, T1046s1, T1046s2, T1047s1, T1047s2, T1049, T1050, T1053, T1054, T1055, T1056, T1057, T1058, T1064, T1065s1, T1065s2, T1067, T1070, T1073, T1074, T1076, T1078, T1079, T1080, T1082, T1089, T1090, T1091, T1099.

These are the full extent of the publicly available CASP14 targets as of July 2022.

No filtering is performed on these test sets, as ESMFold is able to make predictions on all sequences, including the length-2166 target T1044.

A.1.4 CAMEO Dataset Difficulty Categories

The CAMEO evaluation places each target into three categories: easy, medium, and hard. This placement is done based on the average performance of all public structure prediction servers. Targets are classified as "easy" if the average lDDT is >0.75, "hard" if the average lDDT is < 0.5, and "medium" otherwise. In the main text, we report average performance across all targets in CAMEO. In Table S4 we provide statistics for each difficulty category.

A.2 Language Models

A.2.1 Computing unsupervised contact prediction from language models

We use the methodology of Rao et al. (20) to measure unsupervised learning of tertiary structure in the form of contact maps. A logistic regression is used to identify contacts. The probability of a contact is defined as

$$p(c_{ij}) = \left(1 + \exp\left(-\beta_0 - \sum_{l=1}^{N} \sum_{k=1}^{K} \beta_{kl}\, a_{ij}^{kl}\right)\right)^{-1} \tag{2}$$

where c_{ij} is a boolean random variable which is true if amino acids *i*,*j* are in contact. Suppose our transformer has *N* layers and *K* attention heads per layer. Then A_{kl} is the symmetrized and APC-corrected (64) attention map for the *k*-th attention head in the *l*-th layer of the transformer, and a_{ij}^{kl} is the value of that attention map at position *i*,*j*.

The metric we use, long range P@L, for each protein, is defined as the precision of the top *L* predicted long range contacts by confidence for a protein of length *L*. long range is defined as contacts that are ≥24 residues apart in the protein sequence. This is averaged over each protein that we test over. We also use P@L/5 in some sections of this work, which computes precision over the top *L*/5 predictions instead.

The parameters are fit in scikit-learn (65) using L1-regularized logistic regression with $\lambda = 0.15$. The regression is fit using the same 20 protein training set used in Rao et al. (20), which was simply a random selection from the trRosetta (11) training set. We performed a variability analysis using 20 bootstrapped samples of 20 training proteins from the total set of 14862 proteins. The average long-range P@L was 0.4287 with a standard deviation of 0.0028. We also performed experiments using larger training sets but observed no significant performance change. Given these results, we are confident that selecting a subset of 20 proteins for training provides a good estimate of contact precision performance.

Unsupervised contact prediction results are reported for the 14842-protein test set used in Rao et al. (20), which is also derived from the trRosetta training set, excluding the 20 proteins used in fitting the regression. For both training and test a contact is defined as two amino acids with C-α distance < 8Å.

A.2.2 Language model perplexity calculations

Perplexity is a measure of a language model's fidelity and is defined as the exponential of the negative log-likelihood of the sequence. Unfortunately, there is no efficient method of computing the log-likelihood of a sequence under a masked language model. Instead, there are two methods we can use for estimating perplexity.

First, let the mask *M* be a random variable denoting a set of tokens from input sequence *x*. Each token has a 15% probability of inclusion. If included the tokens have an 80% probability of being replaced with a mask token, a 10% probability of being replaced with a random token, and a 10% probability of being replaced with an unmasked token. Let $\hat{x}_{i\in M}$ denote the set of modified input tokens. The perplexity is then defined as

$$\textsc{Perplexity}(x) = \exp\left\{-\log p\left(x_{i\in M} \middle| x_{j\notin M} \cup \hat{x}_{i\in M}\right)\right\} \tag{3}$$

As the set *M* is a random variable, this expression is non-deterministic. This makes it a poor estimate of the perplexity of a single sequence. However, it requires only a single forward pass of the model to compute, so it is possible to efficiently obtain an estimate of the expectation of this expression over a large dataset. When reporting the perplexity over a large dataset (such as our UniRef validation set), this estimate is used.

The second perplexity calculation is the pseudo-perplexity, which is the exponential of the negative pseudo-log-likelihood of a sequence. This estimate provides a deterministic value for each sequence, but requires L forward passes to compute, where L is the length of the input sequence. It is defined as

$$\textsc{Pseudoperplexity}(x) = \exp\left\{-\frac{1}{L}\sum_{i=1}^{L} \log p\left(x_i \middle| x_{j\neq i}\right)\right\} \tag{4}$$

When reporting the perplexity for an individual sequence (e.g. on CASP14 or CAMEO), this estimate is used. For brevity, we refer to both of these estimates as the "perplexity," as they can be interpreted in a similar manner.

A.2.3 ESM-2 model architecture

We use a BERT (25) style encoder only transformer architecture (23) with modifications. We change the number of layers, number of attention heads, hidden size and feed forward hidden size as we scale the ESM model (Table S3).

The original transformer paper uses absolute sinusoidal positional encoding to inform the model about token positions. These positional encodings are added to the input embeddings at the bottom of the encoder stack. In ESM-1b (14), we replaced this static sinusoidal encoding with a learned one. Both static and learned absolute encodings provide the model a very cheap way of adding positional information. However, absolute positional encoding methods don't extrapolate well beyond the context window they are trained on. In ESM-2, we used Rotary Position Embedding (RoPE) (66) to allow the model to extrapolate beyond the context window it is trained on. RoPE slightly increases the computational cost of the model since it multiplies every query and key vector inside the self-attention with a sinusoidal embedding. In our experiments, we observed that this improves model quality for small models. However, we observed that the performance improvements start to disappear as the model size and training duration get bigger.

A.2.4 Training ESM-2

In ESM-2, we have made multiple small modifications to ESM-1b with the goal of increasing the effective capacity. ESM-1b had dropout both in hidden layers and attention which we removed completely to free up more capacity. In our experiments, we did not observe any significant performance regressions with this change.

We trained most of our models on a network with multiple nodes connected via a network interface. As the models get bigger, the amount of communication becomes the fundamental bottleneck for the training speed. Since BERT style models have been shown to be amenable to very large batch sizes (67), we increased our effective batch size to 2M tokens.

For model training optimization, we used Adam with $\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 10^{-8}$ and L_2 weight decay of 0.01 for all models except the 15 billion parameter model, where we used a weight decay of 0.1. The learning rate is warmed up over the first 2,000 steps to a peak value of 4e-4 (1.6e-4 for the 15B parameter model), and then linearly decayed to one tenth of its peak value over the 90% of training duration. We trained all models for 500K updates except the 15B model which we trained for 270K steps. All models used 2 million tokens as batch size except the 15B model where we used 3.2 million tokens batch size. In order to efficiently process large proteins, we cropped long proteins to random 1024 tokens. We used BOS and EOS tokens to signal the beginning and end of a real protein, to allow the model to separate a full-sized protein from a cropped one.

We used standard distributed data parallelism for models up to 650M parameters and used sharded data parallelism (FSDP) (68) for the 2.8B and 15B parameter models. FSDP shards model weights and optimization parameters across multiple GPUs, allowing us to train models that can't fit into a single GPU memory.

