nature

https://doi.org/10.1038/s41586-021-03819-2

Accelerated Article Preview

Highly accurate protein structure prediction with AlphaFold

Received: 11 May 2021

Accepted: 12 July 2021

Accelerated Article Preview Published online 15 July 2021

Cite this article as: Jumper, J. et al. Highly accurate protein structure prediction with AlphaFold. *Nature* https://doi.org/10.1038/s41586-021-03819-2 (2021).

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli & Demis Hassabis

Highly accurate protein structure prediction with AlphaFold

https://doi.org/10.1038/s41586-021-03819-2

Received: 11 May 2021

Accepted: 12 July 2021

Published online: 15 July 2021

John Jumper[1,4]✉, Richard Evans[1,4], Alexander Pritzel[1,4], Tim Green[1,4], Michael Figurnov[1,4], Olaf Ronneberger[1,4], Kathryn Tunyasuvunakool[1,4], Russ Bates[1,4], Augustin Žídek[1,4], Anna Potapenko[1,4], Alex Bridgland[1,4], Clemens Meyer[1,4], Simon A. A. Kohl[1,4], Andrew J. Ballard[1,4], Andrew Cowie[1,4], Bernardino Romera-Paredes[1,4], Stanislav Nikolov[1,4], Rishub Jain[1,4], Jonas Adler[1], Trevor Back[1], Stig Petersen[1], David Reiman[1], Ellen Clancy[1], Michal Zielinski[1], Martin Steinegger[2,3], Michalina Pacholska[1], Tamas Berghammer[1], Sebastian Bodenstein[1], David Silver[1], Oriol Vinyals[1], Andrew W. Senior[1], Koray Kavukcuoglu[1], Pushmeet Kohli[1] & Demis Hassabis[1,4]✉

Proteins are essential to life, and understanding their structure can facilitate a mechanistic understanding of their function. Through an enormous experimental effort[1–4], the structures of around 100,000 unique proteins have been determined[5], but this represents a small fraction of the billions of known protein sequences[6,7]. Structural coverage is bottlenecked by the months to years of painstaking effort required to determine a single protein structure. Accurate computational approaches are needed to address this gap and to enable large-scale structural bioinformatics. Predicting the 3-D structure that a protein will adopt based solely on its amino acid sequence, the structure prediction component of the 'protein folding problem'[8], has been an important open research problem for more than 50 years[9]. Despite recent progress[10–14], existing methods fall far short of atomic accuracy, especially when no homologous structure is available. Here we provide the first computational method that can regularly predict protein structures with atomic accuracy even where no similar structure is known. We validated an entirely redesigned version of our neural network-based model, AlphaFold, in the challenging 14th Critical Assessment of protein Structure Prediction (CASP14)[15], demonstrating accuracy competitive with experiment in a majority of cases and greatly outperforming other methods. Underpinning the latest version of AlphaFold is a novel machine learning approach that incorporates physical and biological knowledge about protein structure, leveraging multi-sequence alignments, into the design of the deep learning algorithm.

The development of computational methods to predict 3-D protein structure from protein sequence has proceeded along two complementary paths that focus on either physical interactions or evolutionary history, respectively. The physical interaction programme heavily integrates our understanding of molecular driving forces into either thermodynamic or kinetic simulation of protein physics[16] or statistical approximations thereof[17]. While theoretically very appealing, this approach has proven highly challenging for even moderate-sized proteins due to the computational intractability of molecular simulation, the context-dependence of protein stability, and the difficulty of producing sufficiently accurate models of protein physics. The evolutionary program has provided an alternative in recent years, where constraints on protein structure are derived from bioinformatic analysis of protein evolutionary history, homology to solved structures[18,19], and pairwise evolutionary correlations[20–24]. This bioinformatic approach has benefited greatly from the steady growth of experimental protein structures deposited in the Protein Data Bank (PDB)[5], the explosion of genomic sequencing, and rapid development of deep learning techniques to interpret these correlations. Despite these advances, contemporary physical and evolutionary history-based approaches produce predictions that are far short of experimental accuracy in the majority of cases where a close homologue has not been solved experimentally and this has limited their utility for many biological applications.

In this work, we develop the first computational approach capable of predicting protein structures to near experimental accuracy in a majority of cases. The neural network AlphaFold that we developed was entered into the CASP14 assessment (May - July 2020; entered under the team name '*AlphaFold2*' and a completely different model from our CASP13 AlphaFold system[10]). The CASP assessment is carried out biennially using recently solved structures that have not been deposited in PDB or publicly disclosed so that it is a blind test for the participating methods, and has long served as the gold-standard assessment for the accuracy of structure prediction[25,26].

[1]DeepMind, London, UK. [2]School of Biological Sciences, Seoul National University, Seoul, South Korea. [3]Artificial Intelligence Institute, Seoul National University, Seoul, South Korea. [4]These authors contributed equally: John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Demis Hassabis.
✉e-mail: jumper@deepmind.com; dhcontact@deepmind.com

In CASP14, AlphaFold structures were vastly more accurate than competing methods. AlphaFold structures had a median backbone accuracy of 0.96 Å $RMSD_{95}$ (Cα root-mean-square deviation at 95% residue coverage) (95% CI = 0.85 Å - 1.16 Å) while the next best performing method had a median backbone accuracy of 2.8 Å $RMSD_{95}$ (95% CI = 2.7 Å - 4.0 Å) (measured on CASP domains, Fig. 1a for backbone and Suppl. Fig. 14 for all-atom). As a comparison point for this accuracy, the width of a Carbon atom is approximately 1.4 Å. In addition to very accurate domain structures (Fig. 1b), AlphaFold is able to produce highly accurate side chains (Fig. 1c) when the backbone is highly accurate and significantly improves over template-based methods even when strong templates are available. AlphaFold's all-atom accuracy was 1.5 Å $RMSD_{95}$ (95% CI = 1.2 Å - 1.6 Å) as compared to 3.5 Å $RMSD_{95}$ (95% CI = 3.1 Å - 4.2 Å) for the best alternative method. Finally, our methods are scalable to very long proteins with accurate domains and domain-packing (see Fig. 1d for prediction of a 2,180-residue protein with no structural homologues). Finally, the model is able to provide precise, per-residue estimates of its reliability which should enable confident use of these predictions.

We demonstrate in Fig. 2a that the high accuracy AlphaFold demonstrated in CASP14 extends to a large sample of recent PDB structures, where all structures were deposited in PDB after our training data cutoff and are analyzed as full chains (see Methods, Suppl. Fig. 15, and Suppl. Table 6 for more details). Furthermore, we observe high side chain accuracy when the backbone prediction is accurate (Fig. 2b), and we show that our predicted Local Distance Difference Test (pLDDT) confidence measure reliably predicts the Ca Local Distance Difference Test (lDDT-Cα) accuracy of the corresponding prediction (Fig. 2c). We also find that the global superposition metric template modeling score (TM-score)[27] can be accurately estimated (Fig. 2d). Overall, these validate that the high accuracy and reliability of AlphaFold on CASP14 proteins also transfers to an uncurated collection of recent PDB submissions, as would be expected (see Suppl. Methods 1.15 and Suppl. Fig. 11 for confirmation that this high accuracy extends to novel folds).

AlphaFold network

AlphaFold greatly improves the accuracy of structure prediction by incorporating novel neural network architectures and training procedures based on the evolutionary, physical, and geometric constraints of protein structure. In particular, we demonstrate a new architecture to jointly embed multiple sequence alignments (MSAs) and pairwise features, a new output representation and associated loss which enable accurate end-to-end structure prediction, a new equivariant attention architecture, use of intermediate losses to achieve iterative refinement of predictions, masked MSA loss to jointly train with structure, learning from unlabelled protein sequences using self-distillation, and self-estimates of accuracy.

The AlphaFold network directly predicts the 3-D coordinates of all heavy atoms for a given protein using the primary amino acid sequence and aligned sequences of homologues as inputs (Fig. 1e, see Methods for details of inputs including databases, MSA construction, and use of templates). A description of the most important ideas and components are below. The full network architecture and training procedure are provided in the Suppl. Methods.

The network comprises two main stages. First, the trunk of the network processes the inputs through repeated layers of a novel neural network block that we term Evoformer to produce an $N_{seq} \times N_{res}$ array (N_{seq}: number of sequences, N_{res}: number of residues) that represents a processed MSA and an $N_{res} \times N_{res}$ array that represents residue pairs. The MSA representation is initialized with the raw MSA, though see the Suppl. Methods 1.2.7 for details of handling very deep MSAs. The Evoformer blocks contain a number of novel attention-based and non-attention-based components. We show evidence in the Interpretability section that a concrete structural hypothesis arises early within the Evoformer blocks and is continuously refined. The key innovations in the Evoformer block are new mechanisms to exchange information within the MSA and pair representations that allow direct reasoning about spatial and evolutionary relationships.

The trunk of the network is followed by the Structure Module that introduces an explicit 3-D structure in the form of a rotation and translation for each residue of the protein (global rigid body frames). These representations are initialized in a trivial state with all rotations set to the identity and all positions set to the origin, but rapidly develop and refine a highly accurate protein structure with precise atomic details. Key innovations in this section of the network include breaking the chain atomic structure to allow simultaneous local refinement of all parts of the structure, a novel equivariant transformer to allow the network to implicitly reason about the unrepresented side chain atoms, and a loss term that places significant weight on orientational correctness of the residues. Both within the Structure Module and throughout the whole network, we reinforce the notion of iterative refinement by repeatedly applying the final loss to outputs then feeding the outputs recursively to the same modules. The iterative refinement using the whole network (that we term "recycling" and is related to approaches in computer vision[28,29]) contributes significantly to accuracy with minor extra training time (see Suppl. Methods 1.8 for details).