We trained each model over 512 NVIDIA V100 GPUs. ESM2 700M took 8 days to train. The 3B parameter LM took 30 days. The 15B model took 60 days. All language models were trained for 500K updates, except the 15B language model which we stopped after 270K updates due to computational constraints.

A.2.5 ESM-2 ablation experiments

We ran ablation experiments using 150M parameter models trained for 100K steps. Ablations were performed for RoPE, the training dataset (comparing to the ESM-1b training dataset), and UniRef90 sampling (Table S5).

Unsupervised contact prediction results show that both RoPE and newer data significantly improve the results. We do observe a slight regression when sampling from UniRef90 clusters, however we believe this difference is small and the UniRef90 cluster sampling is likely to help for the larger models.

A.2.6 Relationship between change in Perplexity and Contact Accuracy

The relationship between improvements in perplexity and improvements in contact accuracy can be measured via normalized discounted cumulative gain (NDCG). In particular, we hypothesize that large improvements in perplexity correspond with large improvements in contact accuracy. We define the change in perplexity as the difference in language model perplexity for a particular protein sequence between adjacent model sizes. Similarly, we define the change in contact accuracy as the difference in unsupervised contact precision for a particular protein sequence between adjacent model sizes. By ranking proteins according to the change in perplexity, we then compute the NDCG with respect to the change in contact accuracy. The average NDCG across the five model classes is 0.87.

A.3 ESMFold

Algorithm 1 Folding block.

```
procedure FoldingBlock($\boldsymbol{s} \in \boldsymbol{R}^{C_s \times L}, \boldsymbol{z} \in \boldsymbol{R}^{C_z \times L \times L}, \boldsymbol{C_s} = \mathbf{1024}, \boldsymbol{C_z} = \mathbf{128}$)
  $\boldsymbol{b} \leftarrow$ Linear($\boldsymbol{z}$)
  $\boldsymbol{s} \leftarrow \boldsymbol{s} +$ MultiHeadSelfAttention($\boldsymbol{s}$,bias $= \boldsymbol{b}$)
  $\boldsymbol{s} \leftarrow \boldsymbol{s} +$ MLP($\boldsymbol{s}$)
  $\boldsymbol{z} \leftarrow \boldsymbol{z} +$ Linear(Concat([OuterProduct($\boldsymbol{s}$),OuterDifference($\boldsymbol{s}$)]))⊣ )
  $\boldsymbol{z} \leftarrow \boldsymbol{z} +$ TriangularMultiplicativeUpdateOutgoing($\boldsymbol{z}$)
  $\boldsymbol{z} \leftarrow \boldsymbol{z} +$ TriangularMultiplicativeUpdateIncoming($\boldsymbol{z}$)
  $\boldsymbol{z} \leftarrow \boldsymbol{z} +$ TriangularSelfAttentionOutgoing($\boldsymbol{z}$)
  $\boldsymbol{z} \leftarrow \boldsymbol{z} +$ TriangularSelfAttentionIncoming($\boldsymbol{z}$)
  $\boldsymbol{z} \leftarrow \boldsymbol{z} +$ MLP($\boldsymbol{z}$)
 return $s$, $z$
end procedure
```

Algorithm 2 ESMFold with *N* folding blocks. ESM_hiddens returns all hidden representations from an ESM language model. layer_weights contains a trainable weight for each layer of ESM.

```
procedure ESMFold(sequence, $C_{\text{esm}} = 1280, C_s = 1024, C_z = 128, N = 48, L =$ Length )
  ESMFold(sequence)
  $s \leftarrow$ ESM_hiddens(sequence)
  $s \leftarrow$ (softmax(layer_weights) $\times$ $s$).sum(over layers)⊣ )
  $s \leftarrow$ MLP($s$)
  $z \leftarrow$ PairwiseRelativePositionalEncoding($L$)
  for $i \leftarrow 1, \ldots, N$ do
    $s, z \leftarrow$ FoldingBlock$_i$($s$, $z$)
  end for
  return StructureModule($s, z$)
end procedure
```

A.3.1 ESMFold model architecture

The ESMFold model uses a simple architecture that leverages the evolutionary information captured by the language model. The architecture is split into two parts, similarly to AlphaFold2: a folding module which takes the language model features as input and produces representations, and a structure module which takes the output from the folding module and outputs 3d atomic coordinates. For the structure module, we use the equivariant transformer architecture with invariant point attention proposed in AlphaFold2. For the folding block we simplify the Evoformer block used in AlphaFold2. No templates are used in ESMFold.

The major change that needs to be made to adapt the Evoformer block to language model features is to remove its dependence on MSAs. Since MSAs are two dimensional, the Evoformer employs axial attention (69) over the columns and rows of the MSA. The language model features are one dimensional, so we can replace the axial attention with a standard attention over this feature space. The self-attention uses a bias derived from the pairwise representations. The sequence representation communicates with pairwise representation via both an outer product and outer difference. Other operations in the Evoformer block are kept the same. We call this simplified architecture the Folding block, described in detail in Algorithm 1, and shown in Fig. 2a.

Our final architecture, ESMFold, described in Algorithm 2, has 48 folding blocks. It is trained for an initial 125K steps on protein crops of size 256, and then fine-tuned with the structural violation loss for 25K steps, on crop sizes of 384. We use the Frame Aligned Point Error (FAPE) and distogram losses introduced in AlphaFold2, as well as heads for predicting LDDT and the pTM score. We omit the masked language modeling loss. For training AlphaFold2, distance errors in the FAPE loss were clamped to a maximum of 10 angstroms for 90% of batches. We instead calculate both clamped and unclamped losses and take the sum, with weights of 0.9 and 0.1 respectively. Language model parameters are frozen for training ESMFold. We use the 3B parameter ESM-2 language model, the largest model that permits inference on a single GPU.

We use a learned weighted sum of ESM embeddings to produce the initial hidden state into the model. This is then fed through a multi-layer perceptron (MLP). The initial pairwise state is simply the pairwise relative positional encoding described in Jumper et al. (12). We found that using the attention maps initially gives a boost in performance, but this disappears during training. For experiments that do not use any folding blocks, we use an MLP applied to the ESM attention maps as input and add the pairwise relative positional encoding to the attention map scores. Finally, the StructureModule projects these representations into coordinates.

The predicted LDDT head is output from the hidden representation of the StructureModule. The predicted TM head uses the pairwise representation z. Finally, we also predict the distogram, from the same representation. All output heads are linear transformations with no nonlinearities.

A.3.2 Masked prediction

It is possible to sample alternate predictions from ESMFold by masking inputs to the language model. We test this procedure with the following protocol: Input 1000 different sequences into ESMFold with different masking patterns in the language model. The masking patterns are uniformly sampled, where 0 to 15% of the sequence is masked out. A prediction is made for each masked sequence, and the sequence with highest pLDDT is chosen as the final model prediction. On average, applying this procedure only results in a 0.021 LDDT increase on CAMEO, but on some PDBs can substantially improve the accuracy, e.g. for PDB 6s44, TM-score improves from 0.81 to 0.94 (Fig. S6).

A.3.3 Extracting coordinates from ESM-2

The following methodology is used to project out coordinates from the language model representations (Fig. 1, Table S1). We train an equivariant structure module directly on top of the frozen ESM representations using a dataset of experimentally determined structures. The training set is the same as used for ESMFold, and we use the same losses and architecture as the AlphaFold2 structure module. We initialize the pairwise representation of the structure module with the output of an MLP that processes the attention maps of the language model. Note that we do not use the predicted structures dataset as data augmentation in these experiments; we train the projection only with experimentally determined structures.

As language models grow in size, we find a large increase in LDDT, from 0.48 on the 8M parameter LM to 0.72 on the 15B parameter LM. This demonstrates that a simple head on top of a powerful language model already gives reasonably accurate structure predictions.

A.3.4 Timing analysis

We evaluate the speed of the model by testing sequences of varying length on a single NVIDIA V100 GPU. ESMFold makes a prediction on a protein with 384 residues in 14.2 seconds, 6x faster than a single AlphaFold2 model. On shorter sequences we see a 60x improvement (Fig. S2). Note that this excludes the CPU time for MSA and template search, as well as the 5x from the default ensemble of models used by AlphaFold2. ESMFold can be run reasonably quickly on CPU, and an Apple M1 Macbook Pro makes the same prediction in just over 5 minutes.

ESMFold provides multiple options for reducing GPU memory utilization including chunked attention, mixed precision, and CPU offloading, some of which come at the cost of inference speed. Combined, the optimizations allow predictions on long sequences (such as length-2166 CASP14 target T1044) on an NVIDIA V100 GPU.

A.4 Metagenomic predictions

A.4.1 Folding 617 million sequences from MGnify

We obtained MGnify (32) version 2022 at 90% sequence similarity (MGnify90). We built a fault tolerant distributed system with a main node which, via TCP, communicates sequences to many workers and receives results as folded protein structures. We were able to leverage the resources of a heterogeneous GPU cluster consisting of P100s, V100s, and A100s of various configurations. We estimate that on a homogeneous network GPU cluster of V100s, the entire 620 million sequences would take approximately 28,000 GPU days to fold, which we were able to complete in 2 weeks. We obtained structure predictions and corresponding pLDDT values for each of these sequences.

A.4.2 Analysis of folded metagenomics structures

On a random sample of 1M high confidence structures, we used Foldseek search (version 3.915ef7d) (53) to perform an all-by-all structural similarity search against the PDB (as of April 12, 2022) based on TM-score. We use foldseek with default parameters, except increasing the E-value to 1.0 from the default 1e-3 (foldseek search -e 1.0), to increase recall. We also used MMseqs2 search (version 13.45111) to perform an all-by-all sequence similarity search against UniRef90. We use MMseqs2 with default parameters, except that we re-ran MMseqs2 with the most sensitive setting (-s 7.0) for any sequences that returned an empty result, to increase the recall.

As mentioned in the main text, for 3.4% (33,521 proteins) of the Atlas 1M high confidence subsample, no significant match is found in UniRef90 with MMseqs2. For reference, a random subsample of MGnify90, without confidence threshold, has 26.4% (264,075 proteins) without hits in UniRef90. Predictions that fall below the high confidence threshold can still be useful as they may have regions of well-predicted structure.

To visualize this landscape of 1M MGnify sequences, we first used ESM-1b to embed each sequence as a 1280-dimensional vector. These embeddings were then visualized using the umap version 0.5.3, scanpy version 1.9.1, and anndata 0.8.0 Python packages (70-72), where dimensionality reduction was applied directly to the embedding vectors (use_rep=‘X’ in scanpy.tl.umap) with default parameters (15-nearest-neighbors graph via approximate Euclidean distance, UMAP min_dist=0.5).

We further analyzed a random subsample of very high-confidence structures with mean pLDDT greater than 0.9, corresponding to ∼59K structures. For each of these structures, we used Foldseek easy-search (`–alignment-type 1`) to identify similar structures in the PDB. To assess the quality of structure predictions with no Foldseek matches, we used full AlphaFold2 with MSAs to also obtain structure predictions, where we picked the top of five relaxed models ranked by mean pLDDT. We then computed RMSD values of aligned backbone coordinates and all-atom TM-score between the ESMFold-predicted and AlphaFold2-predicted structures and found good agreement of the predictions between both methods (Fig. S7).

To select our case studies, we then used blastp version 2.10.0+ (60) to search for similar sequences in UniRef90 to compute sequence identity. For case-study sequences with no significant matches in UniRef90, we also used the jackhmmer web server (https://www.ebi.ac.uk/Tools/hmmer/search/jackhmmer) (54) to manually query four reference proteomes for similar sequences. Highlighted structure predictions with low similarity to known structures were manually selected and are summarized in Fig. 4. For these structures, we also performed an additional structural similarity search using the Foldseek webserver (https://search.foldseek.com/search) with default parameters to identify the closest structures in PDB100 211201 beyond the TM-score cutoff of 0.5.

A.5 Multimer Benchmark

A.5.1 Recent-PDB-Multimers

To evaluate ESMFold on protein complexes, we construct an evaluation set using the methods described in Evans et al. (49). This dataset consists of targets deposited in the Protein Data Bank between 2020-05-01 and 2022-06-01. The following filtering steps are performed:

- Complexes must contain more than 1 chain and less than 9 chains.
- Chains with length <20 residues, or where one residue makes up >20% of the chain are excluded.
- Complexes must contain fewer than 1536 residues, excluding chains which fail the previous step.
- Each chain is assigned to a 40% overlap cluster using clusters provided by the PDB
- Each complex is assigned a cluster which is the union of chain cluster ids
- From each cluster complex, the example with highest resolution is selected as the representative

These steps result in a total of 2978 clusters. Predictions are made on the full complex, but metrics are computed on a per chain-pair basis using the DockQ program (50). Chain pairs are greedily selected for evaluation if their pair cluster id has not been previously evaluated. Chain pairs which DockQ identifies as having no contacting residues in the ground truth are not evaluated. This results in a total of 3505 unique chain pairs.

A.5.2 Multimer Predictions

To predict complexes in the benchmark shown in Fig. 2d and S4, we give a residue index break of 1000 to ESMFold and link chains with a 25-residue poly-glycine linker, which we remove before displaying. Note that this is using ESMFold out of distribution since single chains are used during training.

A.6 Orphan Proteins

Orphan proteins are sequences with few to no evolutionary homologs in either structure or sequence databases. Due to a lack of evolutionary information, these sequences can be very challenging for current structure prediction models. To evaluate ESMFold on orphan proteins, we construct an orphan protein dataset using the following procedure:

- Select structures deposited in the PDB from 2020-05-01 to 2022-05-01 with resolution greater than 9Å. and at least 20 modeled residues.
- Cluster at a 70% sequence identity threshold with MMseqs, and select the cluster representatives.
- Run hhblits for 1 iteration (all other parameters default) against UniRef (2020_06), select sequences with no hits.
- Run the standard AlphaFold2 MSA generation pipeline against UniRef, MGnify, and BFD, selecting sequences with <100 total sequence hits and no template hits with TM-score >0.5.

Fig. S8 shows results at different MSA depth thresholds. After filtering, there are 104 sequences with MSA depth ≤100, 70 sequences with MSA depth ≤10, and 22 sequences with MSA depth = 1. Beyond the constraint that no template has TM-score > 0.5, no filtering on the number of templates is performed.