Evoformer

The key principle of the building block of the network, named Evoformer (Fig. 1e and 3a), is to view protein structure prediction as a graph inference problem in 3-D space where the edges of the graph are defined by residues in proximity. The elements of the pair representation encode information about the relation between residues (Fig. 3b). The columns of the MSA representation encode the individual residues of the input sequence while the rows represent the sequences in which those residues appear. Within this framework, we define a number of update operations that are applied in each block where the different update operations are applied in series.

The MSA representation updates the pair representation via an element-wise outer product that is summed over the MSA sequence dimension. Unlike previous work[30], this operation is applied within every block rather than once in the network which enables continuous communication from the evolving MSA representation to the pair representation.

Within the pair representation, there are two different update patterns. Both are inspired by the necessity of consistency of the pair representation -- for a pairwise description of amino acids to be representable as a single 3-D structure, many constraints must be satisfied including the triangle inequality on distances. Based on this intuition, we arrange the update operations on the pair representations in terms of triangles of edges involving three different nodes (Fig. 3c). In particular, we add an extra logit bias to axial attention[31] to include the "missing edge" of the triangle and we define a non-attention update operation "triangle multiplicative update" that uses two edges to update the missing third edge (see Suppl. Methods 1.6.5 for details). The triangle multiplicative update was developed originally as a more symmetric and somewhat cheaper replacement for the attention, and networks that use only the attention or multiplicative update are both able to produce high accuracy structures. The combination of the two updates is more accurate however.

We also use a variant of axial attention within the MSA representation. During the per-sequence attention in the MSA, we project additional logits from the pair stack to bias the MSA attention. This closes the loop by providing information flow from the pair representation back into the MSA representation, ensuring that the overall Evoformer block is able to fully mix information between the pair and MSA representations and prepare for structure generation within the Structure Module.

End-to-end structure prediction

The Structure Module (Fig. 3d) operates on a concrete 3-D backbone structure using the pair representation and the original sequence row

("single representation") of the MSA representation from the trunk. The 3-D backbone structure is represented as N_{res} independent rotations and translations each with respect to the global frame ("residue gas", Fig. 3e). These rotations and translations, representing the geometry of the N-Cα-C atoms, prioritize the orientation of the protein backbone so that the location of the side chain for each residue is highly constrained within that frame. Conversely, the peptide bond geometry is totally unconstrained and the network is observed to frequently violate the chain constraint during the application of the Structure Module as breaking this constraint allows local refinement of all parts of the chain without solving complex loop closure problems. Satisfaction of the peptide bond geometry is encouraged during fine-tuning by a violation loss term. Exact enforcement of peptide bond geometry is only achieved in the post-prediction relaxation of the structure by gradient descent in the Amber[32] forcefield. Empirically, this final relaxation does not improve accuracy of the model as measured by the global distance test (GDT)[33] or lDDT-Cα[34] but does remove distracting stereochemical violations without loss of accuracy.

The residue gas representation is updated iteratively in two stages (Fig. 3d). First a novel geometry-aware attention operation that we term Invariant Point Attention (IPA) is used to update an N_{res} set of neural activations ("single representation") without changing the 3-D positions, then an equivariant update operation is performed on the residue gas using the updated activations. The invariant point attention augments each of the usual attention queries, keys, and values with 3-D points produced in the local frame of each residue such that the final value is invariant to global rotations and translations (see Methods "Invariant Point Attention (IPA)" for details). The 3-D queries and keys also impose a strong spatial/locality bias on the attention which is well-suited to iterative refinement of the protein structure. After each attention operation and element-wise transition block, the module computes an update to the rotation and translation of each backbone frame. The application of these updates within the local frame of each residue makes the overall attention and update block an equivariant operation on the residue gas.

Predictions of side chain chi angles as well as the final, per-residue accuracy of the structure (pLDDT) are computed with small per-residue networks on the final activations at the end of the network. The estimate of the TM-score (pTM) is obtained from a pairwise error prediction that is computed as a linear projection from the final pair representation. The final loss (that we term the frame-aligned point error (FAPE) (Fig. 3f)) compares the predicted atom positions to the true positions under many different alignments. For each alignment, defined by aligning the predicted frame $(R_k, \mathbf{t}_k)$ to the corresponding true frame, we compute the distance of all predicted atom positions $\mathbf{x}_i$ from the true atom positions. The resulting $N_{\text{frames}} \times N_{\text{atoms}}$ distances are penalized with a clamped L1-loss. This creates a strong bias for atoms to be correct relative to the local frame of each residue and hence correct with respect to its side chain interactions, as well as providing the main source of chirality for AlphaFold (Suppl. Methods 1.9.3 and Suppl. Fig. 9).

Training with labelled and unlabelled data

The AlphaFold architecture is able to train to high accuracy using only supervised learning on PDB data, but we are able to enhance accuracy (see Fig. 4a) using an approach similar to noisy student self-distillation[35]. In this procedure, we use a trained network to predict the structure of ~350,000 diverse sequences from Uniclust30[36] and make a new dataset of predicted structures filtered to a high-confidence subset. We then train the same architecture again from scratch using a mixture of PDB and this new dataset of predicted structures as the training data, where the various training data augmentations such as cropping and MSA subsampling make it challenging for the network to recapitulate the previously-predicted structures. This self-distillation procedure makes effective use of the unlabelled sequence data and significantly improves the accuracy of the resulting network.

Additionally, we randomly mask out or mutate individual residues within the MSA and have a Bidirectional Encoder Representations from Transformers (BERT)-style[37] objective to predict the masked elements of the MSA sequences. This objective encourages the network to learn to interpret phylogenetic and covariation relationships without hard-coding a particular correlation statistic into the features. The BERT objective is trained jointly with the normal PDB structure loss on the same training examples and is not pre-trained, in contrast to recent independent work[38].

Interpreting the neural network

To understand how AlphaFold predicts protein structure, we trained a separate Structure Module for each of the 48 Evoformer blocks in the network while keeping all parameters of the main network frozen (Suppl. Methods 1.14). Including our recycling stages, this provides a trajectory of 192 intermediate structures, one per full Evoformer block, where each intermediate represents the network's belief of the most likely structure at that block. The resulting trajectories are surprisingly smooth after the first few blocks, showing that AlphaFold makes constant incremental improvements to the structure until it can no longer improve (see Fig. 4b for a trajectory of accuracy). These trajectories also illustrate the role of network depth. For very challenging proteins like SARS-CoV-2 Orf8 (T1064), the network searches and rearranges secondary structure elements for many layers before settling on a good structure. For other proteins like LmrP (T1024), the network finds the final structure within the first few layers. Refer to the Suppl. Videos 1-4 for structure trajectories of CASP14 targets T1024, T1044, T1064, and T1091 that show a clear iterative building process for a range of protein sizes and difficulties. In Suppl. Methods 1.16 and Suppl. Figs. 12-13, we interpret the attention maps produced by AlphaFold layers.

Fig. 4a contains detailed ablations of the components of AlphaFold that demonstrate that a variety of different mechanisms contribute to AlphaFold accuracy. Please refer to the Suppl. Methods 1.13 for detailed descriptions of each ablation model, their training details, extended discussion of ablation results, and the effect of MSA depth on each ablation (Suppl. Fig. 10).

MSA depth and cross-chain contacts

While AlphaFold has high accuracy across the vast majority of deposited PDB structures, we note that there are still factors that affect accuracy or limit applicability of the model. The model uses multiple sequence alignments and accuracy drops substantially when the mean alignment depth is less than ~30 sequences (see Fig. 5a for details). We observe a threshold effect where improvements in MSA depth over ~100 sequences lead to small gains. We hypothesize that the MSA information is needed to coarsely find the correct structure within the early stages of the network, but refinement of that prediction into a high-accuracy model does not depend crucially on the MSA information. The other substantial limitation that we have observed is that AlphaFold is much weaker for proteins that have few intra-chain or homotypic contacts as compared to the number of heterotypic contacts. This typically occurs for bridging domains within larger complexes where the shape of the protein is created almost entirely by interactions with other chains in the complex. Conversely, AlphaFold is often able to give high accuracy predictions for homomers, even when the chains are substantially intertwined (e.g. Fig. 5b). We expect the AlphaFold ideas to be readily applicable to predicting full hetero-complexes in a future system and that this will remove the difficulty with protein chains that have a large number of hetero-contacts.

Related work

Protein structure prediction has had a long and varied development, which is extensively covered in a number of excellent reviews[14,39–42]. Despite the long history of applying neural networks to structure prediction[14,41,42], they have only recently come to improve structure prediction[10,11,43,44]. These approaches effectively leverage the rapid improvement in computer vision systems[45] by treating the problem of protein structure prediction as converting an "image" of evolutionary couplings[22–24] to an "image" of the protein distance matrix then integrating the distance predictions into a heuristic system that produces the final 3-D coordinate prediction. A few recent works have been developed to predict 3-D coordinates directly[46–49], but the accuracy of these approaches does not match traditional, hand-crafted structure prediction pipelines[50]. In parallel, the success of attention-based networks for language processing[51] and more recently computer vision[31,52] has inspired exploration of attention-based methods for interpreting protein sequences[53–55].

Discussion

The methodology we have taken in designing AlphaFold is a combination of the bioinformatic and physical approaches: we use a physical and geometric inductive bias to build components that learn from PDB data with minimal imposition of handcrafted features (e.g. AlphaFold builds hydrogen bonds effectively without a hydrogen bond score function). This results in a network that learns far more efficiently from the limited data in the PDB but is able to cope with the complexity and variety of structural data.