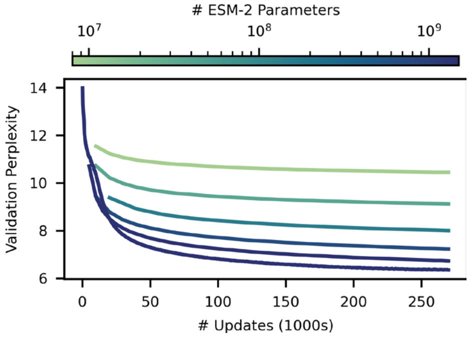


Figure S1:

ESM-2 masked language modeling training curves. Training curves for ESM-2 models from 8M (highest curve, light) to 15B parameters (lowest curve, dark). Models are trained to 270K updates. Validation perplexity is measured on a 0.5% random-split holdout of UniRef50. After 270K updates the 8M parameter model has a perplexity of 10.45, and the 15B model reaches a perplexity of 6.37.

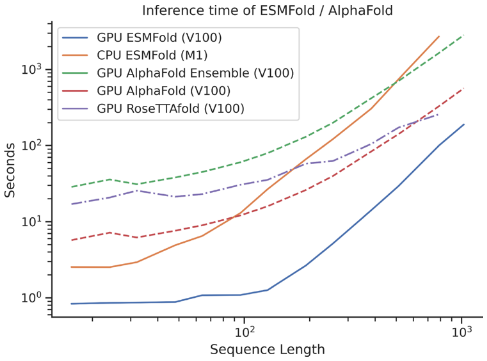


Figure S2:

ESMFold timing. Comparison to AlphaFold2 and RoseTTAfold. We test the speed of ESMFold on sequence lengths up to 1024. Note that this comparison is only on the network forward time, and does not include the cost of the search to generate MSAs. ESMFold performance at low sequence lengths is dominated by the forward pass of the language model. At high sequence lengths the $O(n^3)$ computation of pairwise representations takes over. Most of ESMFold's speed advantage comes from not needing to process the MSA branch. We see an over 60x speed advantage for shorter protein sequences, and a reasonable speed advantage for longer protein sequences. We do not count Jax graph compilation times or MSA search times for AlphaFold2, meaning in practice there is a larger performance difference in the cold start case. We also use an optimized Colabfold 1.3.0 (30) to do speed comparison. No significant optimization has been performed on ESMFold, and we suspect that further gains can be made by optimizing ESMFold as well. For RoseTTAFold, the speed of the SE(3) Transformer (73) dominates, especially at low sequence lengths. The number of SE(3) max-iterations are artificially limited to 20 (default 200) and no MSAs are used as input for these measurements. For RoseTTAfold predictions we do not include the cost of computing sidechains with PyRosetta.

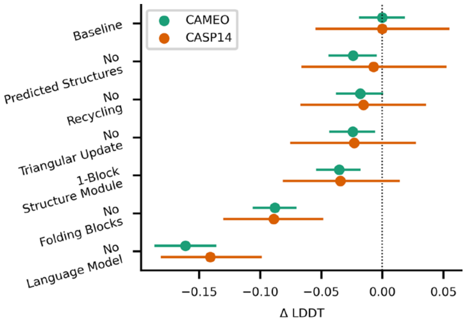


Figure S3:

ESMFold ablations on CAMEO and CASP14. ESMFold ablations on CAMEO and CASP14 test sets show the largest contributing factors to performance are the language model and the use of folding blocks. Other ablations reduce performance on CASP14 and CAMEO by 0.01-0.04 LDDT.

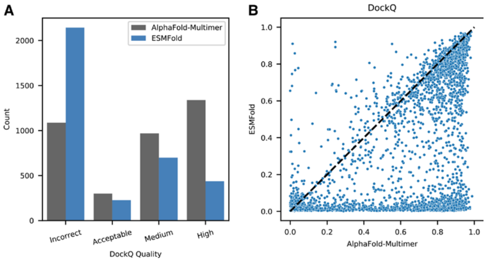


Figure S4:

Comparison of ESMFold and AlphaFold-Multimer on recent-PDB-multimers dataset. DockQ (50) scores for AlphaFold-Multimer and ESMFold predictions for chain pairs in the Recent-PDB-Multimers dataset. DockQ qualitative categorizations (left) and quantitative comparison (right) are provided for all chain pairs. ColabFold (30) was used to generate paired MSAs for each complex using the ‘paired+unpaired‘ MSA generation setting. UniRef, environmental, and template databases were used. ESMFold predictions are in the same qualitative DockQ categorization for 53.2% of complexes, even though ESMFold is not trained on protein complexes. Dataset generation and scoring methodology described in Appendix A.5.1.

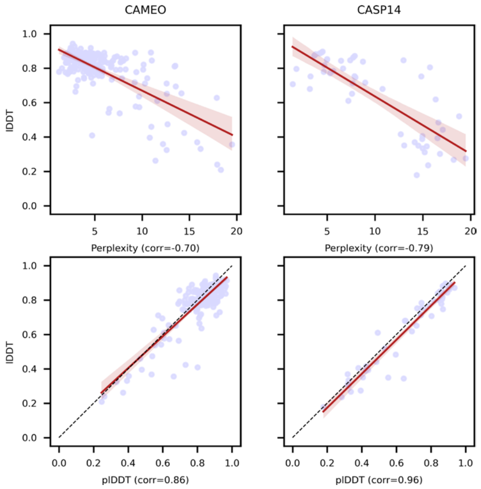
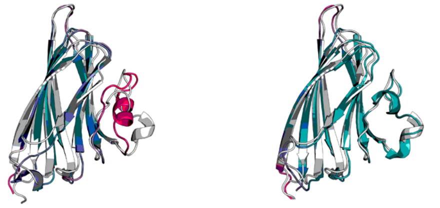


Figure S5:

ESMFold calibration with respect to perplexity and pLDDT on CASP14 and CAMEO. Language model perplexity and ESMFold pLDDT are both well correlated with actual structure prediction accuracy on CASP14 and CAMEO. Well understood sequences with language model perplexity <6 are usually well predicted by ESMFold. The strong correlation between pLDDT and LDDT suggests filtering predictions by pLDDT will mostly capture well predicted structures.



6s44, CA Sequence
pLDDT=72.6, TM-score=0.81

Highest pLDDT prediction after 1K masked ensemble
pLDDT=82.6, TM-score=0.94

Figure S6:

Masked prediction on Cα sequence of PDB 6s44. Left: ESMFold prediction (TM-score=0.81) on the Cα sequence of PDB 6s44. Right: Best prediction out of 1000 masked sequences generated via the procedure described in Appendix A.3.2. Prediction with highest pLDDT is shown and has improved TM-score (TM-score=0.94).

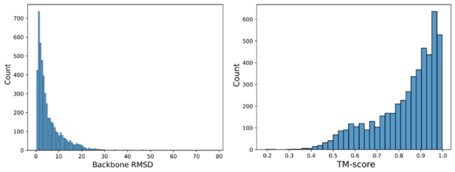


Figure S7:

Comparison to AlphaFold2 of structurally remote ESMFold predictions. Distributions of backbone RMSDs (left) and TM-scores (right) of ESMFold-AlphaFold2 predictions of the same sequence, where the ESMFold prediction has both high confidence (mean pLDDT >0.9) and low structural similarity to the PDB (Foldseek closest PDB TM-score <0.5).

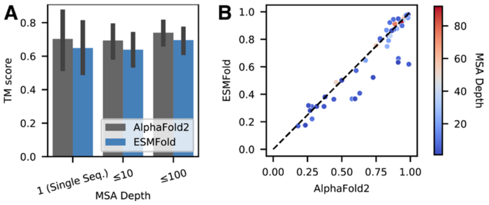


Figure S8:

Comparison of ESMFold and AlphaFold2 on a set of orphan proteins. Performance of ESMFold and AlphaFold2 on a set of "orphan proteins" - sequences with few sequence or structural homologs. All compared sequences are temporally held out from the training set. The standard AlphaFold2 sequence and template search pipeline is used to find homologs (dataset construction described in Appendix A.6). (A) Comparison on natural proteins with various MSA depths. Depth is the total number of hits across UniRef and metagenomic databases. (B) TM-score comparison of all individual orphans.

Model	# Params	# Updates	Validation Perplexity	LR P@L	LR P@L/5	CASP14	CAMEO
ESM-2	8M	270K	10.45	0.16	0.28	0.37	0.48
	35M	270K	9.12	0.29	0.49	0.41	0.56
	150M	270K	8.00	0.42	0.68	0.47	0.63
	650M	270K	7.23	0.50	0.77	0.51	0.68
	3B	270K	6.73	0.53	0.80	0.51	0.71
	8M	500K	10.33	0.17	0.29	0.37	0.48
	35M	500K	8.95	0.30	0.51	0.41	0.56
	150M	500K	7.75	0.44	0.70	0.49	0.65
	650M	500K	6.95	0.52	0.79	0.51	0.70
	3B	500K	6.49	**0.54**	0.81	0.52	**0.72**
	15B	270K	**6.37**	**0.54**	**0.82**	**0.55**	**0.72**
ESM-1b	650M	—	—	0.41	0.66	0.42	0.64
Prot-T5-XL (UR50) (18)	3B	—	—	0.48	0.72	0.50	0.69
Prot-T5-XL (BFD) (18)	3B	—	—	0.36	0.58	0.46	0.63
CARP (74)	640M	—	—	—	—	0.42	0.59

Table S1:

Detailed language model comparison. Comparison at different numbers of parameters and at different numbers of training updates. Training updates and validation perplexity are not reported for baseline models, since there is no straightforward comparison. For the number of training updates, different models use different batch sizes, so the number of sequences seen can vary even if the number of updates are the same. For validation perplexity, baseline models are not trained on the same dataset, and do not share a common heldout validation set with ESM-2. Prot-T5 is an encoder-decoder language model. Only the encoder portion of the model was used in this evaluation, however the number of parameters reported is the total number of parameters used for training. Unsupervised contact precision results, in the form of long range precision at L and at L / 5, do allow us to compare all transformer language models despite variance in training data. However, CARP, a convolution-based language model, does not have attention maps. Note: ESM-1b is evaluated only on sequences of length <1024, due to constraints with position embedding.

MGnify ID	**Mean pLDDT**	**ESM-2 (3B) Perplexity**	**Foldseek server closest TM-score**	**Foldseek server closest PDB**	**Closest blastp sequence identity (UniRef90)**	**Closest blastp sequence (UniRef90)**
MGYP000712274586	0.96	3.4	0.45	1ttg_A	54%	UniRef90_A0A539E457 Uncharacterized protein (Acidimicrobiaceae bacterium)
MGYP000911143359	0.90	6.5	0.67	5nni_A	43%	UniRef90_A0A7Y5V7P8 Uncharacterized protein (Flavobacteriales bacterium)
MGYP001220175542	0.94	4.2	0.38	5y1x_A	98%	UniRef90_UPI0013011942 Helix-turn-helix domain-containing protein (Caenibacillus caldisaponilyticus)
MGYP001812528822	0.93	4.4	0.39	5hh3_C	50%	UniRef90_A0A545U581 Fatty acid desaturase (Exilibacterium tricleocarpae)
MGYP000706186022	0.92	11.5	0.47	1xks_A	29%	UniRef90_A0A6N6S1Z1 Uncharacterized protein (Candidatus brocadia)
MGYP000279975524	0.93	5.9	0.49	4l5s_B	38%	UniRef90_A0A1F4EWL6 Uncharacterized protein (Betaproteobacteria bacterium)
MGYP004000959047	0.90	7.8	0.80	6bym_A	No significant matches	NA
MGYP000936678158	0.95	10.6	0.68	5yet_B	No significant matches	NA

Table S2.

Information on highlighted MGnify proteins. MGnify sequence identifiers corresponding to predicted structures highlighted throughout this study, including the PDB chain and corresponding TM-score of the closest structure identified by the Foldseek webserver as well as the UniRef90 entry and sequence identity of the closest sequence identified by blastp (Appendix A.4.2).

	8M	**35M**	**150M**	**650M**	**3B**	**15B**
Dataset	UR50/D	UR50/D	UR50/D	UR50/D	UR50/D	UR50/D
Number of layers	6	12	30	33	36	48
Embedding dim	320	480	640	1280	2560	5120
Attention heads	20	20	20	20	40	40
Training steps	500K	500K	500K	500K	500K	270K
Learning rate	4e-4	4e-4	4e-4	4e-4	4e-4	1.6e-4
Weight decay	0.01	0.01	0.01	0.01	0.01	0.1
Clip norm	0	0	0	0	1.0	1.0
Distributed backend	DDP	DDP	DDP	DDP	FSDP	FSDP

Table S3.

ESM-2 model parameters at different scales.

Dataset	Split	Count	MSA Depth (Total)	MSA Depth (UniRef)	ESMFold	AlphaFold2	RoseTTAFold
CAMEO	easy	97	21,458	17,627	0.90	0.93	0.89
	medium	89	3,032	860	0.79	0.86	0.76
	hard	8	328.5	56	0.45	0.62	0.49
CASP14	—	51	1228	161	0.68	0.85	0.81

Table S4.

CAMEO dataset statistics broken down by difficulty class. Median MSA depth is reported for each difficulty class of the CAMEO dataset, along with mean TM-score for ESMFold, AlphaFold, and RoseTTAFold. Half of the samples from the CAMEO dataset consist of “easy” examples, which are well predicted by all models. Differentiation is greater in the “medium” and “hard” classes, which have lower MSA depth and are better predicted by AlphaFold2. Statistics for CASP14 are provided as a comparison. MSA depth numbers provided are from the AlphaFold2 MSA generation pipeline.

	LR P@L	LR P@L/5	Validation Perplexity
Baseline	0.381	0.626	8.42
No RoPE	0.365	0.599	8.62
Older UniRef Data	0.368	0.599	7.98
No UR90 Sampling	0.387	0.631	8.40

Table S5.

ESM-2 architecture ablations.