In particular, AlphaFold is able to handle missing physical context and produce accurate models in challenging cases like intertwined homomers or proteins that only fold in the presence of an unknown heme group. The ability to handle underspecified structural conditions is essential to learning from PDB structures as the PDB represents the full range of conditions in which structures have been solved. In general, AlphaFold is trained to produce the protein structure most likely to appear as part of a PDB structure. In cases where a particular stoichiometry or ligand/ion is predictable from the sequence alone, AlphaFold is likely to produce a structure that respects those constraints implicitly.

AlphaFold has already demonstrated its utility to the experimental community, both for molecular replacement[56], and interpreting cryogenic electron microscopy (cryo-EM) maps[57]. Moreover, because AlphaFold outputs protein coordinates directly, AlphaFold produces predictions in graphics processing unit (GPU)-minutes to GPU-hours depending on the length of the protein sequence (e.g. around one GPU-minute per model for 384 residues, see Methods for details). This opens up the exciting possibility of predicting structures at the proteome-scale and beyond.

The explosion in available genomic sequencing techniques and data has revolutionized bioinformatics but the intrinsic challenge of experimental structure determination has prevented a similar expansion in our structural knowledge. By developing an accurate protein structure prediction algorithm, coupled with existing large and well-curated structure and sequence databases assembled by the experimental community, we hope to accelerate the advancement of structural bioinformatics that can keep pace with the genomics revolution. We hope that AlphaFold, and computational approaches that apply its techniques for other biophysical problems, will become essential tools of modern biology.

Online content

Any methods, additional references, Nature Research reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41586-021-03819-2.

1. Thompson, M. C., Yeates, T. O. & Rodriguez, J. A. Advances in methods for atomic resolution macromolecular structure determination. *F1000Res.* **9**, (2020).
2. Bai, X.-C., McMullan, G. & Scheres, S. H. W. How cryo-EM is revolutionizing structural biology. *Trends Biochem. Sci.* **40**, 49–57 (2015).
3. Jaskolski, M., Dauter, Z. & Wlodawer, A. A brief history of macromolecular crystallography, illustrated by a family tree and its Nobel fruits. *FEBS J.* **281**, 3985–4009 (2014).
4. Wüthrich, K. The way to NMR structures of proteins. *Nat. Struct. Biol.* **8**, 923–925 (2001).
5. wwPDB Consortium. Protein Data Bank: the single global archive for 3D macromolecular structure data. *Nucleic Acids Res.* **47**, D520–D528 (2018).
6. Mitchell, A. L. *et al.* MGnify: the microbiome analysis resource in 2020. *Nucleic Acids Res.* **48**, D570–D578 (2019).
7. Steinegger, M., Mirdita, M. & Söding, J. Protein-level assembly increases protein sequence recovery from metagenomic samples manyfold. *Nat. Methods* **16**, 603–606 (2019).
8. Dill, K. A., Ozkan, S. B., Shell, M. S. & Weikl, T. R. The protein folding problem. *Annu. Rev. Biophys.* **37**, 289–316 (2008).
9. Anfinsen, C. B. Principles that Govern the Folding of Protein Chains. *Science* **181**, 223–230 (1973).
10. Senior, A. W. *et al.* Improved protein structure prediction using potentials from deep learning. *Nature* **577**, 706–710 (2020).
11. Wang, S., Sun, S., Li, Z., Zhang, R. & Xu, J. Accurate De Novo Prediction of Protein Contact Map by Ultra-Deep Learning Model. *PLoS Comput. Biol.* **13**, e1005324 (2017).
12. Zheng, W. *et al.* Deep-learning contact-map guided protein structure prediction in CASP13. *Proteins: Struct. Funct. Bioinf.* **87**, 1149–1164 (2019).
13. Abriata, L. A., Tamò, G. E. & Dal Peraro, M. A further leap of improvement in tertiary structure prediction in CASP13 prompts new routes for future assessments. *Proteins: Struct. Funct. Bioinf.* **87**, 1100–1112 (2019).
14. Pearce, R. & Zhang, Y. Deep learning techniques have significantly impacted protein structure prediction and protein design. *Curr. Opin. Struct. Biol.* **68**, 194–207 (2021).
15. Moult, J., Fidelis, K., Kryshtafovych, A., Schwede, T. & Topf, M. *Critical Assessment of Techniques for Protein Structure Prediction, Fourteenth Round: Abstract Book*. (2020).
16. Brini, E., Simmerling, C. & Dill, K. Protein storytelling through physics. *Science* **370**, (2020).
17. Sippl, M. J. Calculation of conformational ensembles from potentials of mena force: an approach to the knowledge-based prediction of local structures in globular proteins. *J. Mol. Biol.* **213**, 859–883 (1990).
18. Šali, A. & Blundell, T. L. Comparative protein modelling by satisfaction of spatial restraints. *J. Mol. Biol.* **234**, 779–815 (1993).
19. Roy, A., Kucukural, A. & Zhang, Y. I-TASSER: a unified platform for automated protein structure and function prediction. *Nat. Protoc.* **5**, 725–738 (2010).
20. Altschuh, D., Lesk, A. M., Bloomer, A. C. & Klug, A. Correlation of co-ordinated amino acid substitutions with function in viruses related to tobacco mosaic virus. *Journal of Molecular Biology* **193**, 693–707 (1987).
21. Shindyalov, I. N., Kolchanov, N. A. & Sander, C. Can three-dimensional contacts in protein structures be predicted by analysis of correlated mutations? *Protein Eng.* **7**, 349–358 (1994).
22. Weigt, M., White, R. A., Szurmant, H., Hoch, J. A. & Hwa, T. Identification of direct residue contacts in protein-protein interaction by message passing. *Proceedings of the National Academy of Sciences* **106**, 67–72 (2009).
23. Marks, D. S. *et al.* Protein 3D structure computed from evolutionary sequence variation. *PLoS One* **6**, e28766 (2011).
24. Jones, D. T., Buchan, D. W. A., Cozzetto, D. & Pontil, M. PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. *Bioinformatics* **28**, 184–190 (2012).
25. Moult, J., Pedersen, J. T., Judson, R. & Fidelis, K. A large-scale experiment to assess protein structure prediction methods. *Proteins: Structure, Function, and Genetics* **23**, (1995).
26. Kryshtafovych, A., Schwede, T., Topf, M., Fidelis, K. & Moult, J. Critical assessment of methods of protein structure prediction (CASP)—Round XIII. *Proteins: Struct. Funct. Bioinf.* **87**, 1011–1020 (2019).
27. Zhang, Y. & Skolnick, J. Scoring function for automated assessment of protein structure template quality. *Proteins: Struct. Funct. Bioinf.* **57**, 702–710 (2004).
28. Tu, Z. & Bai, X. Auto-context and its application to high-level vision tasks and 3d brain image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **32**, 1744–1757 (2009).
29. Carreira, J., Agrawal, P., Fragkiadaki, K. & Malik, J. Human pose estimation with iterative error feedback. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 4733–4742 (2016).
30. Mirabello, C. & Wallner, B. rawMSA: End-to-end Deep Learning using raw Multiple Sequence Alignments. *PLoS One* **14**, e0220182 (2019).
31. Huang, Z. *et al.* CCNet: Criss-Cross Attention for Semantic Segmentation. in *Proceedings of the IEEE/CVF International Conference on Computer Vision* 603–612 (2019).
32. Hornak, V. *et al.* Comparison of multiple Amber force fields and development of improved protein backbone parameters. *Proteins: Struct. Funct. Bioinf.* **65**, 712–725 (2006).
33. Zemla, A. LGA – a Method for Finding 3D Similarities in Protein Structures. *Nucleic Acids Res.* **31**, 3370–3374 (2003).
34. Mariani, V., Biasini, M., Barbato, A. & Schwede, T. lDDT: A local superposition-free score for comparing protein structures and models using distance difference tests. *Bioinformatics* **29**, 2722–2728 (2013).
35. Xie, Q., Luong, M.-T., Hovy, E. & Le, Q. V. Self-training with noisy student improves imagenet classification. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* 10687–10698 (2020).