References and Notes

1. C. Yanofsky, V. Horn, D. Thorpe, Protein Structure Relationships Revealed By Mutational Analysis. *Science* **146**, 1593–1594 (1964). doi:10.1126/science.146.3651.1593 Medline

2. D. Altschuh, T. Vernet, P. Berti, D. Moras, K. Nagai, Coordinated amino acid changes in homologous protein families. *Protein Eng.* **2**, 193–199 (1988). doi:10.1093/protein/2.3.193 Medline

3. U. Göbel, C. Sander, R. Schneider, A. Valencia, Correlated mutations and residue contacts in proteins. *Proteins* **18**, 309–317 (1994). doi:10.1002/prot.340180402 Medline

4. A. S. Lapedes, B. G. Giraud, L. Liu, G. D. Stormo, Correlated Mutations in Models of Protein Sequences: Phylogenetic and Structural Effects. *Lect. Notes Monogr. Ser.* **33**, 236–256 (1999). doi:10.1214/lnms/1215455556

5. J. Thomas, N. Ramakrishnan, C. Bailey-Kellogg, Graphical models of residue coupling in protein families. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **5**, 183–197 (2008). doi:10.1109/TCBB.2007.70225 Medline

6. M. Weigt, R. A. White, H. Szurmant, J. A. Hoch, T. Hwa, Identification of direct residue contacts in protein-protein interaction by message passing. *Proc. Natl. Acad. Sci. U.S.A.* **106**, 67–72 (2009). doi:10.1073/pnas.0805923106 Medline

7. F. Morcos, A. Pagnani, B. Lunt, A. Bertolino, D. S. Marks, C. Sander, R. Zecchina, J. N. Onuchic, T. Hwa, M. Weigt, Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proc. Natl. Acad. Sci. U.S.A.* **108**, E1293–E1301 (2011). doi:10.1073/pnas.1111471108 Medline

8. S. Wang, S. Sun, Z. Li, R. Zhang, J. Xu, Accurate De Novo Prediction of Protein Contact Map by Ultra-Deep Learning Model. *PLOS Comput. Biol.* **13**, e1005324 (2017). doi:10.1371/journal.pcbi.1005324 Medline

9. Y. Liu, P. Palmedo, Q. Ye, B. Berger, J. Peng, Enhancing Evolutionary Couplings with Deep Convolutional Neural Networks. *Cell Syst.* **6**, 65–74.e3 (2018). doi:10.1016/j.cels.2017.11.014 Medline

10. A. W. Senior, R. Evans, J. Jumper, J. Kirkpatrick, L. Sifre, T. Green, C. Qin, A. Žídek, A. W. R. Nelson, A. Bridgland, H. Penedones, S. Petersen, K. Simonyan, S. Crossan, P. Kohli, D. T. Jones, D. Silver, K. Kavukcuoglu, D. Hassabis, Improved protein structure prediction using potentials from deep learning. *Nature* **577**, 706–710 (2020). doi:10.1038/s41586-019-1923-7 Medline

11. J. Yang, I. Anishchenko, H. Park, Z. Peng, S. Ovchinnikov, D. Baker, Improved protein structure prediction using predicted interresidue orientations. *Proc. Natl. Acad. Sci. U.S.A.* **117**, 1496–1503 (2020). doi:10.1073/pnas.1914677117 Medline

12. J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, D. Hassabis, Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583–589 (2021). doi:10.1038/s41586-021-03819-2 Medline

13. M. Baek, F. DiMaio, I. Anishchenko, J. Dauparas, S. Ovchinnikov, G. R. Lee, J. Wang, Q. Cong, L. N. Kinch, R. D. Schaeffer, C. Millán, H. Park, C. Adams, C. R. Glassman, A. DeGiovanni, J. H. Pereira, A. V. Rodrigues, A. A. van Dijk, A. C. Ebrecht, D. J. Opperman, T. Sagmeister, C. Buhlheller, T. Pavkov-Keller, M. K. Rathinaswamy, U. Dalwadi, C. K. Yip, J. E. Burke, K. C. Garcia, N. V. Grishin, P. D. Adams, R. J. Read, D. Baker, Accurate prediction of protein structures and interactions using a three-track neural network. *Science* **373**, 871–876 (2021). doi:10.1126/science.abj8754 Medline

14. A. Rives, J. Meier, T. Sercu, S. Goyal, Z. Lin, J. Liu, D. Guo, M. Ott, C. L. Zitnick, J. Ma, R. Fergus, Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl. Acad. Sci. U.S.A.* **118**, e2016239118 (2021). doi:10.1073/pnas.2016239118 Medline

15. T. Bepler, B. Berger, Learning the protein language: Evolution, structure, and function. *Cell Syst.* **12**, 654–669.e3 (2021). doi:10.1016/j.cels.2021.05.017 Medline

16. E. C. Alley, G. Khimulya, S. Biswas, M. AlQuraishi, G. M. Church, Unified rational protein engineering with sequence-based deep representation learning. *Nat. Methods* **16**, 1315–1322 (2019). doi:10.1038/s41592-019-0598-1 Medline

17. M. Heinzinger, A. Elnaggar, Y. Wang, C. Dallago, D. Nechaev, F. Matthes, B. Rost, Modeling aspects of the language of life through transfer-learning protein sequences. *BMC Bioinformatics* **20**, 723 (2019). doi:10.1186/s12859-019-3220-8 Medline

18. A. Elnaggar, M. Heinzinger, C. Dallago, G. Rehawi, Y. Wang, L. Jones, T. Gibbs, T. Feher, C. Angerer, M. Steinegger, D. Bhowmik, B. Rost, ProtTrans: Towards Cracking the Language of Lifes Code Through Self-Supervised Deep Learning and High Performance Computing. *IEEE Trans. Pattern Anal. Mach. Intell.* **14**, 1 (2021). doi:10.1109/TPAMI.2021.3095381

19. J. Vig, A. Madani, L. R. Varshney, C. Xiong, R. Socher, N. F. Rajani, BERTology Meets Biology: Interpreting Attention in Protein Language Models. arXiv:2006.15222 [cs, q-bio] (2021).

20. R. Rao, J. Meier, T. Sercu, S. Ovchinnikov, A. Rives, Transformer protein language models are unsupervised structure learners. bioRxiv 422761 [Preprint] (2021); https://doi.org/10.1101/2020.12.15.422761.

21. R. Chowdhury, N. Bouatta, S. Biswas, C. Floristean, A. Kharkar, K. Roy, C. Rochereau, G. Ahdritz, J. Zhang, G. M. Church, P. K. Sorger, M. AlQuraishi, Single-sequence protein structure prediction using a language model and deep learning. *Nat. Biotechnol.* **40**, 1617–1623 (2022). doi:10.1038/s41587-022-01432-w Medline

22. C. E. Shannon, Prediction and entropy of printed English. *Bell Syst. Tech. J.* **30**, 50–64 (1951). doi:10.1002/j.1538-7305.1951.tb01366.x

23. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, “Attention is All you Need” in *Advances in Neural Information Processing Systems* (Curran Associates, 2017), pp. 5998–6008.

24. A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving language understanding by generative pre-training (2018).

25. J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (Association for Computational Linguistics, 2019), pp. 4171–4186.

26. T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, "Language Models are Few-Shot Learners" in *Advances in Neural Information Processing Systems* (Curran Associates, 2020), pp. 1877–1901.

27. J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, Q. V. Le, Finetuned Language Models Are Zero-Shot Learners. arXiv:2109.01652 [cs.CL] (2021).

28. J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, D. Zhou, Chain of Thought Prompting Elicits Reasoning in Large Language Models. arXiv:2201.11903 [cs] (2022).

29. A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. Garcia, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A. M. Dai, T. S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozv, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Diaz, O. Firat, M. Catasta, J. Wei, K. Meier-Hellstern, D. Eck, J. Dean, S. Petrov, N. Fiedel, PaLM: Scaling Language Modeling with Pathways. arXiv:2204.02311 [cs] (2022).

30. M. Mirdita, K. Schütze, Y. Moriwaki, L. Heo, S. Ovchinnikov, M. Steinegger, ColabFold: Making protein folding accessible to all. *Nat. Methods* **19**, 679–682 (2022). doi:10.1038/s41592-022-01488-1 Medline

31. M. Steinegger, M. Mirdita, J. Söding, Protein-level assembly increases protein sequence recovery from metagenomic samples manyfold. *Nat. Methods* **16**, 603–606 (2019). doi:10.1038/s41592-019-0437-4 Medline

32. A. L. Mitchell, A. Almeida, M. Beracochea, M. Boland, J. Burgin, G. Cochrane, M. R. Crusoe, V. Kale, S. C. Potter, L. J. Richardson, E. Sakharova, M. Scheremetjew, A. Korobeynikov, A. Shlemov, O. Kunyavskaya, A. Lapidus, R. D. Finn, MGnify: The microbiome analysis resource in 2020. *Nucleic Acids Res.* **48**, D570–D578 (2020). doi:10.1093/nar/gkz1035 Medline

33. S. Mukherjee, D. Stamatis, J. Bertsch, G. Ovchinnikova, J. C. Sundaramurthi, J. Lee, M. Kandimalla, I. A. Chen, N. C. Kyrpides, T. B. K. Reddy, Genomes OnLine Database (GOLD) v.8: Overview and updates. *Nucleic Acids Res.* **49** (D1), D723–D733 (2021). doi:10.1093/nar/gkaa983 Medline

34. K. Tunyasuvunakool, J. Adler, Z. Wu, T. Green, M. Zielinski, A. Žídek, A. Bridgland, A. Cowie, C. Meyer, A. Laydon, S. Velankar, G. J. Kleywegt, A. Bateman, R. Evans, A. Pritzel, M. Figurnov, O. Ronneberger, R. Bates, S. A. A. Kohl, A. Potapenko, A. J.

Ballard, B. Romera-Paredes, S. Nikolov, R. Jain, E. Clancy, D. Reiman, S. Petersen, A. W. Senior, K. Kavukcuoglu, E. Birney, P. Kohli, J. Jumper, D. Hassabis, Highly accurate protein structure prediction for the human proteome. *Nature* **596**, 590–596 (2021). doi:10.1038/s41586-021-03828-1 Medline

35. M. Varadi, S. Anyango, M. Deshpande, S. Nair, C. Natassia, G. Yordanova, D. Yuan, O. Stroe, G. Wood, A. Laydon, A. Žídek, T. Green, K. Tunyasuvunakool, S. Petersen, J. Jumper, E. Clancy, R. Green, A. Vora, M. Lutfi, M. Figurnov, A. Cowie, N. Hobbs, P. Kohli, G. Kleywegt, E. Birney, D. Hassabis, S. Velankar, AlphaFold Protein Structure Database: Massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic Acids Res.* **50**, D439–D444 (2022). doi:10.1093/nar/gkab1061 Medline

36. O. Shimomura, F. H. Johnson, Y. Saiga, Extraction, purification and properties of aequorin, a bioluminescent protein from the luminous hydromedusan, Aequorea. *J. Cell. Comp. Physiol.* **59**, 223–239 (1962). doi:10.1002/jcp.1030590302 Medline

37. K. Mullis, F. Faloona, S. Scharf, R. Saiki, G. Horn, H. Erlich, Specific enzymatic amplification of DNA in vitro: The polymerase chain reaction. *Cold Spring Harb. Symp. Quant. Biol.* **51**, 263–273 (1986). doi:10.1101/SQB.1986.051.01.032 Medline

38. M. Jinek, K. Chylinski, I. Fonfara, M. Hauer, J. A. Doudna, E. Charpentier, A programmable dual-RNA-guided DNA endonuclease in adaptive bacterial immunity. *Science* **337**, 816–821 (2012). doi:10.1126/science.1225829 Medline

39. B. E. Suzek, Y. Wang, H. Huang, P. B. McGarvey, C. H. Wu; UniProt Consortium, UniRef clusters: A comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics* **31**, 926–932 (2015). doi:10.1093/bioinformatics/btu739 Medline

40. S. K. Burley, H. M. Berman, C. Bhikadiya, C. Bi, L. Chen, L. Di Costanzo, C. Christie, K. Dalenberg, J. M. Duarte, S. Dutta, Z. Feng, S. Ghosh, D. S. Goodsell, R. K. Green, V. Guranović, D. Guzenko, B. P. Hudson, T. Kalro, Y. Liang, R. Lowe, H. Namkoong, E. Peisach, I. Periskova, A. Prlić, C. Randle, A. Rose, P. Rose, R. Sala, M. Sekharan, C. Shao, L. Tan, Y.-P. Tao, Y. Valasatava, M. Voigt, J. Westbrook, J. Woo, H. Yang, J. Young, M. Zhuravleva, C. Zardecki, RCSB Protein Data Bank: Biological macromolecular structures enabling research and education in fundamental biology, biomedicine, biotechnology and energy. *Nucleic Acids Res.* **47** (D1), D464–D474 (2019). doi:10.1093/nar/gky1004 Medline

41. J. Haas, A. Barbato, D. Behringer, G. Studer, S. Roth, M. Bertoni, K. Mostaguir, R. Gumienny, T. Schwede, Continuous Automated Model EvaluatiOn (CAMEO) complementing the critical assessment of structure prediction in CASP12. *Proteins* **86** (Suppl 1), 387–398 (2018). doi:10.1002/prot.25431 Medline

42. A. Kryshtafovych, T. Schwede, M. Topf, K. Fidelis, J. Moult, Critical assessment of methods of protein structure prediction (CASP)-Round XIV. *Proteins* **89**, 1607–1617 (2021). doi:10.1002/prot.26237 Medline

43. Y. Zhang, J. Skolnick, Scoring function for automated assessment of protein structure template quality. *Proteins* **57**, 702–710 (2004). doi:10.1002/prot.20264 Medline

44. R. M. Rao, J. Liu, R. Verkuil, J. Meier, J. Canny, P. Abbeel, T. Sercu, A. Rives, “MSA Transformer” in *Proceedings of the 38th International Conference on Machine Learning* (PMLR, 2021), pp. 8844–8856.

45. G. Ahdritz, N. Bouatta, S. Kadyan, Q. Xia, W. Gerecke, T. J. O’Donnell, D. Berenberg, I. Fisk, N. Zanichelli, B. Zhang, A. Nowaczynski, B. Wang, M. M. Stepniewska-Dziubinska, S. Zhang, A. Ojewole, M. E. Guney, S. Biderman, A. M. Watkins, S. Ra, P. R. Lorenzo, L. Nivon, B. Weitzner, Y.-E. A. Ban, P. K. Sorger, E. Mostaque, Z. Zhang, R. Bonneau, M. AlQuraishi, Openfold: Retraining alphafold2 yields new insights into its learning mechanisms and capacity for generalization. bioRxiv 517210 [Preprint] (2022). https://doi.org/10.1101/2022.11.20.517210.