36. Mirdita, M. *et al*. Uniclust databases of clustered and deeply annotated protein sequences and alignments. *Nucleic Acids Res*. **45**, D170–D176 (2017).
37. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* **1**, 4171–4186 (2019).
38. Rao, R. *et al*. MSA Transformer. *biorXiv preprint 2021. 02. 12. 430858* (2021).
39. Kuhlman, B. & Bradley, P. Advances in protein structure prediction and design. *Nat. Rev. Mol. Cell Biol.* **20**, 681–697 (2019).
40. Marks, D. S., Hopf, T. A. & Sander, C. Protein structure prediction from sequence variation. *Nat. Biotechnol*. **30**, 1072–1080 (2012).
41. Qian, N. & Sejnowski, T. J. Predicting the secondary structure of globular proteins using neural network models. *J. Mol. Biol*. **202**, 865–884 (1988).
42. Fariselli, P., Olmea, O., Valencia, A. & Casadio, R. Prediction of contact maps with neural networks and correlated mutations. *Protein Eng*. **14**, 835–843 (2001).
43. Yang, J. *et al*. Improved protein structure prediction using predicted interresidue orientations. *Proc. Natl. Acad. Sci. U. S. A*. **117**, 1496–1503 (2020).
44. Li, Y. *et al*. Deducing high-accuracy protein contact-maps from a triplet of coevolutionary matrices through deep residual convolutional networks. *PLoS Comput. Biol*. **17**, e1008865 (2021).
45. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 770–778 (2016).
46. AlQuraishi, M. End-to-End Differentiable Learning of Protein Structure. *Cell Systems* **8**, 292–301.e3 (2019).
47. Senior, A. W. *et al*. Protein structure prediction using multiple deep neural networks in the 13th Critical Assessment of Protein Structure Prediction (CASP13). *Proteins: Struct. Funct. Bioinf*. **87**, 1141–1148 (2019).
48. Ingraham, J., Riesselman, A. J., Sander, C. & Marks, D. S. Learning Protein Structure with a Differentiable Simulator. in *Proceedings of the International Conference on Learning Representations* (2019).
49. Li, J. Universal Transforming Geometric Network. *arXiv preprint arXiv:1908. 00723* (2019).
50. Xu, J., Mcpartlon, M. & Li, J. Improved protein structure prediction by deep learning irrespective of co-evolution information. *Nature Machine Intelligence* (2021).
51. Vaswani, A. *et al*. Attention Is All You Need. in *Advances in Neural Information Processing Systems* 5998–6008 (2017).
52. Wang, H. *et al*. Axial-deeplab: Stand-alone axial-attention for panoptic segmentation. in *European Conference on Computer Vision* 108–126 (Springer, 2020).
53. Alley, E. C., Khimulya, G., Biswas, S., AlQuraishi, M. & Church, G. M. Unified rational protein engineering with sequence-based deep representation learning. *Nat. Methods* **16**, 1315–1322 (2019).
54. Heinzinger, M. *et al*. Modeling aspects of the language of life through transfer-learning protein sequences. *BMC Bioinformatics* **20**, 1–17 (2019).
55. Rives, A. *et al*. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences* **118**, (2021).
56. Pereira, J. *et al*. High-accuracy protein structure prediction in CASP14. *Proteins* (2021) https://doi.org/10.1002/prot.26171.
57. Gupta, M. *et al*. CryoEM and AI reveal a structure of SARS-CoV-2 Nsp2, a multifunctional protein involved in key host processes. *bioRxiv* (2021) https://doi.org/10.1101/2021.05.10.443524.

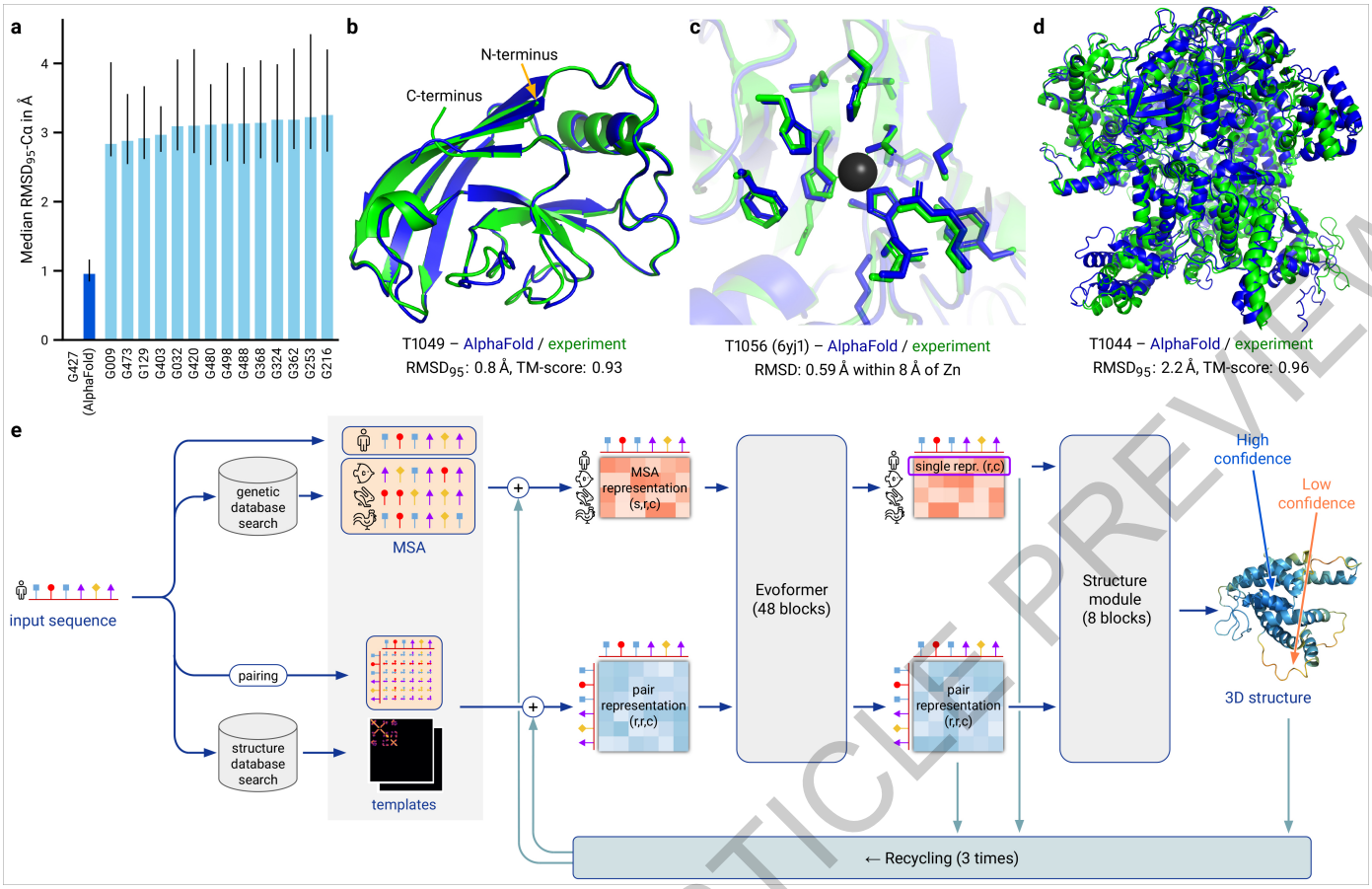

Fig. 1 | AlphaFold produces highly accurate structures. (**a**) AlphaFold's performance on the CASP14 set (N=87 protein domains) relative to the top-15 entries (out of 146), group numbers correspond to the numbers assigned to entrants by CASP; error bars represent the 95% confidence interval of the median, estimated with 10,000 bootstrap samples. (**b**) Our prediction of CASP14 target T1049 (blue) compared to the true (experimental) structure (green). Four residues from the C-terminus of the crystal structure are B-factor outliers and are not depicted. (**c**) An example of a well predicted zinc binding site (AlphaFold has accurate side chains even though it does not explicitly predict the zinc ion). (**d**) CASP target T1044, a 2,180-residue single chain, predicted with correct domain packing (prediction made after CASP using AlphaFold without intervention). (**e**) Model architecture. Arrows show the information flow among the various components described in this paper. Array shapes are shown in brackets with s: number of sequences, r: number of residues and c: number of channels.