46. J. Dauparas, I. Anishchenko, N. Bennett, H. Bai, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, A. Courbet, R. J. de Haas, N. Bethel, P. J. Y. Leung, T. F. Huddy, S. Pellock, D. Tischer, F. Chan, B. Koepnick, H. Nguyen, A. Kang, B. Sankaran, A. K. Bera, N. P. King, D. Baker, Robust deep learning-based protein sequence design using ProteinMPNN. *Science* **378**, 49–56 (2022). doi:10.1126/science.add2187 Medline

47. J. Wang, S. Lisanza, D. Juergens, D. Tischer, J. L. Watson, K. M. Castro, R. Ragotte, A. Saragovi, L. F. Milles, M. Baek, I. Anishchenko, W. Yang, D. R. Hicks, M. Expòsit, T. Schlichthaerle, J.-H. Chun, J. Dauparas, N. Bennett, B. I. M. Wicky, A. Muenks, F. DiMaio, B. Correia, S. Ovchinnikov, D. Baker, Scaffolding protein functional sites using deep learning. *Science* **377**, 387–394 (2022). doi:10.1126/science.abn2100 Medline

48. B. I. M. Wicky, L. F. Milles, A. Courbet, R. J. Ragotte, J. Dauparas, E. Kinfu, S. Tipps, R. D. Kibler, M. Baek, F. DiMaio, X. Li, L. Carter, A. Kang, H. Nguyen, A. K. Bera, D. Baker, Hallucinating symmetric protein assemblies. *Science* **378**, 56–61 (2022). doi:10.1126/science.add1964 Medline

49. R. Evans, M. O’Neill, A. Pritzel, N. Antropova, A. Senior, T. Green, A. Žídek, R. Bates, S. Blackwell, J. Yim, O. Ronneberger, S. Bodenstein, M. Zielinski, A. Bridgland, A. Potapenko, A. Cowie, K. Tunyasuvunakool, R. Jain, E. Clancy, P. Kohli, J. Jumber, D. Hassabis, Protein complex prediction with AlphaFold-Multimer. bioRxiv 463034 [Preprint] (2021). https://doi.org/10.1101/2021.10.04.463034.

50. S. Basu, B. Wallner, DockQ: A Quality Measure for Protein-Protein Docking Models. *PLOS ONE* **11**, e0161879 (2016). doi:10.1371/journal.pone.0161879 Medline

51. K. Weissenow, M. Heinzinger, B. Rost, Protein language-model embeddings for fast, accurate, and alignment-free protein structure prediction. *Structure* **30**, 1169–1177.e4 (2022). doi:10.1016/j.str.2022.05.001 Medline

52. R. Wu, F. Ding, R. Wang, R. Shen, X. Zhang, S. Luo, C. Su, Z. Wu, Q. Xie, B. Berger, J. Ma, J. Peng, High-resolution de novo structure prediction from primary sequence. bioRxiv 500999 [Preprint] (2022). https://doi.org/10.1101/2022.07.21.500999

53. M. van Kempen, S. S. Kim, C. Tumescheit, M. Mirdita, C. L. M. Gilchrist, J. Söding, M. Steinegger, Foldseek: fast and accurate protein structure search. bioRxiv 479398 [Preprint] (2022). https://doi.org/10.1101/2022.02.07.479398.

54. S. C. Potter, A. Luciani, S. R. Eddy, Y. Park, R. Lopez, R. D. Finn, HMMER web server: 2018 update. *Nucleic Acids Res.* **46**, W200–W204 (2018). doi:10.1093/nar/gky448 Medline

55. M. Steinegger, J. Söding, MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nat. Biotechnol.* **35**, 1026–1028 (2017). doi:10.1038/nbt.3988 Medline

56. Y. Zhang, Protein structure prediction: When is it useful? *Curr. Opin. Struct. Biol.* **19**, 145–155 (2009). doi:10.1016/j.sbi.2009.02.005 Medline

57. Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, Y. Shmueli, A. dos Santos Costa, M. Fazel-Zarandi, T. Sercu, S. Candido, A. Rives, ESM-2 and ESMFold-v0 Model Code and Weights, Zenodo (2023). https://doi.org/10.5281/zenodo.7566741.

58. Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, Y. Shmueli, A. dos Santos Costa, M. Fazel-Zarandi, T. Sercu, S. Candido, A. Rives, ESM Atlas v0 representative random sample of predicted protein structures, Zenodo (2022). https://doi.org/10.5281/zenodo.7623482.

59. Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, Y. Shmueli, A. dos Santos Costa, M. Fazel-Zarandi, T. Sercu, S. Candido, A. Rives, ESM Atlas v0 random sample of high confidence predicted protein structures, Zenodo (2022). https://doi.org/10.5281/zenodo.7623627.

60. S. F. Altschul, W. Gish, W. Miller, E. W. Myers, D. J. Lipman, Basic local alignment search tool. *J. Mol. Biol.* **215**, 403–410 (1990). doi:10.1016/S0022-2836(05)80360-2 Medline

61. J. Meier, R. Rao, R. Verkuil, J. Liu, T. Sercu, A. Rives, "Language models enable zero-shot prediction of the effects of mutations on protein function" in *Advances in Neural Information Processing Systems* (Curran Associates, 2021), pp. 29287–29303.

62. C. Hsu, R. Verkuil, J. Liu, Z. Lin, B. Hie, T. Sercu, A. Lerer, A. Rives, "Learning inverse folding from millions of predicted structures" in *Proceedings of the 39th International Conference on Machine Learning* (PMLR, 2022), pp. 8946–8970.

63. P. Kunzmann, K. Hamacher, Biotite: A unifying open source computational biology framework in Python. *BMC Bioinformatics* **19**, 346 (2018). doi:10.1186/s12859-018-2367-z Medline

64. S. D. Dunn, L. M. Wahl, G. B. Gloor, Mutual information without the influence of phylogeny or entropy dramatically improves residue contact prediction. *Bioinformatics* **24**, 333–340 (2008). doi:10.1093/bioinformatics/btm604 Medline

65. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, É. Duchesnay, Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).

66. J. Su, Y. Lu, S. Pan, B. Wen, Y. Liu, RoFormer: Enhanced Transformer with Rotary Position Embedding. arXiv:2104.09864 [cs] (2021).

67. Y. You, J. Li, S. Reddi, J. Hseu, S. Kumar, S. Bhojanapalli, X. Song, J. Demmel, K. Keutzer, C.-J. Hsieh, Large Batch Optimization for Deep Learning: Training BERT in 76 Minutes. arXiv:1904.00962 [cs.LG] (2020).

68. S. Rajbhandari, J. Rasley, O. Ruwase, Y. He, "ZeRO: Memory Optimizations toward Training Trillion Parameter Models" in *Proceedings of the International Conference for*

High Performance Computing, Networking, Storage and Analysis (IEEE Press, 2020), article 20.

69. J. Ho, N. Kalchbrenner, D. Weissenborn, T. Salimans, Axial Attention in Multidimensional Transformers. arXiv:1912.12180 [cs.CV] (2019).

70. L. McInnes, J. Healy, J. Melville, UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. arXiv:1802.03426 [cs, stat] (2020).

71. F. A. Wolf, P. Angerer, F. J. Theis, SCANPY: Large-scale single-cell gene expression data analysis. *Genome Biol.* **19**, 15 (2018). doi:10.1186/s13059-017-1382-0 Medline

72. I. Virshup, S. Rybakov, F. J. Theis, P. Angerer, F. A. Wolf, anndata: Annotated data. bioRxiv 473007 [Preprint]. (2021). https://doi.org/10.1101/2021.12.16.473007.

73. F. Fuchs, D. Worrall, V. Fischer, M. Welling, "SE (*3*)-Transformers: 3D Roto-Translation Equivariant Attention Networks" in *Advances in Neural Information Processing Systems* (Curran Associates, 2020), pp. 1970–1981.

74. K. K. Yang, A. X. Lu, N. Fusi, Convolutions are competitive with transformers for protein sequence pretraining. bioRxiv 492714 [Preprint] (2022). https://doi.org/10.1101/2022.05.19.492714.