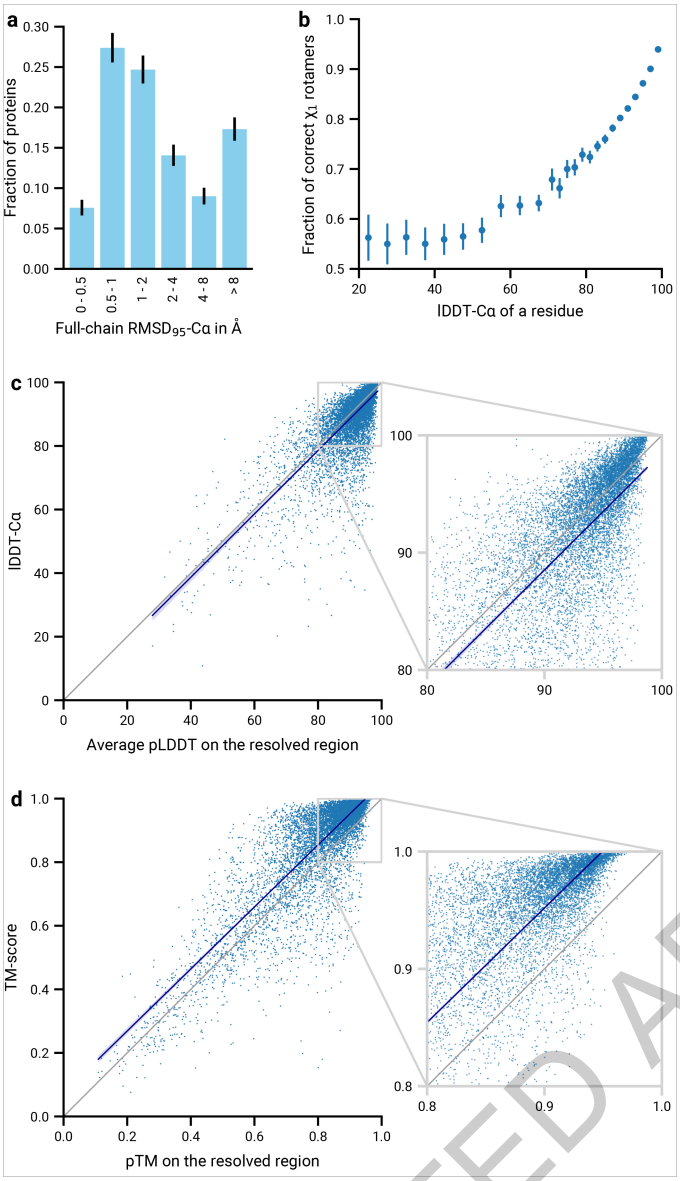

Fig. 2 | AlphaFold accuracy on recent PDB. Structures analyzed are newer than any structure in the training set. Further filtering is applied to reduce redundancy (see Methods). (**a**) Histogram of backbone RMSD for full chains (Cα RMSD at 95% coverage); error bars are 95% confidence intervals (Poisson). Excludes proteins with a template (identified by hmmsearch) from the training set with more than 40% sequence identity covering more than 1% of the chain (N = 3,144 protein chains). Overall median is 1.46 Å. Note that this measure will be highly sensitive to domain packing and domain accuracy; high RMSD is expected for some chains with uncertain packing or packing errors. (**b**) Correlation between backbone accuracy and sidechain accuracy. Filtered to structures with any observed side chains and resolution better than 2.5 Å (N=5,317 protein chains); sidechains further filtered to B-factor < 30 $Å^2$. A rotamer is classified as correct if the predicted torsion angle is within 40 degrees. Each point aggregates a range of lDDT-Cα, with bin size 2 units above 70 lDDT-Cα and 5 units otherwise. Points correspond to mean accuracy; error bars are 95% confidence intervals (Student-t) for the mean on a per-residue basis. **(c)** Confidence score compared to true accuracy on chains. Least-squares linear fit lDDT_Cα = 0.997 * pLDDT - 1.17 (Pearson r=0.76). (N=10,795 protein chains). Shaded region of the linear fit represents a 95% confidence interval estimated with 10,000 bootstrap samples. **(d)** Correlation between pTM and full-chain TM-score. Least-squares linear fit TM-score = 0.98 * pTM + 0.07 (Pearson r=0.85). (N=10,795 protein chains). Shaded region of the linear fit represents a 95% confidence interval estimated with 10,000 bootstrap samples.

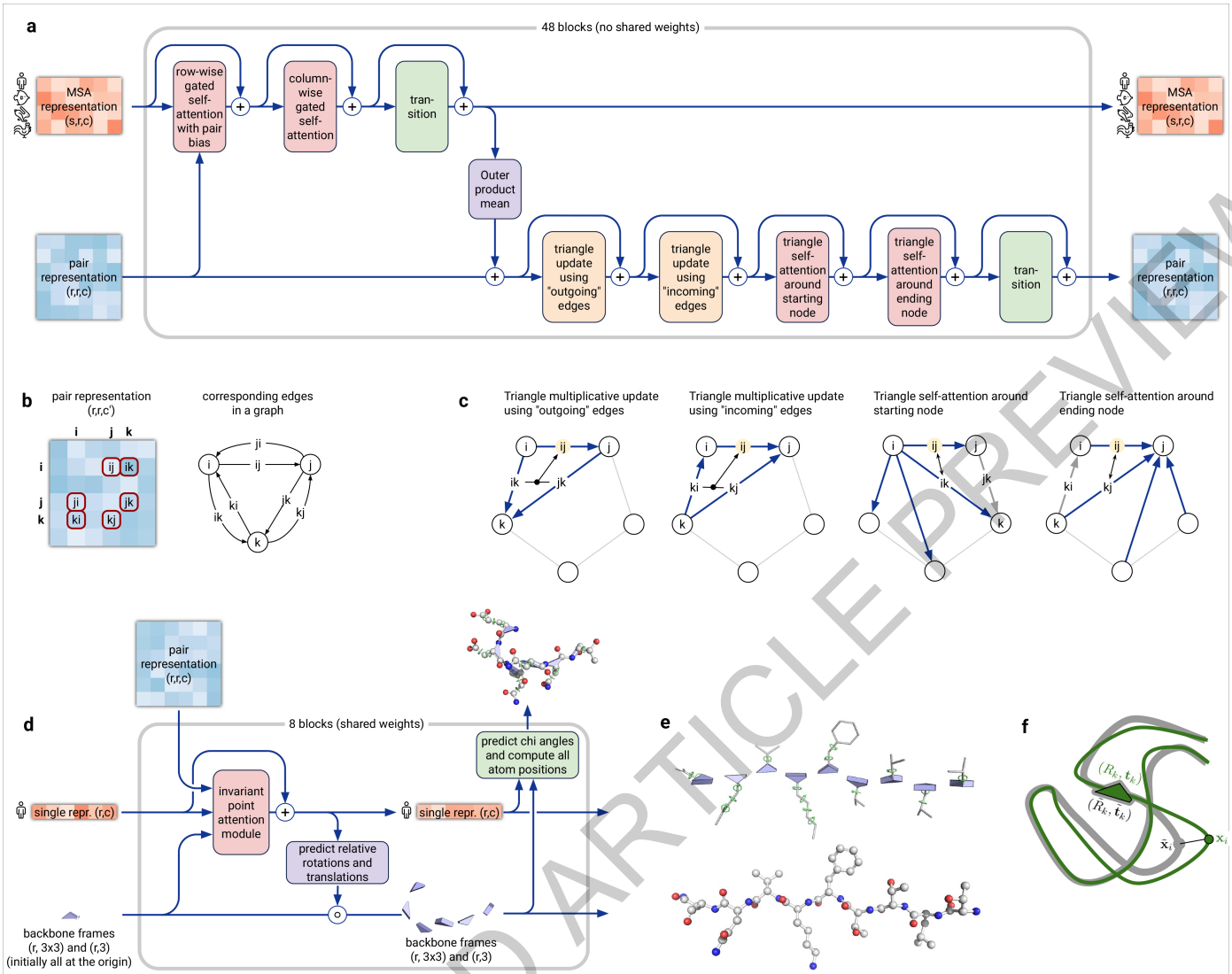

Fig. 3 | Architectural details. (a) Evoformer block. Arrows show the information flow. The shape of the arrays is shown in brackets with s: number of sequences, r: number of residues, and c: number of channels. **(b)** The pair representation interpreted as directed edges in a graph. **(c)** Triangle multiplicative update and Triangle self-attention. The circles represent residues. Entries in the pair representation are illustrated as directed edges and in each diagram, the edge being updated is *ij*. **(d)** Structure Module. The single representation is a copy of the first row of the MSA representation. **(e)** "Residue gas": Representation of each residue as one free-floating rigid body for the backbone (blue triangles) and chi-angles for the side chains (green circles). The corresponding atomic structure is shown below. **(f)** Frame-aligned point error (FAPE). green: predicted structure, grey: true structure, $(R_k, \mathbf{t}_k)$: frames, $\mathbf{x}_i$: atom positions.

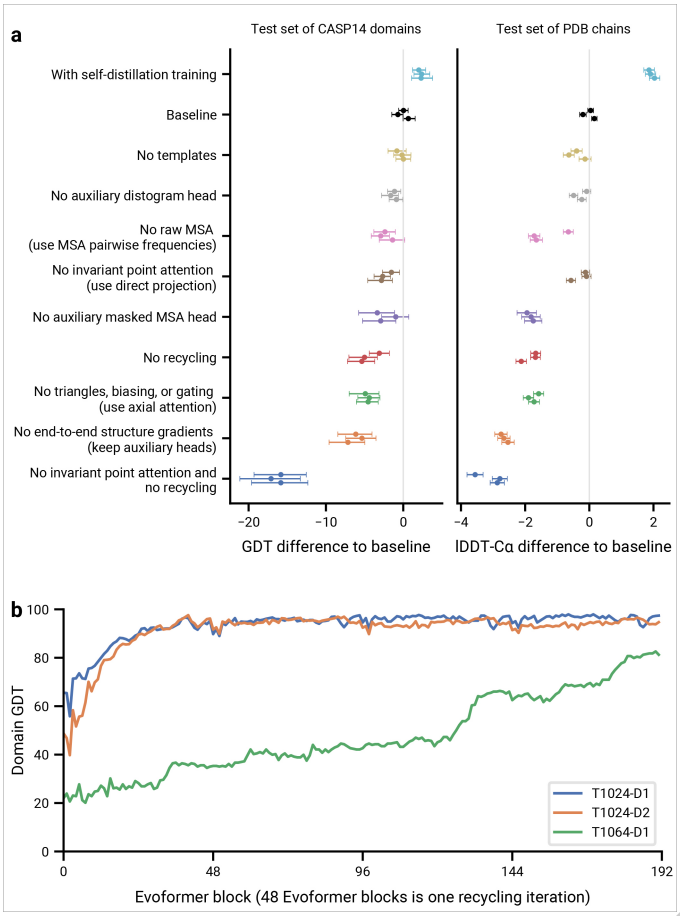

Fig. 4 | Interpreting the neural network. (a) Ablation results on two target sets: CASP14 set of domains (N=87 protein domains) and PDB test set of chains with template coverage ≤ 30% at 30% identity (N=2,261 protein chains). Domains are scored with GDT and chains are scored with lDDT-Cα. The ablations are reported as a difference to the average of the 3 baseline seeds. Means and error bars are computed using bootstrap estimates with 10,000 samples. The points denote mean and the error bars denote 95% bootstrap percentile intervals. **(b)** Domain GDT trajectory over 4 recycling iterations and 48 Evoformer blocks on CASP14 targets LmrP (T1024) and Orf8 (T1064). Both T1024 domains obtain the correct structure early in the network, while the structure of T1064 changes multiple times and requires nearly the full depth of the network to reach the final structure.

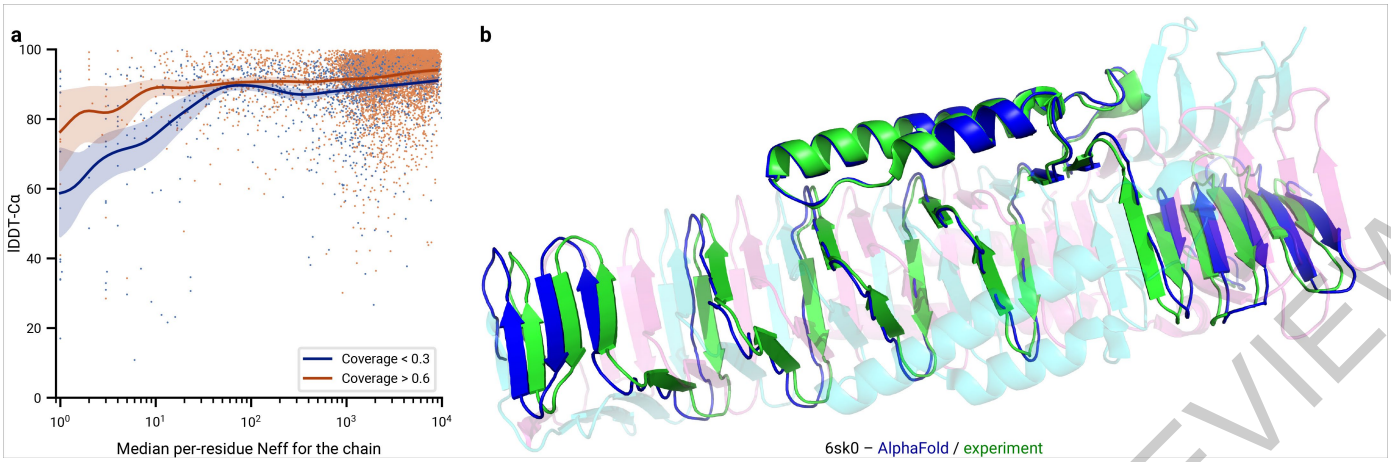

Fig. 5 | Effect of MSA depth and cross-chain contacts. (**a**) Backbone accuracy (lDDT-Cα) for the redundancy-reduced set of the PDB after our training data cutoff, restricting to proteins where at most 25% of long-range contacts are between different heteromer chains. We further consider two groups of proteins based on template coverage at 30% sequence identity: covering more than 60% of the chain (N=6,743 protein chains) and covering less than 30% of the chain (N=1,596 protein chains). MSA depth is computed by counting the number of non-gap residues for each position in the MSA (using the Neff weighting scheme, see Methods for details) and taking the median across residues. The curves are obtained via Gaussian kernel average smoothing (window size is 0.2 units in $\log_{10}$ Neff); the shaded area is the 95% confidence interval estimated using bootstrap with 10,000 samples. **(b)** An intertwined homotrimer is correctly predicted without input stoichiometry and only a weak template (blue is predicted and green is experimental).

Methods

Full algorithm details

Extensive explanations of the components and their motivations are available in Suppl. Methods 1.1-1.10, as well as pseudocode in Suppl. Algorithms 1-32, network diagrams in Suppl. Figs. 1-8, input features in Suppl. Table 1, and extra details in Suppl. Tables 2-3. Training and inference details are provided in Suppl. Methods 1.11-1.12 and Suppl. Tables 4-5.

Invariant Point Attention (IPA)

The invariant point attention module combines the pair representation, the single representation and the geometric representation to update the single representation (see Suppl. Fig. 8). Each of them contributes affinities to the shared attention weights and then uses these weights to map its values to the output. The invariant point attention operates in 3-D space. Each residue produces query points, key points and value points in its local frame. These points are projected into the global frame using the residue's backbone frame where they interact with each other. The resulting points are then projected back into the local frame. The affinity computation in the 3-D space uses squared distances and the coordinate transformations ensure the invariance of this module with respect to the global frame (see Suppl. Methods 1.8.2 **"Invariant point attention (IPA)"** for the algorithm, proof of invariance, and a description of the full multi-head version). A related construction that uses classical geometric invariants to construct pairwise features in place of the learned 3-D points has been applied to protein design[58].

In addition to the invariant point attention, standard dot product attention is computed on the abstract single representation and a special attention on the pair representation. The pair representation augments both the logits and the values of the attention process, which is the primary way in which the pair representation controls the structure generation.

Inputs and data sources (including BFD)

Inputs to the network are the primary sequence, sequences from evolutionarily related proteins in the form of a multiple sequence alignment (MSA) created by standard tools including jackhmmer[59] and HHBlits[60], and 3-D atom coordinates of a small number of homologous structures (templates) where available. For both the MSA and templates, the search processes are tuned for high recall; spurious matches will likely appear in the raw MSA but this matches the training condition of the network.

One of the sequence databases used, Big Fantastic Database (BFD), was custom made and released publicly (see Data Availability) and was used by several CASP teams. BFD is the largest publicly available collection of protein families. It consists of 65,983,866 families represented as MSAs and Hidden Markov Models (HMMs) covering 2,204,359,010 protein sequences from reference databases, metagenomes and metatranscriptomes.

BFD was built in three steps: (1) 2,423,213,294 protein sequences were collected from Uniprot (Swiss-Prot&TrEMBL 2017-11)[61], a soil reference protein catalog (SRC) and the marine eukaryotic reference catalog (MERC)[7] and clustered to 30% sequence identity, while enforcing a 90% alignment coverage of the shorter sequence using MMseqs2/Linclust[62]. This resulted in 345,159,030 clusters. For computational efficiency we removed all clusters with less than three members, resulting in 61,083,719 clusters. (2) We added 166,510,624 representative protein sequences from Metaclust NR (2017-05; discarding all sequences shorter than 150 residues)[62] by aligning them against the cluster representatives using MMseqs2[63]. Sequences that fulfilled the sequence identity and coverage criteria were assigned to the best scoring cluster. The remaining 25,347,429 sequences that could not be assigned were clustered separately and added as new clusters, resulting in the final clustering. (3) For each of the clusters, we computed an MSA using FAMSA[64] and computed the HMMs following the Uniclust HH-suite database protocol[36].

The following versions of public datasets were used in this study: Our models were trained on a copy of the Protein Data Bank (PDB)[5] downloaded on 28th August 2019. For finding template structures at prediction time, we used a copy of the PDB downloaded on 14th May 2020, and the PDB70[65] clustering database downloaded on 13th May 2020. For MSA lookup at both training and prediction time, we used Uniref90[66] version 2020_01, BFD, Uniclust30[36] version 2018_08 and MGnify[6] version 2018_12. For sequence distillation, we used Uniclust30[36] version 2018_08 to construct a distillation structure dataset. Full details are given in the Suppl. Methods 1.2.

For MSA search on BFD+Uniclust30, and template search against PDB70, we used HHBlits[60] and HHSearch[65] from hh-suite version 3.0-beta.3 (14/07/2017). For MSA search on Uniref90 and clustered MGnify, we used jackhmmer from HMMER3[67]. For constrained relaxation of structures, we used OpenMM v7.3.1[68] with the Amber99sb force field[32]. For neural network construction, running and other analysis, we used TensorFlow[69], Sonnet[70], NumPy[71], Python[72], and Colab[73].

To quantify the impact of the different sequence data sources, we re-ran the CASP14 proteins using the same models but varying how the MSA was constructed. Removing BFD reduced mean accuracy by 0.4 GDT, removing Mgnify reduced mean accuracy by 0.7 GDT, and removing both reduced mean accuracy by 6.1 GDT. In each case, we observe most targets to have very small changes in accuracy but a few outliers to have very large (20+ GDT) differences. This is consistent with the results in Fig. 5a where the depth of the MSA is relatively unimportant until it approaches a threshold value of ~30 sequences when the MSA-size effects become quite large. We observe fairly overlapping effects between inclusion of BFD and Mgnify, but having at least one of these metagenomics databases is very important for target classes that are poorly represented in UniRef, and having both was necessary to achieve full CASP accuracy.

Training regimen

To train we use structures from the PDB with a maximum release date of 2018-04-30. Chains are sampled in inverse proportion to cluster size of a 40% sequence identity clustering. We then randomly crop them to 256 residues and assemble into batches of size 128. We train the model on Tensor Processing Unit (TPU) v3 with a batch size of 1 per TPU core, hence the model uses 128 TPUv3 cores. The model is trained until convergence (~10 M samples) and further fine-tuned using longer crops of 384 residues, larger MSA stack, and reduced learning rate (see Suppl. Methods 1.11 for the exact configuration). The initial training stage takes approximately one week, and the fine-tuning stage takes approximately 4 more days.

The network is supervised by the FAPE loss and a number of auxiliary losses. First, the final pair representation is linearly projected to a binned distance distribution (distogram) predictions, scored with a cross-entropy loss. Second, we employ random masking on the input MSAs and require the network to reconstruct the masked regions from the output MSA representations using a BERT-like loss[37]. Third, the output single representations of the Structure Module are used to predict binned per-residue lDDT-Cα values. Finally, we employ an auxiliary side-chain loss during training, and an auxiliary structure violation loss during fine-tuning. Detailed description and weighting is provided in the SI.

An initial model trained with the above objectives was used to make structure predictions for a Uniclust dataset of 355,993 sequences with the full MSAs. These predictions were then used to train a final model with identical hyperparameters, except for sampling examples 75% of the time from the Uniclust prediction set, with sub-sampled MSAs, and 25% of the time from the clustered PDB set.

We train five different models using different random seeds, some with templates and some without, to encourage diversity in the

predictions (see Suppl. Table 5 and Suppl. Methods 1.12.1 for details). We also fine-tune these models after CASP14 to add a pTM prediction objective (Suppl. Methods 1.9.7), and use the obtained models for Fig. 2(d).

Inference regimen

We inference the five trained models and use the predicted confidence score to select the best model per target.

Using our CASP14 configuration for AlphaFold, the trunk of the network is run multiple times with different random choices for the MSA cluster centres (see Suppl. Methods 1.11.2 for details of the ensembling procedure). The full time to make a structure prediction varies significantly in the length of the protein. Representative timings for the neural network using a single model on V100 GPU are 4.8 minutes at 256 residues, 9.2 minutes at 384 residues, and 18 hours at 2,500 residues. These timings are measured using our open-source code, and the open-source code is somewhat faster than the version we ran in CASP14 since we now use the XLA compiler[74].

Since CASP14, we have found that the accuracy of the network without ensembling is very close or equal to the accuracy with ensembling and we turn off ensembling for most inference. Without ensembling, the network is 8x faster and the representative timings for a single model are 0.6 minutes at 256 residues, 1.1 minutes at 384 residues, and 2.1 hours at 2,500 residues.

Inferencing large proteins can easily exceed the memory of a single GPU. For a V100 with 16 GB of memory, we can predict the structure of proteins up to ~1,300 residues without ensembling and the 256- and 384-residue inference times are using a single GPU's memory. The memory usage is approximately quadratic in the number of residues, so a 2,500 residue protein involves using unified memory so that we can greatly exceed the memory of a single V100. In our cloud setup, a single V100 is used for computation on a 2,500 residue protein but we requested four GPUs to have sufficient memory.

Searching genetic sequence databases to prepare inputs and final relaxation of the structures take additional central processing unit (CPU) time but do not require a GPU or TPU.

Metrics

The predicted structure is compared to the true structure from the PDB in terms of the local distance difference test (lDDT) metric[34] since this metric reports domain accuracy without requiring a domain segmentation of chain structures. The distances are either computed between all heavy atoms (lDDT) or the Cα atoms only to measure the backbone accuracy (lDDT-Cα). As lDDT-Cα only focuses on the Cα atoms, it does not include the penalty for structural violations and clashes. Domain accuracies in CASP are reported in GDT[33] and TM-score[27] is used as a full-chain global superposition metric.

We also report accuracies by the $RMSD_{95}$ (Cα RMSD at 95% coverage). We perform 5 iterations of (a) least-squares alignment of the predicted structure and the PDB structure on the currently chosen Cα atoms (using all Cα atoms at the first iteration); (b) selecting 95% Cα atoms with the lowest alignment error. The root mean squared deviation (RMSD) of the atoms chosen at the final iterations is the $RMSD_{95}$. This metric is more robust to apparent errors that can originate from crystal structure artefacts, though of course in some cases the removed 5% of residues will contain genuine modelling errors.

Test set of recent PDB sequences

For evaluation on recent PDB sequences (Fig. 2a, 2b, 2c, 2d, 4a, 5a) we used a copy of the PDB downloaded 15/02/2021. Structures were filtered to those with a release date after 30/04/2018 (the date limit for inclusion in AlphaFold's training set). Chains were further filtered to remove sequences consisting of a single amino acid as well as sequences with an ambiguous chemical component at any residue position. Exact duplicates were removed, with the chain with the most resolved Cα atoms used as the representative sequence. Subsequently, structures with less than 16 resolved residues, with unknown residues, or solved by nuclear magnetic resonance (NMR) methods were removed. Since the PDB contains many near-duplicate sequences, the chain with the highest resolution was selected from each cluster in the PDB 40% sequence clustering of the data. Furthermore, we removed all sequences where fewer than 80 amino acids had the alpha carbon resolved and removed chains with more than 1,400 residues. The final dataset contains 10,795 protein sequences.

The procedure for filtering the recent PDB dataset based on prior template identity was as follows. Hmmsearch was run with default parameters against a copy of the PDB SEQRES fasta downloaded 15/2/21. Template hits were accepted if the associated structure had a release date earlier than 2018/04/30. Each residue position in a query sequence was assigned the max identity of any template hit covering that position. Filtering then proceeded as described in the individual figure legends, based on a combination of maximum identity and sequence coverage.

The MSA depth analysis was based on computing the normalized number of effective sequences (Neff) for each position of a query sequence. Per-residue Neff was obtained by counting the number of non-gap residues in the MSA for this position and weighting the sequences using the Neff scheme[75] with a threshold of 80% sequence identity measured on the region that is non-gap in either sequence.

Reporting summary

Further information on research design is available in the Nature Research Reporting Summary linked to this paper.

Data availability

All input data are freely available from public sources.

Structures from the PDB were used for training and as templates (https://www.wwpdb.org/ftp/pdb-ftp-sites; for the associated sequence data and 40% sequence clustering see also https://ftp.wwpdb.org/pub/pdb/derived_data/ and https://cdn.rcsb.org/resources/sequence/clusters/bc-40.out). Training used a version of the PDB downloaded 28/08/2019, while CASP14 template search used a version downloaded 14/05/2020. Template search also used the PDB70 database, downloaded 13/05/2020 (https://wwwuser.gwdg.de/~compbiol/data/hhsuite/databases/hhsuite_dbs/).

We show experimental structures from the PDB with accessions 6Y4F[76], 6YJ1[77], 6VR4[78], 6SK0[79], 6FES[80], 6W6W[81], 6T1Z[82], and 7JTL[83].

For MSA lookup at both training and prediction time, we used UniRef90 v2020_01 (https://ftp.ebi.ac.uk/pub/databases/uniprot/previous_releases/release-2020_01/uniref/), BFD (https://bfd.mmseqs.com), Uniclust30 v2018_08 (https://wwwuser.gwdg.de/~compbiol/uniclust/2018_08/), and MGnify clusters v2018_12 (https://ftp.ebi.ac.uk/pub/databases/metagenomics/peptide_database/2018_12/). Uniclust30 v2018_08 was further used as input for constructing a distillation structure dataset.

Code availability

Source code for the AlphaFold model, trained weights, and inference script are available under an open-source license at https://github.com/deepmind/alphafold .

Neural networks were developed with TensorFlow v1 (https://github.com/tensorflow/tensorflow), Sonnet v1 (https://github.com/deepmind/sonnet), JAX v0.1.69 (https://github.com/google/jax/), and Haiku v0.0.4 (https://github.com/deepmind/dm-haiku). The XLA compiler is bundled with JAX and does not have a separate version number.

For MSA search on BFD+Uniclust30, and for template search against PDB70, we used HHBlits and HHSearch from hh-suite v3.0-beta.3 14/07/2017 (https://github.com/soedinglab/hh-suite). For MSA search on UniRef90 and clustered MGnify, we used jackhmmer from HMMER v3.3 (http://eddylab.org/software/hmmer/). For constrained relaxation

of structures, we used OpenMM v7.3.1 (https://github.com/openmm/openmm) with the Amber99sb force field.

Construction of BFD used MMseqs2 version 925AF (https://github.com/soedinglab/MMseqs2) and FAMSA v1.2.5 (https://github.com/refresh-bio/FAMSA).

Data analysis used Python v3.6 (https://www.python.org/), NumPy v1.16.4 (https://github.com/numpy/numpy), SciPy v1.2.1 (https://www.scipy.org/), seaborn v0.11.1 (https://github.com/mwaskom/seaborn), Matplotlib v3.3.4 (https://github.com/matplotlib/matplotlib), bokeh v1.4.0 (https://github.com/bokeh/bokeh), pandas v1.1.5 (https://github.com/pandas-dev/pandas), plotnine v0.8.0 (https://github.com/has2k1/plotnine), statsmodels v0.12.2 (https://github.com/statsmodels/statsmodels) and Colab (https://research.google.com/colaboratory). TM-align v20190822 (https://zhanglab.dcmb.med.umich.edu/TM-align/) was used for computing TM-scores. Structure visualizations were created in Pymol v2.3.0 (https://github.com/schrodinger/pymol-open-source).

58. Ingraham, J., Garg, V. K., Barzilay, R. & Jaakkola, T. Generative models for graph-based protein design. in *Proceedings of the 33rd Conference on Neural Information Processing Systems* (2019).
59. Johnson, L. S., Eddy, S. R. & Portugaly, E. Hidden Markov model speed heuristic and iterative HMM search procedure. *BMC Bioinformatics* **11**, 1–8 (2010).
60. Remmert, M., Biegert, A., Hauser, A. & Söding, J. HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nat. Methods* **9**, 173–175 (2012).
61. Bateman, A. *et al*. UniProt: the universal protein knowledgebase in 2021. *Nucleic Acids Res*. (2020).
62. Steinegger, M. & Söding, J. Clustering huge protein sequence sets in linear time. *Nat. Commun*. **9**, 1–8 (2018).
63. Steinegger, M. & Söding, J. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nat. Biotechnol*. **35**, 1026–1028 (2017).
64. Deorowicz, S., Debudaj-Grabysz, A. & Gudyś, A. FAMSA: Fast and accurate multiple sequence alignment of huge protein families. *Sci. Rep*. **6**, 1–13 (2016).
65. Steinegger, M. *et al*. HH-suite3 for fast remote homology detection and deep protein annotation. *BMC Bioinformatics* **20**, 1–15 (2019).
66. Suzek, B. E. *et al*. UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics* **31**, 926–932 (2015).
67. Eddy, S. R. Accelerated profile HMM searches. *PLoS Comput. Biol*. **7**, e1002195 (2011).
68. Eastman, P. *et al*. OpenMM 7: Rapid development of high performance algorithms for molecular dynamics. *PLoS Comput. Biol*. **13**, 1–17 (2017).
69. Ashish, A. M. A. *et al*. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. *arXiv preprint arXiv:1603. 04467* (2015).
70. Reynolds, M. *et al*. Open sourcing Sonnet - a new library for constructing neural networks. https://deepmind.com/blog/open-sourcing-sonnet/ (2017).
71. Harris, C. R. *et al*. Array programming with NumPy. *Nature* **585**, 357–362 (2020).
72. Van Rossum, G. & Drake, F. L. *Python 3 Reference Manual*. (CreateSpace, 2009).
73. Bisong, E. Google Colaboratory. in *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners* 59–64 (Apress, 2019).
74. XLA: Optimizing *Compiler for TensorFlow*. https://www.tensorflow.org/xla.
75. Wu, T., Hou, J., Adhikari, B. & Cheng, J. Analysis of several key factors influencing deep learning-based inter-residue contact prediction. *Bioinformatics* **36**, 1091–1098 (2020).
76. Jiang, W. *et al*. MrpH, a new class of metal-binding adhesin, requires zinc to mediate biofilm formation. *PLoS Pathog*. **16**, e1008707 (2020).
77. Dunne, M., Ernst, P., Sobieraj, A., Pluckthun, A. & Loessner, M. J. The M23 peptidase domain of the Staphylococcal phage 2638A endolysin. https://doi.org/10.2210/pdb6YJ1/pdb (2020).
78. Drobysheva, A. V. *et al*. Structure and function of virion RNA polymerase of a crAss-like phage. *Nature* **589**, 306–309 (2021).
79. Flaugnatti, N. *et al*. Structural basis for loading and inhibition of a bacterial T6SS phospholipase effector by the VgrG spike. *EMBO J*. **39**, e104129 (2020).
80. ElGamacy, M. *et al*. An Interface-Driven Design Strategy Yields a Novel, Corrugated Protein Architecture. *ACS Synth. Biol*. **7**, 2226–2235 (2018).
81. Lim, C. J. *et al*. The structure of human CST reveals a decameric assembly bound to telomeric DNA. *Science* **368**, 1081–1085 (2020).
82. Debruycker, V. *et al*. An embedded lipid in the multidrug transporter LmrP suggests a mechanism for polyspecificity. *Nat. Struct. Mol. Biol*. **27**, 829–835 (2020).
83. Flower, T. G. *et al*. Structure of SARS-CoV-2 ORF8, a rapidly evolving immune evasion protein. *Proc. Natl. Acad. Sci. U. S. A*. **118**, (2021).

Acknowledgements We would like to thank Alban Rrustemi, Albert Gu, Alexey Guseynov, Blake Hechtman, Charlie Beattie, Chris Jones, Craig Donner, Emilio Parisotto, Erich Elsen, Florentina Popovici, George Necula, Hector Maclean, Jacob Menick, James Kirkpatrick, James Molloy, Jason Yim, Jeff Stanway, Karen Simonyan, Laurent Sifre, Lena Martens, Matthew Johnson, Michael O'Neill, Natasha Antropova, Raia Hadsell, Sam Blackwell, Sanjoy Das, Shaobo Hou, Stephan Gouws, Steven Wheelwright, Tom Hennigan, Tom Ward, Zachary Wu, Žiga Avsec and the Research Platform Team for their contributions; Milot Mirdita for his help with datasets; Mildred Piovesan-Forster, Alexander Nelson and Rupert Kemp for their help managing the project; the JAX, TensorFlow, and XLA teams for detailed support and enabling machine learning models of the complexity of AlphaFold, and our colleagues at DeepMind, Google, and Alphabet for their encouragement and support. We would also like to thank John Moult and the CASP14 organizers, and the experimentalists whose structures enabled the assessment. Martin Steinegger acknowledges support from the National Research Foundation of Korea grant [2019R1A6A1A10073437, 2020M3A9G7103933] and the Creative-Pioneering Researchers Program through Seoul National University.

Author contributions J.J. and D.H. led the research. J.J., R.E., Al.P., M.F., O.R., R.B., An.P., S.K., B.R.-P., J.A., M.P., Ta.B. and O.V. developed neural network architecture and training. T.G., A.Z., K.T., R.B., Al.B., R.E., An.B., A.C., S.N., R.J., D.R., M.Z. and S.B. developed the data, analytics, and inference systems. D.H., K.K., P.K., C.M. and E.C. managed the research. T.G. led the technical platform. P.K., A.S., K.K., O.V., D.S., S.P. and Tr.B. contributed technical advice and ideas. M.S. created the BFD genomics database and provided technical assistance on HHBlits. D.H., R.E., A.S. and K.K. conceived the AlphaFold project. J.J., R.E. and A.S. conceived the end-to-end approach. J.J., Al.P., O.R., An.P., R.E., M.F., T.G., K.T., C.M. and D.H. wrote the paper.

Competing interests J.J., R.E., Al.P., T.G., M.F., O.R., R.B., A.B., S.K., D.R. and A.S. have filed provisional patent applications relating to machine learning for predicting protein structures. The remaining authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41586-021-03819-2.

Correspondence and requests for materials should be addressed to J.J. or D.H.

Peer review information *Nature* thanks Charlotte Deane, Mohammed AlQuraishi and Yang Zhang for their contribution to the peer review of this work.

nature research

Corresponding author(s): John Jumper, Demis Hassabis

Last updated by author(s): Jul 11, 2021

Reporting Summary

Nature Research wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Research policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed	
☐	☒	The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☒	☐	A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☐	☒	The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	☐	A description of all covariates tested
☐	☒	A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☐	☒	A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☒	☐	For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☐	☒	For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	☐	For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☐	☒	Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	Source code for the AlphaFold model, trained weights, and inference script will be made available under an open-source license at https://github.com/deepmind/ upon publication. Neural networks were developed with TensorFlow v1 (https://github.com/tensorflow/tensorflow), Sonnet v1 (https://github.com/deepmind/sonnet), JAX v0.1.69 (https://github.com/google/jax/), and Haiku v0.0.4 (https://github.com/deepmind/dm-haiku). The XLA compiler is bundled with JAX and does not have a separate version number. For MSA search on BFD+Uniclust30, and for template search against PDB70, we used HHBlits and HHSearch from hh-suite v3.0-beta.3 14/07/2017 (https://github.com/soedinglab/hh-suite). For MSA search on UniRef90 and clustered MGnify, we used jackhmmer from HMMER v3.3 (http://eddylab.org/software/hmmer/). For constrained relaxation of structures, we used OpenMM v7.3.1 (https://github.com/openmm/openmm) with the Amber99sb force field. Construction of BFD used MMseqs2 version 925AF (https://github.com/soedinglab/MMseqs2) and FAMSA v1.2.5 (https://github.com/refresh-bio/FAMSA).
Data analysis	Data analysis used Python v3.6 (https://www.python.org/), NumPy v1.16.4 (https://github.com/numpy/numpy), SciPy v1.2.1 (https://www.scipy.org/), seaborn v0.11.1 (https://github.com/mwaskom/seaborn), Matplotlib v3.3.4 (https://github.com/matplotlib/matplotlib), bokeh v1.4.0 (https://github.com/bokeh/bokeh), pandas v1.1.5 (https://github.com/pandas-dev/pandas), plotnine v0.8.0 (https://github.com/has2k1/plotnine), statsmodels v0.12.2 (https://github.com/statsmodels/statsmodels) and Colab (https://research.google.com/colaboratory). TM-align v20190822 (https://zhanglab.dcmb.med.umich.edu/TM-align/) was used for computing TM-scores. Structure visualizations were created in Pymol v2.3.0 (https://github.com/schrodinger/pymol-open-source).

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Research guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A list of figures that have associated raw data
- A description of any restrictions on data availability

All input data are freely available from public sources.

Structures from the PDB were used for training and as templates (https://www.wwpdb.org/ftp/pdb-ftp-sites; for the associated sequence data and 40% sequence clustering see also https://ftp.wwpdb.org/pub/pdb/derived_data/ and https://cdn.rcsb.org/resources/sequence/clusters/bc-40.out). Training used a version of the PDB downloaded 28/08/2019, while CASP14 template search used a version downloaded 14/05/2020. Template search also used the PDB70 database, downloaded 13/05/2020 (https://wwwuser.gwdg.de/~compbiol/data/hhsuite/databases/hhsuite_dbs/).

We show experimental structures from the PDB with accessions 6Y4F77, 6YJ178, 6VR479, 6SK080, 6FES81, 6W6W82, 6T1Z83, and 7JTL84.

For MSA lookup at both training and prediction time, we used UniRef90 v2020_01 (https://ftp.ebi.ac.uk/pub/databases/uniprot/previous_releases/release-2020_01/uniref/), BFD (https://bfd.mmseqs.com), Uniclust30 v2018_08 (https://wwwuser.gwdg.de/~compbiol/uniclust/2018_08/), and MGnify clusters v2018_12 (https://ftp.ebi.ac.uk/pub/databases/metagenomics/peptide_database/2018_12/).
Uniclust30 v2018_08 was further used as input for constructing a distillation structure dataset.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	No sample size was chosen; the method was evaluated on the full CASP14 benchmark set, and all PDB chains not in the training set (subject to the exclusions noted below).
Data exclusions	The recent PDB set was filtered (see Methods for full details). Briefly this excludes chains with too few resolved residues, longer than 1400 residues, solved by NMR or with unknown/ambiguous residues. This set was also redundancy reduced (by taking representatives from a sequence clustering), and for some figures a sequence similarity-based filter was applied to remove entries too similar to the training set (see Methods and figure legends for details).
Replication	Not applicable, no experimental work is described in this study. The results are the output of a computational method which will be made available.
Randomization	Not applicable, we are not making a comparison between two groups
Blinding	Not applicable, we are not making a comparison between two groups

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology and archaeology
☒	☐ Animals and other organisms
☒	☐ Human research participants
☒	☐ Clinical data
☒	☐ Dual use research of concern

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging