nature biotechnology

ARTICLES

https://doi.org/10.1038/s41587-022-01432-w

Single-sequence protein structure prediction using a language model and deep learning

Ratul Chowdhury[1,8], Nazim Bouatta[1,8] ✉, Surojit Biswas[2,3,8], Christina Floristean[4,8], Anant Kharkare[4], Koushik Roye[4], Charlotte Rochereau[5], Gustaf Ahdritz[6], Joanna Zhang[4], George M. Church[1,2], Peter K. Sorger[1,7] ✉ and Mohammed AlQuraishi[4,6] ✉

AlphaFold2 and related computational systems predict protein structure using deep learning and co-evolutionary relationships encoded in multiple sequence alignments (MSAs). Despite high prediction accuracy achieved by these systems, challenges remain in (1) prediction of orphan and rapidly evolving proteins for which an MSA cannot be generated; (2) rapid exploration of designed structures; and (3) understanding the rules governing spontaneous polypeptide folding in solution. Here we report development of an end-to-end differentiable recurrent geometric network (RGN) that uses a protein language model (AminoBERT) to learn latent structural information from unaligned proteins. A linked geometric module compactly represents C_α backbone geometry in a translationally and rotationally invariant way. On average, RGN2 outperforms AlphaFold2 and RoseTTAFold on orphan proteins and classes of designed proteins while achieving up to a 10^6-fold reduction in compute time. These findings demonstrate the practical and theoretical strengths of protein language models relative to MSAs in structure prediction.

Predicting three-dimensional (3D) protein structure from amino acid sequence is a major challenge in biophysics of practical and theoretical importance. Progress has long relied on physics-based methods that estimate energy landscapes and dynamically fold proteins within these landscapes[1–4]. A decade ago, the focus shifted to extracting residue–residue contacts from co-evolutionary relationships embedded in MSAs[5] (Supplementary Fig. 1). Algorithms such as the first AlphaFold[6] and trRosetta[7] use deep neural networks to generate distograms able to guide classic physics-based folding engines. These algorithms perform substantially better than algorithms based on physical energy models alone. More recently, the superior performance of AlphaFold2 (AF2) (ref. [8]) in folding a wide range of protein targets that were part of the recent CASP14 prediction challenge shows that, when MSAs are available, machine learning (ML)–based methods can predict protein structure with sufficient accuracy to complement X-ray crystallography, cryogenic electron microscopy and nuclear magnetic resonance (NMR) as a practical means to determine structures of interest.

Predicting the structures of single sequences using ML nonetheless remains a challenge: the requirement in AF2 for co-evolutionary information from MSAs makes it less performant with proteins that lack sequence homologs, currently estimated at ~20% of all metagenomic protein sequences[9] and ~11% of eukaryotic and viral proteins[10]. Protein design and studies quantifying the effects of sequence variation on function[11] or immunogenicity[12] also require single-sequence structure prediction. More fundamentally, the physical process of polypeptide folding in solution is driven solely by the chemical properties of that chain and its interaction with solvent (excluding, for the moment, proteins that require folding co-factors). An algorithm that predicts structure directly from a single sequence is—like energy-based folding engines[1–4]—closer to the real physical process than an algorithm that uses MSAs. We speculate that ML algorithms able to fold proteins from single sequences will ultimately provide new understanding of protein biophysics.

Structure prediction algorithms that are fast are of great practical value because they make efficient exploration of sequence space possible, particularly in design applications[13]. Fast predictions for large numbers of long proteins would enable many practical applications in enzymology, therapeutics and chemical engineering, including designing new functions[14–16], raising thermostability[17], altering pH sensitivity[18] and increasing compatibility with organic solvents[19]. Efficient and accurate structure prediction is also valuable in the case of orphan proteins, many of which are thought to play a role in taxonomically restricted and lineage-specific adaptations. OSP24, for example, is an orphan virulence factor for the wheat pathogen *Fusarium graminearum* that controls host immunity by regulating proteasomal degradation of a conserved signal transduction kinase[20]. It is one of many orphan genes found in fungi, plants, insects and other organisms[21] for which MSAs are not available.

We previously described an end-to-end differentiable, ML-based RGN (hereafter RGN1)[22] that predicts protein structure from position-specific scoring matrices (PSSMs) derived from MSAs; related end-to-end approaches have since been reported[23–25]. RGN1 PSSM structure relationships are parameterized as torsion angles between adjacent residues, making it possible to sequentially position the protein backbone in 3D space (backbone geometry comprises the arrangement of *N*, *Cα* and *C′* atoms for each amino acid). All RGN1 components are differentiable, and the system can, therefore, be optimized from end to end to minimize prediction error (as measured by distance-based root mean squared deviation

[1]Laboratory of Systems Pharmacology, Program in Therapeutic Science, Harvard Medical School, Boston, MA, USA. [2]Department of Biomedical Informatics, Harvard Medical School, Boston, MA, USA. [3]Nabla Bio, Inc., Boston, MA, USA. [4]Department of Computer Science, Columbia University, New York, NY, USA. [5]Integrated Program in Cellular, Molecular, and Biomedical Studies, Columbia University, New York, NY, USA. [6]Department of Systems Biology, Columbia University, New York, NY, USA. [7]Department of Systems Biology, Harvard Medical School, Boston, MA, USA. [8]These authors contributed equally: Ratul Chowdhury, Nazim Bouatta, Surojit Biswas, Christina Floristean. ✉e-mail: nazim_bouatta@hms.harvard.edu; peter_sorger@hms.harvard.edu; ma4129@cumc.columbia.edu

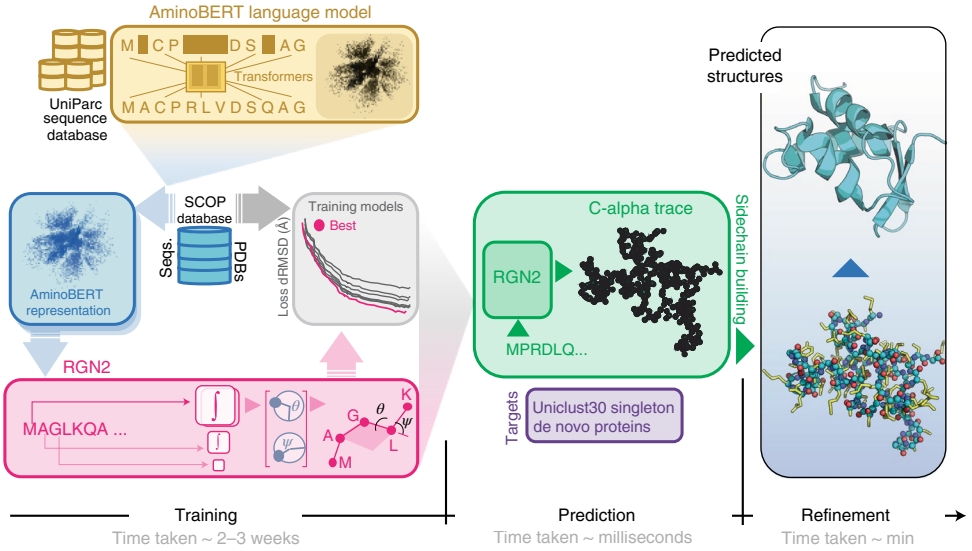

Fig. 1 | Organization and application of RGN2. RGN2 combines a transformer-based protein language model (AminoBERT, yellow) with an RGN that uses Frenet–Serret frames to generate the backbone structure of a protein (green). After initial construction of the sidechains and hydrogen-bonded networks, refinement of the structure is subsequently performed using AF2Rank (blue).

(dRMSD)). Although RGN1 does not rely on the co-evolutionary information used to generate MSAs, a requirement for PSSMs necessitates that multiple homologous sequences be available.

Here we describe an end-to-end differentiable system, RGN2 (Fig. 1), that predicts protein structure from single protein sequences by using a protein language model (AminoBERT). Language models were first developed as a means to extract semantic information from a sequence of words (a key requirement for natural language processing)[26]. In the context of proteins, AminoBERT aims to capture the latent information in a string of amino acids that implicitly specifies protein structure. RGN2 also makes use of a natural way of describing polypeptide geometry that is rotationally and translationally invariant at the level of the polypeptide as a whole. This involves using the Frenet–Serret formulas to embed a reference frame at each C_α carbon; the backbone is then easily constructed by a series of transformations. In this paper, we describe the implementation and training of AminoBERT, the use of Frenet–Serret formulas in RGN2 and a performance assessment for natural and designed proteins with no significant sequence homologs. We found that, on average, the Global Distance Test–Total Score (GDT_TS) achieved by RGN2 is higher than AF2 (ref. [8]) and RoseTTAFold (RF)[27], even though AF2/RF can achieve higher absolute GDT_TS scores than RGN2 on naturally occurring orphan proteins without known homologs and de novo designed proteins. Although RGN2 is not as performant as MSA-based methods for proteins that permit use of MSAs, RGN2 is up to six orders of magnitude faster, enabling efficient exploration of sequence and structure landscapes.

Results

RGN2 and AminoBERT models. RGN2 involves two primary innovations relative to RGN1 and other ML-based structure prediction approaches. First, it uses amino acid sequence itself as the primary input as opposed to a PSSM, making it possible to predict structure from a single sequence. In the absence of a PSSM or MSA, latent information on the relationship between protein sequence (as a whole) and 3D structure is captured using a protein language model that we termed AminoBERT. Second, rather than describe the geometry of protein backbones as a sequence of torsion angles, RGN2 uses a simpler approach based on the Frenet–Serret formulas; these formulas describe motion along a curve using the reference frame of the curve itself. This approach to protein geometry is inherently translationally and rotationally invariant, a key property of polypeptides in solution. We refined structures predicted by RGN2 using an AF2Rank-based protocol[28] that imputes the backbone and sidechain atoms. The refinement process contains steps that are non-differentiable but improve the quality of predicted structures.

Language models were originally developed for natural language processing and operate on a simple but powerful principle: they acquire linguistic understanding by learning to fill in missing words in a sentence, akin to a sentence completion task in standardized tests. By performing this task across large text corpora, language models develop reasoning capabilities. The Bidirectional Encoder Representations from Transformers (BERT) model[29] instantiated this principle using transformers, a class of neural networks in which attention is the primary component of the learning system[30]. In a transformer, each token in the input sentence can 'attend' to all other tokens through the exchange of activation patterns corresponding to the intermediate outputs of neurons in the neural network. In AminoBERT, we use the same approach, substituting protein sequences for sentences and using amino acid residues as tokens.

To generate the AminoBERT language model, we trained a 12-layer transformer using ~250 million natural protein sequences obtained from the UniParc sequence database[31]. To enhance the capture of information in full protein sequences, we introduced two training objectives not part of BERT or previously reported protein language models[26,32–36]. First, 2–8 contiguous residues were masked simultaneously in each sequence (similar to the ProtTrans[37] language model), making the reconstruction task harder and emphasizing learning from global, rather than local, context. Second, 'chunk permutation' was used to swap contiguous protein segments; chunk permutations preserve local sequence information but disrupt global coherence. Training AminoBERT to identify these permutations is another way of encouraging the transformer to discover information from the protein sequence as a whole. The AminoBERT module of RGN2 is trained independently of the geometry module in a self-supervised manner without fine-tuning (Methods).

In RGN2, we parameterized backbone geometry using the discrete version of the Frenet–Serret formulas for one-dimensional

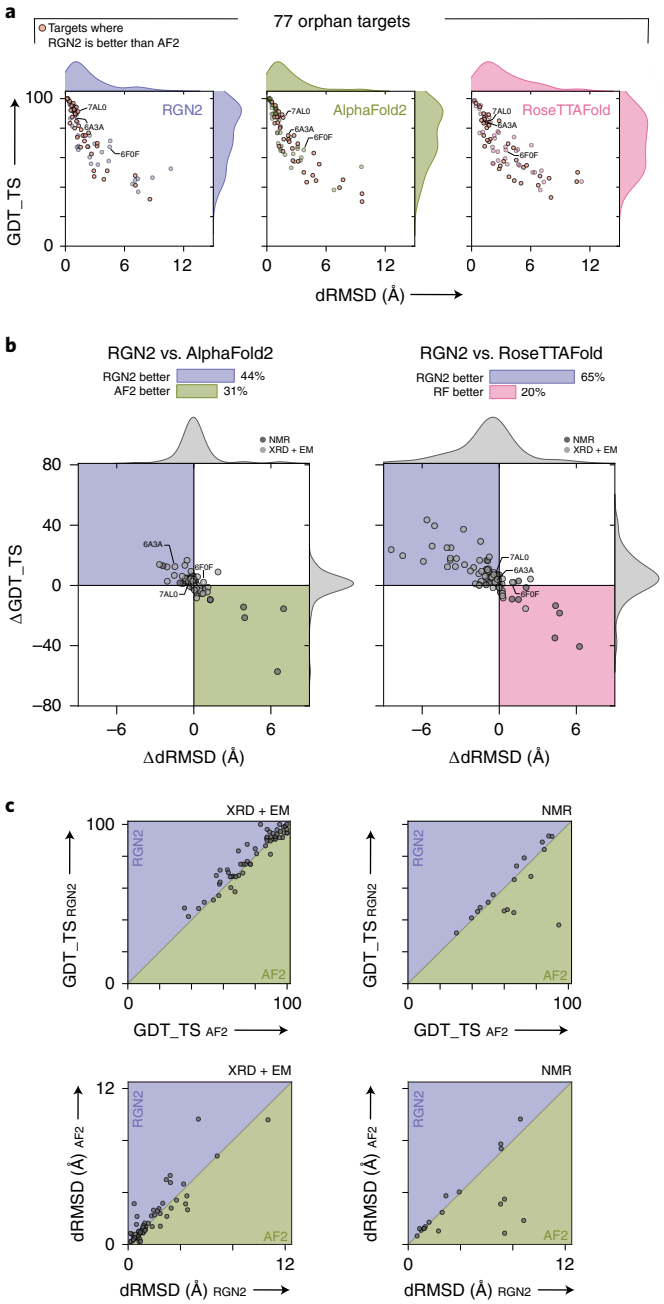

Fig. 2 | Prediction performance on orphan proteins. a, Absolute performance metrics for RGN2 (purple), AF2 (green) and RF (pink) across 77 orphan proteins lacking known homologs. **b**, Differences in prediction accuracy between RGN2 and AF2/RF are shown for the 77 orphan proteins, using dRMSD and GDT_TS as metrics. Points in the top-left quadrant correspond to targets with negative ΔdRMSD and positive ΔGDT_TS—that is, where RGN2 outperforms the competing method on both metrics, and vice versa for the bottom-right quadrant. The other two quadrants (white) indicate targets where there is no clear winner, as the two metrics disagree. The structures of 20% of the targets were determined experimentally using NMR and are denoted with dark gray markers, whereas the remaining 80% of targets were determined using X-ray crystallography or electron microscopy. **c**, Head-to-head comparisons of absolute GDT_TS and dRMSD scores for RGN2 and AF2 are shown broken down by experimental method (NMR and X-ray crystallography/electron microscopy). RGN2 outperforms AF2 for proteins in the upper purple triangle, whereas AF2 outperforms RGN2 for targets in the lower green triangle. XRD, X-ray crystallography.

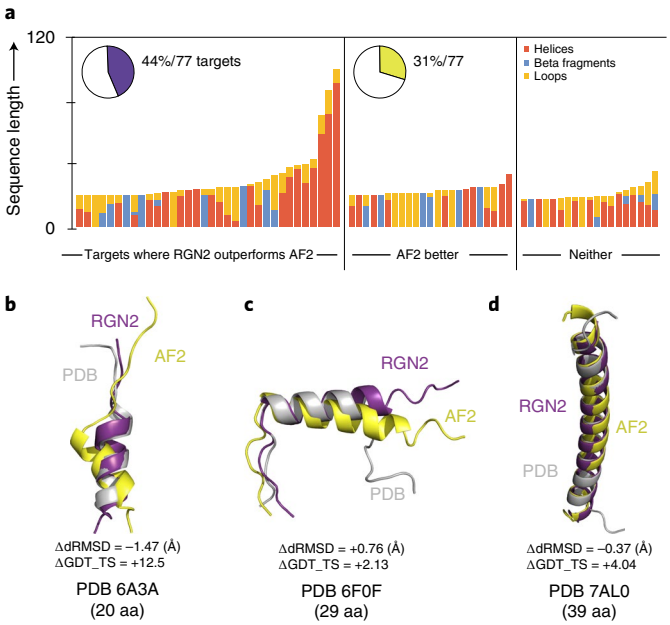

Fig. 3 | Comparing RGN2 and AF2 structure predictions for orphan proteins. a, Stacked bar charts show the relative fractions of secondary structure elements in orphan proteins broken down by these categories: RGN2 outperforms AF2; AF2 outperforms RGN2; and there is no clear winner. Bar height indicates protein length. **b**–**d**, Alpha-helical targets of different lengths (6A3A, 6F0F and 7AL0) that contain bends or hydrogen-bonded turns between helical domains tend to be better predicted by RGN2 than AF2. aa, amino acid.

(1D) curves[38]; similar ideas were considered in ref. [39]. In this parameterization, each residue is represented by its C_α atom and an oriented reference frame centered on that atom. Local residue geometry was described by a single rotation matrix relating the preceding frame to the current one, which is the geometrical object that RGN2 predicts at each residue position. This rotationally and translationally invariant parameterization has two advantages over our previous use of torsion angles in RGN1. First, it ensured that specifying a single biophysical parameter, namely the sequential C_α–C_α distance of ~3.8 Å (which corresponds to a *trans* conformation), results in only physically realizable local geometries. This overcomes a limitation of RGN1, which yielded chemically unrealistic values for some torsion angles. Second, it reduced by ~10-fold the computational cost of chain extension calculations, which often dominates RGN training and inference times (Methods).

RGN2 training was performed using both the ProteinNet12 dataset[40] and a smaller dataset comprised solely of single protein domains derived from the ASTRAL SCOPe dataset (version 1.75)[41]. Because we observed no detectable difference between the two, all results in this paper derive from the smaller dataset, as it required less training time.

Predicting structures of proteins with no homologs. To assess how well RGN2 predicts the structures of orphan proteins having no known sequence homologs (Supplementary Figs. 2 and 3), we compared it to AF2 (ref. [8]) and RF[27], currently the best publicly available methods. In addition to UniRef30, we used two other complementary databases (PDB70 and MGnify) to prepare a list of 77 proteins with the following properties: (1) they are at least 20 residues long; (2) they are orphans (that is, MSA depth = 1) across all three datasets simultaneously; and (3) they have solved structures in the Protein Data Bank (PDB)[42] (see Methods for orphan test set construction details). We note that more than 85% of these

Table 1 | A, A quantitative comparison of average TM scores and precision of top L/2, L/5 and L/10 contacts and contacts within alpha-helical and beta-type folds across 77 orphan proteins and 149 de novo proteins, using ESM-1b and RGN2. B, Comparison of prediction times among RGN2 and AF2, RF and trRosetta across 330 targets spanning our orphan and de novo protein datasets. RGN2 predictions were performed in batches with maximum permissible batch size set to 128 targets. The trRosetta MSA generation step was not used because none of the targets had known homologous proteins

A

Targets	Method	Top L/x			Structural Classes	
		L/2	L/5	L/10	alpha-helix	beta-type
77 orphans	RGN2	29.3	52.3	64.4	86.5	20.3
	ESM-1b	30.1	51.8	69.6	84.1	49.5
149 de novo	RGN2	35.6	55.1	61.8	87.9	23.3
	ESM-1b	29.3	54.5	68.4	84.1	39.2

B

Protein length (*L*) bins (no. residues)	Total, targets	Mean protein length (no. residues)	Mean trRosetta prediction time per structure (s)		Mean AF2 prediction time (s)	Mean RF prediction time (s)	Mean RGN2 prediction time (ms)	Mean RGN2 prediction + refinement time (s)
			Distogram	3D structure				
$0 < L \leq 100$	184	37.5	1,768	1,004	831.5	412.6	2.7	132.99
$100 < L \leq 200$	93	148.7	2,791	1,927	851.6	408.3	2.2	139.89
$200 < L \leq 300$	28	258.3	2,877	1,752	828.4	492.7	3.1	154.34
$300 < L \leq 400$	17	333.4	3,647	2,140	825.6	501.6	5.7	179.52
$400 > L$	8	460.5	4,012	3,011	841.6	498.6	5.9	192.53

sequences are included in the training sets for AF2 and RF, which may result in an overestimate of the accuracy of these methods. We predicted the structures of orphan proteins using all methods and assessed accuracy with respect to experimentally determined structures (Fig. 2a) using GDT_TS (which roughly captures the fraction of the structure that is correctly predicted) and dRMSD. We found that RGN2 outperformed AF2 and RF on both metrics in 44% and 65% of cases, respectively (these correspond to the top-left quadrant in Fig. 2b). In 31% and 20% of cases, AF2 and RF outperformed RGN2 on both metrics, respectively; split results were obtained in the remaining cases. When we computed differences in error metrics obtained for different prediction methods, we found that RGN2 outperformed AF2 and RF by an average ΔdRMSD of 0.65 Å and 1.99 Å and ΔGDT_TS of 5.34 and 12.4 units, respectively. When the same analysis was applied only to structures that had been determined by X-ray crystallography or electron microscopy (that is, 80% of the targets shown in light gray in Fig. 2b), RGN2 exhibited a similar improvement over AF2 and RF: an average ΔdRMSD of 0.66 Å and 2.25 Å and ΔGDT_TS of 5.56 and 13.5 units, respectively.

To investigate the structural basis for these differences in performance, we applied the DSSP algorithm[43] to determine the fraction of each secondary structure element (helical-alpha, 5 and 3/10; beta-strand; bridge; and unstructured loops, bends and hydrogen-bonded turns) in PDB structures for the orphan protein test set (Fig. 3a). We found that RGN2 outperformed all other methods on proteins rich in single helices and bends or hydrogen-bonded turns interspersed with helices, whereas other methods—AF2 in particular—better predicted targets with high fractions of beta-strand and beta-bridges (such as hairpins). Performance on the remaining ~25% targets was split between RGN2 and competing methods (Fig. 2b). We also examined performance as a function of protein length and found that RGN2 generally outperformed AF2 on longer helical proteins. One possible explanation for these findings is that the Frenet–Serret geometry used by RGN2 is based on two local parameters (curvature and torsion), and these parameters have fixed values for helices. Thus, RGN2 has an intrinsic ability to learn helical patterns.

In Fig. 3b–d, we show examples of structures for which RGN2 outperformed AF2. For example, PDB structures 6A3A (Fig. 3b) and 7AL0 (Fig. 3d) are largely alpha-helical, where the helices comprise 75% and 94% of the structure, respectively. RGN2 correctly predicts the challenging, less-structured bends and turns in these proteins, yielding 12.5-point and 4.04-point gains in GDT_TS and ΔdRMSD > 1.47 Å and 0.37 Å over AF2, respectively. A protein for which neither model succeeded across both metrics, 6F0F, has an alpha-helical base with loops at both ends (Fig. 3c). RGN2 accurately predicted the majority of the helical domain of 6F0F but failed to fully capture the loop regions; in contrast, AF2 deviates more from the main helix but more closely follows the direction of the ending loop. For RGN2, this contributed to an 2.13-point increase in GDT_TS and a 0.76-Å increase in dRMSD, respectively.

Predicting the structures of de novo (designed) proteins. We evaluated the accuracy of RGN2 on a test set of 149 synthetic proteins that were originally designed de novo using computationally parametrized energy functions, such as Rosetta and Amber; these proteins are expected to be well-suited to prediction by RF. Many of these proteins are intended to have applications in therapeutic development, such as novel antimicrobial peptides. This test set comprises all known designed proteins that are not part of the AF2 training set, as ascertained by PDB deposition date and filtered to have an 'organism' annotation of 'synthetic construct'. This filter helps to eliminate ambiguous de novo protein entries (for example, 7NBI), which are synthesized single-point mutants of known proteins. As before, we assessed prediction accuracy using dRMSD and GDT_TS. We found that RGN2 outperformed AF2 and RF on both metrics in 47% and 66% of cases, respectively (Fig. 4). On average, RGN2 outperformed AF2 and RF on these targets, with dRMSD and GDT_TS gains of 12.4 Å and 17.1

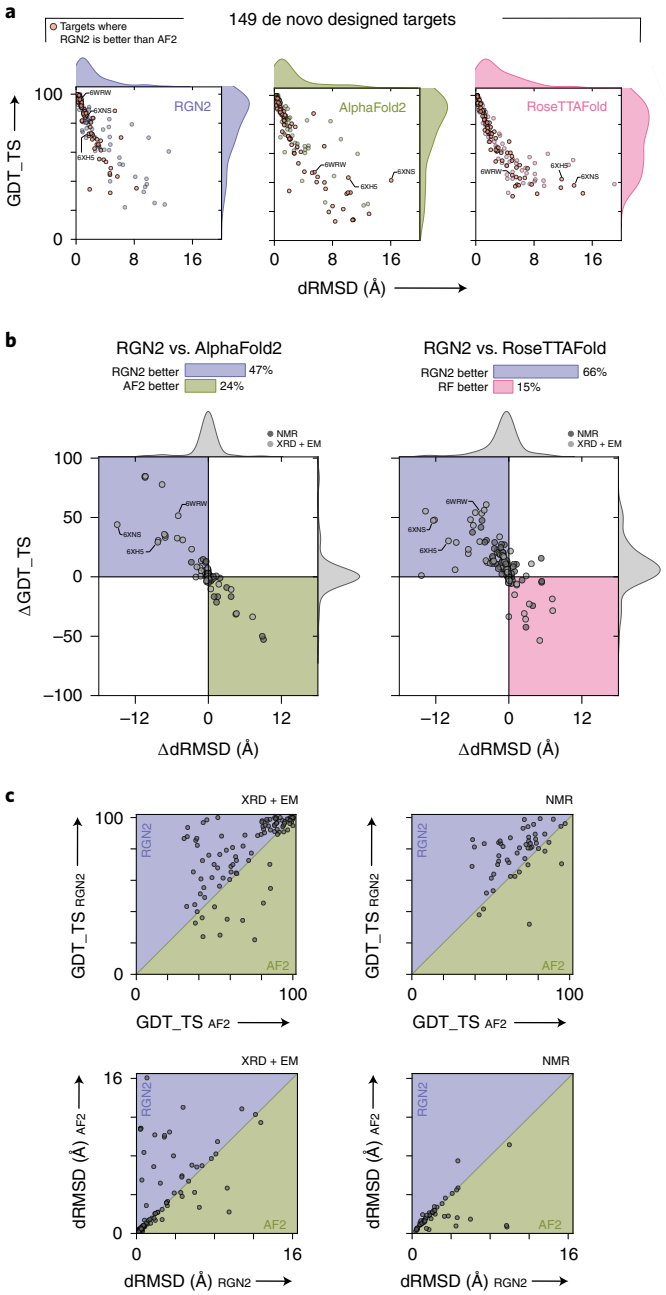

Fig. 4 | Prediction performance on designed proteins. a, Absolute performance metrics for RGN2 (purple), AF2 (green) and RF (pink) across 149 de novo designed proteins. **b**, Differences in prediction accuracy between RGN2 and AF2 are shown for these 149 proteins using dRMSD and GDT_TS as metrics. Points in the top-left quadrant correspond to targets with negative ΔdRMSD and positive ΔGDT_TS—that is, where RGN2 outperforms the competing method on both metrics, and vice versa for the bottom-right quadrant. The other two quadrants (white) indicate targets where there is no clear winner, as the two metrics disagree. The structures of 34% of the targets were determined experimentally using NMR and are denoted with dark gray markers, whereas the remaining 66% of targets were determined using X-ray crystallography or electron microscopy. **c**, Head-to-head comparisons of absolute GDT_TS and dRMSD scores for RGN2 and AF2 are shown broken down by experimental method (NMR and X-ray crystallography/electron microscopy). RGN2 outperforms AF2 for proteins in the upper purple triangle, whereas AF2 outperforms RGN2 for targets in the lower green triangle. XRD, X-ray crystallography.

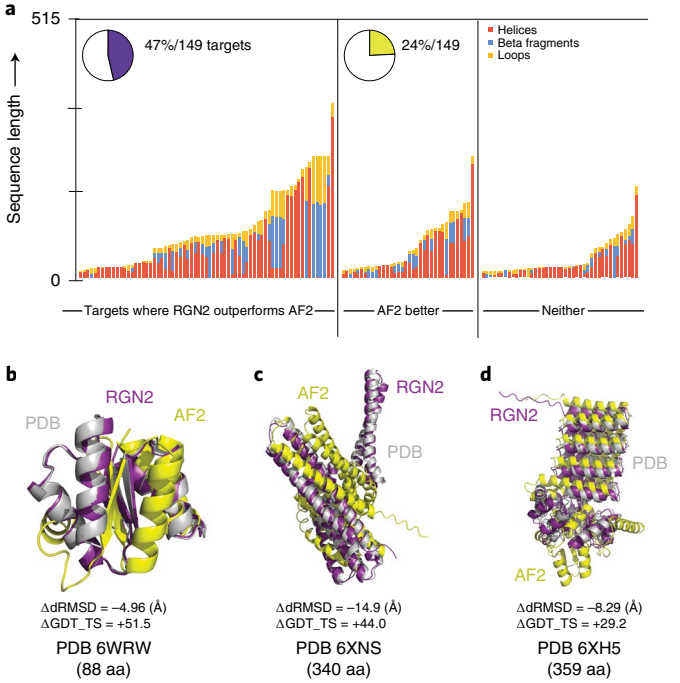

Fig. 5 | Comparing RGN2 and AF2 structure predictions for designed proteins. a, Stacked bar chart shows 149 de novo designed proteins. Bar height indicates protein length. **b**, Overlaid ribbon diagrams of PDB entries (with increasing protein length) 6WRW, 6XNS and 6XH5 (white), and RGN2 (purple) and AF2 (yellow) predicted structures are visually depicted to show how RGN2 outperforms AF2 for each of these cases. aa, amino acid.

and of 1.80 Å and 2.33, respectively (Fig. 4). The same analysis applied only to structures determined by X-ray crystallography or electron microscopy (that is, 66% of the targets shown in light gray in Fig. 4b) yielded similar improvements in RGN2 relative to AF2 and RF: an average ΔdRMSD of 2.38 Å and 2.65 Å and ΔGDT_TS of 15.5 and 16.9 units, respectively. We conclude that RGN2 can better predict sequence–structure relationships for helical regions of de novo protein space than all competing methods (Fig. 5) but that beta-sheet prediction from single sequences remains a challenge.

As an illustration of how RGN2 improves on and complements AF2 predictions, we show in Fig. 5c the PDB structure 6XNS. Similarly to orphans with largely helical secondary structural composition, RGN2 predicts this target more accurately than AF2 (ΔdRMSD = –14.9 Å and GDT_TS = +44.0 ΔGDT_TS). In Fig. 5b,d, we show predicted structures of two different alpha/beta targets (PDB accession codes 6WRW and 6XH5). For these targets, RGN2 more accurately captures the ordered secondary structured elements and hydrogen-bonded turns, resulting in 51.5-point and 29.2-point gains in GDT_TS and ΔdRMSD > 4.96 Å and 8.29 Å over AF2, respectively. Similar observations suggest that future hybrid methods using both a language model and MSAs may outperform either method alone.

Contact prediction precision. We performed a comparative contact prediction analysis between RGN2 and ESM-1b first on our revised set of 124 de novo protein targets (that is, those >20 amino acids long with no homolog across the PDB70, MGnify and UniRef90 datasets) and our set of designed proteins (Table 1A). These tables show the percentage precision of top L/2, L/5 and L/10 contacts. We note that ESM-1b outperforms RGN2 on the beta-rich contacts, but, for alpha-rich contacts, RGN2 remains marginally ahead. We

note that gains in contact prediction accuracy do not necessarily translate to improved tertiary structure prediction[44].

RGN2 prediction speed. Rapid prediction of protein structure is essential for tasks such as protein design and analysis of allelic variation or disease mutations. By virtue of being end-to-end differentiable, RGN2 predicts unrefined structures using fast neural network operations and does not require physics-based conformational sampling to assemble a folded chain. Because it operates directly on single sequences, RGN2 also avoids expensive MSA calculations. To quantify these benefits, we compared the speed of RGN2 and other methods on orphan and de novo protein datasets of varying lengths (breaking down computation time by prediction stage; Table 1B). In MSA-based methods, MSA generation scaled linearly with MSA depth (that is, the number of homologous sequences used), whereas distogram prediction (by trRosetta) scaled quadratically with protein length. AF2 predictions scale cubically with protein length. In contrast, RGN2 scales linearly with protein length, and both template-free and MSA-free implementations of AF2 and RF were $>10^5$-fold slower than RGN2. In the absence of post-prediction refinement, RGN2 is up to 10^6-fold faster, even for relatively short proteins. Adding physics-based refinement increased compute cost for all methods, but, even so, RGN2 remains the fastest available method. Of interest, even when MSA generation is discounted, neural network–based inference for AF2 and RF remains much slower than RGN2, inclusive of post-prediction refinement. This gap will only widen for design tasks involving longer proteins, whose chemical synthesis is increasingly becoming feasible[45]. Thus, fast prediction is a benefit of using a protein language model such as AminoBERT.

Discussion

RGN2 represents one of the first attempts to use ML to predict protein structure from a single sequence. This is computationally efficient and has many advantages in the case of orphan and designed proteins for which generation of multiple sequence alignment is often not possible. RGN2 accomplishes this by fusing a protein language model (AminoBERT) with a simple and intuitive approach to parameterizing C_α backbone geometry based on the Frenet–Serret formulas. Whereas most recent advances in ML-based structure prediction have relied on MSAs[5] to learn latent information about folding, AminoBERT learns this information from proteins without alignment. Training in this case involves sequences with masked residues and block permutations. We speculate that the latent space of the language model also captures recurrent evolutionary relationships[46]. The use of Frenet–Serret formulas in RGN2 addresses the requirement that proteins exhibit translational and rotational invariance. From a practical standpoint, the speed and accuracy of RGN2 shows that language models are effective at learning structural information from primary sequence while having the ability to extrapolate beyond known proteins, thereby enabling effective prediction of orphan and designed proteins. Nonetheless, methods that use MSA information (when it is available) often outperform RGN2, most notably AF2 when assessed on proteins in the 'Free Modeling' category of CASP14 (Supplementary Fig. 4). Thus, language models are not a substitute for MSAs but, rather, a complementary way to get at the latent rules governing protein folding. We speculate that folding systems that use both language models and MSAs will be more performative than systems using one approach alone.

Transformers and their embodiment of local and distant attention is a key feature of language models such as AminoBERT. Very large transformer-based models trained on hundreds of millions and potentially billions of protein sequences are increasingly available[26,35,36], and the scaling previously observed in natural language applications[47] makes it likely that the performance of RGN2 and similar methods will continue to improve and become broadly performative over intrinsically disordered proteins and cyclic peptides as well (Supplementary Fig. 5). AF2 also exploits attention mechanisms based on transformers to capture the latent information in MSAs. Similarly, the self-supervised MSA transformer[48] uses a related attention strategy that attends to both positions and sequences in an MSA and achieves state-of-the-art contact prediction accuracy. Architectures merging language models and MSAs are also likely to benefit from augmentation from high-confidence structures found in the AlphaFold Database[8]. Finally, training on experimental data is almost certain to be invaluable in selected applications requiring high accuracy within members of multi-protein families, such as predicting structural variation within kinases or G protein–coupled receptors.

We consider RGN2 to be a first step in the development of methods able to compute sequence-to-structure maps without a requirement for explicit evolutionary information. One limitation of the current RGN2 implementation to be addressed by future systems is that the immediate output of the recurrent geometric network only constrains local dependencies between C_α atoms (curvature and torsion angles), resulting in sequential reconstruction of backbone geometry. Allowing the network to reason directly on arbitrary pairwise dependencies throughout the structure, and using a better inductive prior than immediate contact, may further improve the quality of model predictions. A second limitation is that refinement in RGN2 is not part of an end-to-end implementation; refinement via a 3D rotationally and translationally equivariant neural network would be more efficient and likely yield better-quality structures. Currently, AF2Rank-based refinement results in an average increase of 24.7 for GDT_TS scores and an average decrease of 2.3 Å for dRMSD values, relative to predictions from RGN2 alone, as evaluated using all 225 orphan and de novo targets described in this study (Supplementary Fig. 6).

It has been known since Anfinsen's refolding experiments that single polypeptide chains contain the information needed to specify fold[49]. The demonstration that a language model can learn information on structure directly from protein sequences and then guide accurate prediction of an unaligned protein suggests that RGN2 behaves in a manner that is more similar to the physical process of protein folding than MSA-based methods. Transformers can learn structural encodings present in both local and distant features of a sequence, which is reflective of the role played by local residues in the molten globule stage and distant residues in the 3D protein fold. Moreover, language models learned by deep neural networks are readily formulated in a maximum entropy framework[50]. The physical process of protein folding is also entropically driven, potentially suggesting a means to compare the two. A fusion of biophysical and learning-based perspectives may ultimately prove the key to direct sequence-to-structure prediction from single polypeptides at experimental accuracy and for understanding folding energetics and dynamics.

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/s41587-022-01432-w.

Received: 28 September 2021; Accepted: 15 July 2022;
Published online: 03 October 2022

References

1. Yang, J. & Zhang, Y. I-TASSER server: new development for protein structure and function predictions. *Nucleic Acids Res.* **43**, W174–W181 (2015).
2. Wang, J., Wang, W., Kollman, P. A. & Case, D. A. Automatic atom type and bond type perception in molecular mechanical calculations. *J. Mol. Graph. Model.* **25**, 247–260 (2006).

3. Hess, B., Kutzner, C., Van Der Spoel, D. & Lindahl, E. GRGMACS 4: algorithms for highly efficient, load-balanced, and scalable molecular simulation. *J. Chem. Theory Comput.* **4**, 435–447 (2008).
4. Alford, R. F. et al. The Rosetta all-atom energy function for macromolecular modeling and design. *J. Chem. Theory Comput.* **13**, 3031–3048 (2017).
5. AlQuraishi, M. Machine learning in protein structure prediction. *Curr. Opin. Chem. Biol.* **65**, 1–8 (2021).
6. Senior, A. W. et al. Improved protein structure prediction using potentials from deep learning. *Nature* **577**, 706–710 (2020).
7. Yang, J. et al. Improved protein structure prediction using predicted interresidue orientations. *Proc. Natl Acad. Sci. USA* **117**, 1496–1503 (2020).
8. Jumper, J. et al. Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583–589 (2021).
9. Pearson, W. R. An introduction to sequence similarity ('homology') searching. Curr. Protoc. Bioinformatics **Chapter 3**, Unit3.1 (2013).
10. Perdigão, N. et al. Unexpected features of the dark proteome. *Proc. Natl Acad. Sci. USA* **112**, 15898–15903 (2015).
11. Price, N. D. et al. A wellness study of 108 individuals using personal, dense, dynamic data clouds. *Nat. Biotechnol.* **35**, 747–756 (2017).
12. Stittrich, A. B. et al. Genomic architecture of inflammatory bowel disease in five families with multiple affected individuals. *Hum. Genome Var.* **3**, 15060 (2016).
13. Huang, X., Pearce, R. & Zhang, Y. EvoEF2: accurate and fast energy function for computational protein design. *Bioinformatics* **36**, 1135–1142 (2020).
14. Jiang, L. et al. De novo computational design of retro-aldol enzymes. *Science* **319**, 1387–1391 (2008).
15. Renata, H., Wang, Z. J. & Arnold, F. H. Expanding the enzyme universe: accessing non-natural reactions by mechanism-guided directed evolution. *Angew. Chem. Int. Ed. Engl.* **54**, 3351–3367 (2015).
16. Richter, F., Leaver-Fay, A., Khare, S. D., Bjelic, S. & Baker, D. De novo enzyme design using Rosetta3. *PLoS ONE* **6**, e19230 (2011).
17. Steiner, K. & Schwab, H. Recent advances in rational approaches for enzyme engineering. *Comput. Struct. Biotechnol. J.* **2**, e201209010 (2012).
18. Sáez-Jiménez, V. et al. Improving the pH-stability of versatile peroxidase by comparative structural analysis with a naturally-stable manganese peroxidase. *PLoS ONE* **10**, e0140984 (2015).
19. Park, H. J., Joo, J. C., Park, K., Kim, Y. H. & Yoo, Y. J. Prediction of the solvent affecting site and the computational design of stable *Candida antarctica* lipase B in a hydrophilic organic solvent. *J. Biotechnol.* **163**, 346–352 (2013).
20. Jiang, C. et al. An orphan protein of *Fusarium graminearum* modulates host immunity by mediating proteasomal degradation of TaSnRK1α. *Nat. Commun.* **11**, 4382 (2020).
21. Tautz, D. & Domazet-Lošo, T. The evolutionary origin of orphan genes. *Nat. Rev. Genet.* **12**, 692–702 (2011).
22. AlQuraishi, M. End-to-end differentiable learning of protein structure. *Cell Syst.* **8**, 292–301 (2019).
23. Ingraham, J., Riesselman, A., Sander, C. & Marks, D. Learning protein structure with a differentiable simulator. in *7th International Conference on Learning Representations.* https://openreview.net/forum?id=Byg3y3C9Km (2019).
24. Li, J. Universal transforming geometric network. Preprint at https://arxiv.org/abs/1908.00723 (2019).
25. Kandathil, S. M., Greener, J. G., Lau, A. M. & Jones, D. T. Ultrafast end-to-end protein structure prediction enables high-throughput exploration of uncharacterised proteins. *Proc. Natl Acad. Sci. USA* **119**, e2113348119 (2022).
26. Rives, A. et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl Acad. Sci. USA* **118**, e2016239118 (2021).
27. Baek, M. et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science* **10**, eabj8754 (2021).
28. Roney, J. P. & Ovchinnikov, S. State-of-the-art estimation of protein model accuracy using AlphaFold. Preprint at https://www.biorxiv.org/content/10.1101/2022.03.11.484043v3 (2022).
29. Devlin, J., Chang, M. W., Lee, K. & Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.* **1**, 4171–4186. https://aclanthology.org/N19-1423/ (2019).
30. Vaswani, A. et al. Attention is all you need. *Adv. Neural Inf. Proc. Syst.* **30**, (2017).
31. Leinonen, R. et al. UniProt archive. *Bioinformatics* **20**, 3236–3237 (2004).
32. Meier, J. et al. Language models enable zero-shot prediction of the effects of mutations on protein function. *Adv. Neural Inf. Process. Syst.* **34**, 29287–29303 (2021).
33. Elnaggar, A. et al. CodeTrans: towards cracking the language of silicone's code through self-supervised deep learning and high performance computing. Preprint at https://arxiv.org/abs/2104.02443 (2021).
34. Alley, E., Khimulya, G., Biswas, S., AlQuraishi, M. & Church, G. Unified rational protein engineering with sequence-only deep representation learning. *Nat. Methods* **16**, 1315–1322 (2019).
35. Heinzinger, M. et al. Modeling the language of life—deep learning protein sequences. Preprint at https://www.biorxiv.org/content/10.1101/614313v1 (2019).
36. Madani, A. et al. ProGen: language modeling for protein generation. Preprint at https://arxiv.org/abs/2004.03497 (2020).
37. Elnaggar, A. et al. ProtTrans: towards cracking the language of life's code through self-supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.* https://doi.org/10.1109/TPAMI.2021.3095381 (2021).
38. Hu, S., Lundgren, M. & Niemi, A. J. Discrete Frenet frame, inflection point solitons, and curve visualization with applications to folded proteins. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.* **83**, 061908 (2011).
39. Penner, R. C., Knudsen, M., Wiuf, C. & Andersen, J. E. Fatgraph models of proteins. *Commun. Pure Appl. Math.* **63**, 1249–1297 (2010).
40. AlQuraishi, M. ProteinNet: a standardized data set for machine learning of protein structure. *BMC Bioinformatics* **20**, 311 (2019).
41. Fox, N. K., Brenner, S. E. & Chandonia, J. M. SCOPe: Structural Classification of Proteins—extended, integrating SCOP and ASTRAL data and classification of new structures. *Nucleic Acids Res.* **42**, D304–D309 (2014).
42. Burley, S. K. et al. RCSB Protein Data Bank: powerful new tools for exploring 3D structures of biological macromolecules for basic and applied research and education in fundamental biology, biomedicine, biotechnology, bioengineering and energy sciences. *Nucleic Acids Res.* **49**, D437–D451 (2021).
43. Touw, W. G. et al. A series of PDB-related databanks for everyday needs. *Nucleic Acids Res.* **43**, D364–D368 (2015).
44. Outeiral, C., Nissley, D. A. & Deane, C. M. Current structure predictors are not learning the physics of protein folding. *Bioinformatics* **38**, 1881–1887 (2022).
45. Hartrampf, N. et al. Synthesis of proteins by automated flow chemistry. *Science* **368**, 980–987 (2020).
46. Rao, R., Meier, J., Sercu, T., Ovchinnikov, S. & Rives, A. Transformer protein language models are unsupervised structure learners. Preprint at https://www.biorxiv.org/content/10.1101/2020.12.15.422761v1 (2020).
47. Kaplan, J. et al. Scaling laws for neural language models. Preprint at https://arxiv.org/abs/2001.08361 (2020).
48. Rao, R. et al. MSA Transformer. *Proceedings of the 38th International Conference on Machine Learning, PMLR* **139**, 8844–8856 (2021).
49. Anfinsen, C. B., Haber, E., Sela, M. & White, F. H. The kinetics of formation of native ribonuclease during oxidation of the reduced polypeptide chain. *Proc. Natl Acad. Sci. USA* **47**, 1309–1314 (1961).
50. Mikolov, T. et al. Strategies for training large scale neural network language models. *2011 IEEE Workshop on Automatic Speech Recognition & Understanding.* 196–211. https://doi.org/10.1109/ASRU.2011.6163930 (2011).

Methods

AminoBERT summary. AminoBERT is a 12-layer transformer where each layer is composed of 12 attention heads. It is trained to distill protein sequence semantics from ~260 million natural protein sequences obtained from the UniParc sequence database[31] (downloaded on 19 May 2019).

During training, each sequence is fed to AminoBERT according to the following algorithm:

1. With probability 0.3, select sequence for chunk permutation, and, with probability 0.7, select sequence for masked language modeling.
2. If sequence was selected for chunk permutation, then: with probability 0.35, chunk permute, or else (with probability 0.65) leave the sequence unmodified.
3. Else if the sequence was selected for masked language modeling, then: with probability 0.3, introduce 0.15 × sequence_length masks into the sequence with clumping, or else (with probability 0.7) introduce the same number of masks into the sequence randomly across the length of the sequence (standard masked language modeling).

The loss for an individual sequence (seq) is given by:

$$Loss\,(seq) = I\,[seq\ is\ chunk\ permuted] \times chunk\_permutation\_loss(seq) + (1 - I[seq\ is\ globally\ perturbed] \times masked\_lm\_loss(seq)$$

where I[x] is the indicator of the event x and returns 1 if x is true and 0 if x is false. Chunk_permutation_loss(seq) is a standard cross-entropy loss reflecting the classification accuracy of predicting whether seq has been chunk permuted. Finally, masked_lm_loss(seq) is the standard masked language modeling loss as previously described in Devlin et al.[29]. Note that mask clumping does not affect how the loss is calculated.

Chunk permutation is performed by first sampling an integer x uniformly between 2 and 10, inclusively. The sequence is then randomly split into x equal-sized fragments, which are subsequently shuffled and rejoined.

Mask clumping is performed as follows:

1. Sample an integer clump_size ~ Poisson (2.5) + 1
2. Let n_mask = 0.15 × sequence_length. Randomly select n_mask/clump_size positions in the sequence around which to introduce a set of clump_size contiguous masks

AminoBERT architecture. Each multi-headed attention layer in AminoBERT contains 12 attention heads, each with hidden size 768. The output dimension of the feed-forward unit at the end of each attention layer is 3,072. As done in BERT[29], we prepend a [CLS] token at the beginning of each sequence, for which an encoding is maintained through all layers of the AminoBERT transformer. Each sequence was padded or otherwise clipped to length 1,024 (including the [CLS] token).

For chunk permutation classification, the final hidden vector of the [CLS] token is fed through another feed-forward layer of output dimension 768, followed by a final feed-forward layer of output dimension 2, which are the logits corresponding to whether the sequence is chunk permuted or not. Masked language modeling loss calculations are set up as described in Devlin et al.[29].

AminoBERT training procedure. AminoBERT was trained with batch size 3,072 for 1,100,000 steps, which is approximately 13 epochs over the 260 million sequence corpus. For our optimizer, we used Adam with a learning rate of 1×10^{-4}, $\beta 1 = 0.9$, $\beta 2 = 0.999$, epsilon $= 1 \times 10^{-6}$, L2 weight decay of 0.01, learning rate warmup over the first 20,000 steps and linear decay of the learning rate. We used a dropout probability of 0.1 on all layers and used GELU activations as done for BERT. Training was performed on a 512-core TPU pod for approximately 1 week.

Geometry module. The geometry of the protein backbone as summarized by the C_α trace can be thought of as a 1D discrete open curve, characterized by a bond and torsion angle at each residue. Following Niemi et al.[38], the starting point for describing such discrete curves is to assign a frame, a triplet of orthonormal vectors, to each C_α atom. If we denote by r_i the vector characterizing the position of a C_α atom at the i-th vertex, we could then define a unit tangent vector along an edge connecting two consecutive C_α atoms

$$t_i = \frac{r_{i+1} - r_i}{|r_{i+1} - r_i|}$$

For assigning frames to each i-th C_α atom, we need two extra vectors, the binormal and normal vectors defined as follows:

$$b_i = \frac{t_{i-1} \times t_i}{|t_{i-1} \times t_i|}$$

$$n_i = b_i \times t_i$$

Although, for a protein (in a given orientation), the tangent vector is uniquely defined, the normal and binormal vectors are arbitrary. Indeed, when assigning frames to each residue, we could take any arbitrary orthogonal basis on the normal plane to the tangent vector. Such arbitrariness does not affect our strategy of predicting 3D structures starting from bond and torsion angles.

To derive the equivalent of the Frenet–Serret formulas—which describe the geometry of continuous and differentiable 1D curves—for the discrete case, we need to relate two consecutive frames along the protein backbone in terms of rotation matrices:

$$\begin{pmatrix} n_{i+1} \\ b_{i+1} \\ t_{i+1} \end{pmatrix} = \mathcal{R}_{i+1,i} \begin{pmatrix} n_i \\ b_i \\ t_i \end{pmatrix}$$

In three dimensions, rotation matrices are, in general, parametrized in terms of three Euler angles. However, in our case, the rotation matrices relating two consecutive frames are fully characterized by only two angles, a bond angle ψ and a torsion angle θ, as the third Euler angle vanishes, reflecting the following condition $b_{i+1}.t_i = 0$. We can now write the equivalent of the Frenet–Serret formulas for the discrete case:

$$\begin{pmatrix} n_{i+1} \\ b_{i+1} \\ t_{i+1} \end{pmatrix} = \begin{pmatrix} \cos\psi\ \cos\theta & \cos\psi \sin\theta & -\sin\psi \\ -\sin\theta & \cos\theta & 0 \\ \sin\psi \cos\theta & \sin\psi \sin\theta & \cos\psi \end{pmatrix} \begin{pmatrix} n_i \\ b_i \\ t_i \end{pmatrix}$$

The bond and torsion angles are defined by the following relations:

$$\cos\psi_{i+1,i} = t_{i+1}.t_i$$

$$\cos\theta_{i+1,i} = b_{i+1}.b_i$$

We now turn to backbone reconstruction starting from bond and torsion angles. First, using tangent vectors along the backbone edges, we can reconstruct all C_α atom positions, and, thus, the full protein backbone in the C_α trace, by using the following relation:

$$r_k = \sum_{i=0}^{k-1} |r_{i+1} - r_i|\,.t_i$$

where $|r_{i+1} - r_i|$ is the length of the virtual bonds connecting two consecutive C_α atoms. In most cases, the average virtual bond length is ~3.8 Å, which corresponds to *trans* conformations. In terms of the familiar torsion angles ϕ, ψ and ω, those conformations are achieved for $\omega \sim \pi$. For *cis* conformations, mainly involving proline residues, the virtual bond length is ~3.0 Å (and it corresponds to $\omega \sim 0$). In RGN2, for backbone reconstruction, we impose the condition that the virtual bond length is strictly equal to 3.8 Å, and, for reconstructing the backbone, we use the following relation:

$$r_k = \sum_{i=0}^{k-1} 3.8 \times t_i$$

The intuition behind the previous equation is the idea of a moving observer along the protein backbone. We could think of the tangent vector t_i as the velocity of the observer along a given edge and the constant virtual bond length as the effective time spent for travelling along the edge. The only freedom allowed for such observer is to abruptly change the direction of the velocity vector at each vertex.

The model outputs bond and torsion angles. By centering the first C_α atom of the protein backbone at the origin of our coordinate system, we sequentially reconstruct all the C_α atom coordinates using the following relation:

$$\begin{pmatrix} n_{i+1} \\ b_{i+1} \\ t_{i+1} \\ r_{i+1} \end{pmatrix} = \begin{pmatrix} & & & 0 \\ & \mathcal{R}_{i+1,i} & & 0 \\ & & & 0 \\ 0 & 0 & 3.8 & 1 \end{pmatrix} \begin{pmatrix} n_i \\ b_i \\ t_i \\ r_i \end{pmatrix}$$

Data preparation for comparison with trRosetta. Performance of RGN2 was compared against trRosetta across two sets of non-homologous proteins: (1) 129 orphans from the Uniclust30 database[51] and (2) 35 de novo proteins by Xu et al.[52]. Both sets were filtered to ensure no overlap with the training sets of RGN2 and trRosetta. Whereas RGN2 is trained on the ASTRAL SCOPe (version 1.75) dataset[41], trRosetta was trained on a set of 15,051 single-chain proteins (released before 1 May 2018).

Structure prediction with trRosetta, AF2 and RF. Conventional trRosetta-based structure prediction involves first feeding the input sequence through a deep MSA generation step. For orphans and de novo proteins without any sequence

homologs, the MSA includes only the original query sequence. Next, the MSA is used by the trRosetta neural network to predict a distogram (and orientogram) that captures inter-residue (C_α–C_α and C_β–C_β) distances and orientations. This information is subsequently used by a final Rosetta-based refinement module. This module first threads a naive sequence of polyalanines of length equaling the target protein that maximally obeys the distance and orientation constraints. After sidechain imputation that reflects the original sequence, multiple steps, including clash elimination, rotamer repacking and energy minimization, are performed to identify the lowest energy structure.

AF2 and RF predictions were performed without MSAs for both our orphan and de novo target proteins. Our predictions were made using their respective official Google Colab notebooks, and all AF2 predictions were obtained from Model 1.

Structure refinement in RGN2. Raw predictions from RGN2 contain a single C_α trace of the target protein. The remaining backbone and sidechain atoms are initially constructed using the unrefined full-atom model generated by ModRefiner[53]. This structure is then fine-tuned based on the methods described in AF2Rank[28], where the RGN2-predicted structure is supplied as a template to AlphaFold2 with no additional coevolutionary information. The sequence associated with the template is replaced with 'gap' tokens, and the input structure is modified such that C_β atoms are added to all glycine residues and all side chain atoms except C_β are masked. The target sequence and RGN2 template are passed to AlphaFold2 for one recycling iteration to obtain the final predicted structure[54].

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

Data availability

The AminoBERT module was trained using the UniParc sequence database (https://www.uniprot.org/help/uniparc). Homologous sequence searches to determine orphan sequences were performed across UniRef90 (https://ftp.uniprot.org/pub/databases/uniprot/uniref/uniref90/), PDB70 (http://prodata.swmed.edu/procain/info/database.html) and MGnify (https://www.ebi.ac.uk/metagenomics/) metagenomic sequence alignment datasets. The six PDB structures discussed in detail in the article (5FKP, 2KWZ, 6E5N, 2L96, 5UP5 and 7KBQ) were all sourced from the Protein Data Bank.

Code availability

RGN2 is available freely as a standalone tool from https://github.com/aqlaboratory/rgn2. Users can make structure predictions using a Python3-based web user interface by uploading the protein sequence in FASTA format (https://colab.research.google.com/github/aqlaboratory/rgn2/blob/master/rgn2_prediction.ipynb).

References

51. Mirdita, M. et al. Uniclust databases of clustered and deeply annotated protein sequences and alignments. *Nucleic Acids Res.* **45**, D170–D176 (2017).
52. Xu, J., McPartlon, M. & Li, J. Improved protein structure prediction by deep learning irrespective of co-evolution information. *Nat. Mach. Intell.* **3**, 601–609 (2021).
53. Xu, D. & Zhang, Y. Improving the physical realism and structural accuracy of protein models by a two-step atomic-level energy minimization. *Biophys. J.* **101**, 2525–2534 (2011).
54. Fleishman, S. J. et al. Rosettascripts: a scripting language interface to the Rosetta macromolecular modeling suite. *PLoS ONE* **6**, e20161 (2011).

Acknowledgements

We gratefully acknowledge the support of NVIDIA Corporation for the donation of GPUs used for this research. This work is supported by DARPA PANACEA program grant HR0011-19-2-0022 and National Cancer Institute grant U54-CA225088 to P.K.S. We also acknowledge support from the TensorFlow Research Cloud for graciously providing the TPU resources used for training AminoBERT.

Author contributions

R.C., N.B., S.B. and M.A. conceived of and designed the study. R.C. and C.F. developed the refinement module. R.C., C.F., A.K. and K.R. performed the analyses. N.B. developed the geometry module and trained RGN2 models. S.B. developed and trained the AminoBERT protein language model and helped integrate its embeddings within RGN2. C.R. trained several RGN2 models and performed RF predictions. C.F. prepared the docker image and helped package the standalone software along with a Python-based user interface (notebook) for generating RGN2 predictions. G.A. performed MSAs to identify orphans. J.Z. helped C.F. in preparation of the RGN2 prediction notebook. P.K.S. and G.M.C. supervised the research and provided funding. R.C., N.B., S.B., M.A. and P.K.S. wrote the manuscript, and all authors discussed the results and edited the final version.

Competing interests

M.A. is a member of the Scientific Advisory Board of FL2021-002, a Foresite Labs company, and consults for Interline Therapeutics. P.K.S. is a member of the Scientific Advisory Board or Board of Directors of Glencoe Software, Applied Biomath, RareCyte and NanoString and is an advisor to Merck and Montai Health. A full list of G.M.C.'s tech transfer, advisory roles, 559 and funding sources can be found on the lab's website: http://arep.med.harvard.edu/gmc/tech.html. S.B. is employed by and holds equity in Nabla Bio, Inc. The remaining authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41587-022-01432-w.

Correspondence and requests for materials should be addressed to Nazim Bouatta, Peter K. Sorger or Mohammed AlQuraishi.

Peer review information *Nature Biotechnology* thanks James Fraser and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

nature research

Corresponding author(s): Mohammed AlQuraishi
Peter K. Sorger

Last updated by author(s): July 11, 2022

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	No software was used
Data analysis	No software was used

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

RGN2 is available freely as a standalone tool from https://github.com/aqlaboratory/rgn2. Users can make structure predictions using a Python 3-based web user interface by uploading the protein sequence in fasta format ((https://colab.research.google.com/github/aqlaboratory/rgn2/blob/master/rgn2_prediction.ipynb).

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	*Describe how sample size was determined, detailing any statistical methods used to predetermine sample size OR if no sample-size calculation was performed, describe how sample sizes were chosen and provide a rationale for why these sample sizes are sufficient.*
Data exclusions	*Describe any data exclusions. If no data were excluded from the analyses, state so OR if data were excluded, describe the exclusions and the rationale behind them, indicating whether exclusion criteria were pre-established.*
Replication	*Describe the measures taken to verify the reproducibility of the experimental findings. If all attempts at replication were successful, confirm this OR if there are any findings that were not replicated or cannot be reproduced, note this and describe why.*
Randomization	*Describe how samples/organisms/participants were allocated into experimental groups. If allocation was not random, describe how covariates were controlled OR if this is not relevant to your study, explain why.*
Blinding	*Describe whether the investigators were blinded to group allocation during data collection and/or analysis. If blinding was not possible, describe why OR explain why blinding was not relevant to your study.*

Behavioural & social sciences study design

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

Study description	*Briefly describe the study type including whether data are quantitative, qualitative, or mixed-methods (e.g. qualitative cross-sectional, quantitative experimental, mixed-methods case study).*
Research sample	*State the research sample (e.g. Harvard university undergraduates, villagers in rural India) and provide relevant demographic information (e.g. age, sex) and indicate whether the sample is representative. Provide a rationale for the study sample chosen. For studies involving existing datasets, please describe the dataset and source.*
Sampling strategy	*Describe the sampling procedure (e.g. random, snowball, stratified, convenience). Describe the statistical methods that were used to predetermine sample size OR if no sample-size calculation was performed, describe how sample sizes were chosen and provide a rationale for why these sample sizes are sufficient. For qualitative data, please indicate whether data saturation was considered, and what criteria were used to decide that no further sampling was needed.*
Data collection	*Provide details about the data collection procedure, including the instruments or devices used to record the data (e.g. pen and paper, computer, eye tracker, video or audio equipment) whether anyone was present besides the participant(s) and the researcher, and whether the researcher was blind to experimental condition and/or the study hypothesis during data collection.*
Timing	*Indicate the start and stop dates of data collection. If there is a gap between collection periods, state the dates for each sample cohort.*
Data exclusions	*If no data were excluded from the analyses, state so OR if data were excluded, provide the exact number of exclusions and the rationale behind them, indicating whether exclusion criteria were pre-established.*
Non-participation	*State how many participants dropped out/declined participation and the reason(s) given OR provide response rate OR state that no participants dropped out/declined participation.*
Randomization	*If participants were not allocated into experimental groups, state so OR describe how participants were allocated to groups, and if allocation was not random, describe how covariates were controlled.*

Ecological, evolutionary & environmental sciences study design

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

Study description	*Briefly describe the study. For quantitative data include treatment factors and interactions, design structure (e.g. factorial, nested, hierarchical), nature and number of experimental units and replicates.*
Research sample	*Describe the research sample (e.g. a group of tagged Passer domesticus, all Stenocereus thurberi within Organ Pipe Cactus National*

Research sample	*Monument), and provide a rationale for the sample choice. When relevant, describe the organism taxa, source, sex, age range and any manipulations. State what population the sample is meant to represent when applicable. For studies involving existing datasets, describe the data and its source.*
Sampling strategy	*Note the sampling procedure. Describe the statistical methods that were used to predetermine sample size OR if no sample-size calculation was performed, describe how sample sizes were chosen and provide a rationale for why these sample sizes are sufficient.*
Data collection	*Describe the data collection procedure, including who recorded the data and how.*
Timing and spatial scale	*Indicate the start and stop dates of data collection, noting the frequency and periodicity of sampling and providing a rationale for these choices. If there is a gap between collection periods, state the dates for each sample cohort. Specify the spatial scale from which the data are taken*
Data exclusions	*If no data were excluded from the analyses, state so OR if data were excluded, describe the exclusions and the rationale behind them, indicating whether exclusion criteria were pre-established.*
Reproducibility	*Describe the measures taken to verify the reproducibility of experimental findings. For each experiment, note whether any attempts to repeat the experiment failed OR state that all attempts to repeat the experiment were successful.*
Randomization	*Describe how samples/organisms/participants were allocated into groups. If allocation was not random, describe how covariates were controlled. If this is not relevant to your study, explain why.*
Blinding	*Describe the extent of blinding used during data acquisition and analysis. If blinding was not possible, describe why OR explain why blinding was not relevant to your study.*

Did the study involve field work? ☐ Yes ☐ No

Field work, collection and transport

Field conditions	*Describe the study conditions for field work, providing relevant parameters (e.g. temperature, rainfall).*
Location	*State the location of the sampling or experiment, providing relevant parameters (e.g. latitude and longitude, elevation, water depth).*
Access & import/export	*Describe the efforts you have made to access habitats and to collect and import/export your samples in a responsible manner and in compliance with local, national and international laws, noting any permits that were obtained (give the name of the issuing authority, the date of issue, and any identifying information).*
Disturbance	*Describe any disturbance caused by the study and how it was minimized.*

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

Antibodies

Antibodies used	*Describe all antibodies used in the study; as applicable, provide supplier name, catalog number, clone name, and lot number.*
Validation	*Describe the validation of each primary antibody for the species and application, noting any validation statements on the manufacturer's website, relevant citations, antibody profiles in online databases, or data provided in the manuscript.*

Eukaryotic cell lines

Policy information about cell lines

Cell line source(s)	*State the source of each cell line used.*

Authentication	*Describe the authentication procedures for each cell line used OR declare that none of the cell lines used were authenticated.*
Mycoplasma contamination	*Confirm that all cell lines tested negative for mycoplasma contamination OR describe the results of the testing for mycoplasma contamination OR declare that the cell lines were not tested for mycoplasma contamination.*
Commonly misidentified lines (See ICLAC register)	*Name any commonly misidentified cell lines used in the study and provide a rationale for their use.*

Palaeontology and Archaeology

Specimen provenance	*Provide provenance information for specimens and describe permits that were obtained for the work (including the name of the issuing authority, the date of issue, and any identifying information).*
Specimen deposition	*Indicate where the specimens have been deposited to permit free access by other researchers.*
Dating methods	*If new dates are provided, describe how they were obtained (e.g. collection, storage, sample pretreatment and measurement), where they were obtained (i.e. lab name), the calibration program and the protocol for quality assurance OR state that no new dates are provided.*

☐ Tick this box to confirm that the raw and calibrated dates are available in the paper or in Supplementary Information.

Ethics oversight	*Identify the organization(s) that approved or provided guidance on the study protocol, OR state that no ethical approval or guidance was required and explain why not.*

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Animals and other organisms

Policy information about studies involving animals; ARRIVE guidelines recommended for reporting animal research

Laboratory animals	*For laboratory animals, report species, strain, sex and age OR state that the study did not involve laboratory animals.*
Wild animals	*Provide details on animals observed in or captured in the field; report species, sex and age where possible. Describe how animals were caught and transported and what happened to captive animals after the study (if killed, explain why and describe method; if released, say where and when) OR state that the study did not involve wild animals.*
Field-collected samples	*For laboratory work with field-collected samples, describe all relevant parameters such as housing, maintenance, temperature, photoperiod and end-of-experiment protocol OR state that the study did not involve samples collected from the field.*
Ethics oversight	*Identify the organization(s) that approved or provided guidance on the study protocol, OR state that no ethical approval or guidance was required and explain why not.*

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Human research participants

Policy information about studies involving human research participants

Population characteristics	*Describe the covariate-relevant population characteristics of the human research participants (e.g. age, gender, genotypic information, past and current diagnosis and treatment categories). If you filled out the behavioural & social sciences study design questions and have nothing to add here, write "See above."*
Recruitment	*Describe how participants were recruited. Outline any potential self-selection bias or other biases that may be present and how these are likely to impact results.*
Ethics oversight	*Identify the organization(s) that approved the study protocol.*

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Clinical data

Policy information about clinical studies

All manuscripts should comply with the ICMJE guidelines for publication of clinical research and a completed CONSORT checklist must be included with all submissions.

Clinical trial registration	*Provide the trial registration number from ClinicalTrials.gov or an equivalent agency.*
Study protocol	*Note where the full trial protocol can be accessed OR if not available, explain why.*
Data collection	*Describe the settings and locales of data collection, noting the time periods of recruitment and data collection.*
Outcomes	*Describe how you pre-defined primary and secondary outcome measures and how you assessed these measures.*

Dual use research of concern

Policy information about dual use research of concern

Hazards

Could the accidental, deliberate or reckless misuse of agents or technologies generated in the work, or the application of information presented in the manuscript, pose a threat to:

No	Yes	
☐	☐	Public health
☐	☐	National security
☐	☐	Crops and/or livestock
☐	☐	Ecosystems
☐	☐	Any other significant area

Experiments of concern

Does the work involve any of these experiments of concern:

No	Yes	
☐	☐	Demonstrate how to render a vaccine ineffective
☐	☐	Confer resistance to therapeutically useful antibiotics or antiviral agents
☐	☐	Enhance the virulence of a pathogen or render a nonpathogen virulent
☐	☐	Increase transmissibility of a pathogen
☐	☐	Alter the host range of a pathogen
☐	☐	Enable evasion of diagnostic/detection modalities
☐	☐	Enable the weaponization of a biological agent or toxin
☐	☐	Any other potentially harmful combination of experiments and agents

ChIP-seq

Data deposition

☐ Confirm that both raw and final processed data have been deposited in a public database such as GEO.

☐ Confirm that you have deposited or provided access to graph files (e.g. BED files) for the called peaks.

Data access links *May remain private before publication.*	*For "Initial submission" or "Revised version" documents, provide reviewer access links. For your "Final submission" document, provide a link to the deposited data.*
Files in database submission	*Provide a list of all files available in the database submission.*
Genome browser session (e.g. UCSC)	*Provide a link to an anonymized genome browser session for "Initial submission" and "Revised version" documents only, to enable peer review. Write "no longer applicable" for "Final submission" documents.*

Methodology

Replicates	*Describe the experimental replicates, specifying number, type and replicate agreement.*
Sequencing depth	*Describe the sequencing depth for each experiment, providing the total number of reads, uniquely mapped reads, length of reads and whether they were paired- or single-end.*
Antibodies	*Describe the antibodies used for the ChIP-seq experiments; as applicable, provide supplier name, catalog number, clone name, and lot number.*
Peak calling parameters	*Specify the command line program and parameters used for read mapping and peak calling, including the ChIP, control and index files used.*
Data quality	*Describe the methods used to ensure data quality in full detail, including how many peaks are at FDR 5% and above 5-fold enrichment.*
Software	*Describe the software used to collect and analyze the ChIP-seq data. For custom code that has been deposited into a community repository, provide accession details.*

Flow Cytometry

Plots

Confirm that:

☐ The axis labels state the marker and fluorochrome used (e.g. CD4-FITC).

☐ The axis scales are clearly visible. Include numbers along axes only for bottom left plot of group (a 'group' is an analysis of identical markers).

☐ All plots are contour plots with outliers or pseudocolor plots.

☐ A numerical value for number of cells or percentage (with statistics) is provided.

Methodology

Sample preparation	*Describe the sample preparation, detailing the biological source of the cells and any tissue processing steps used.*
Instrument	*Identify the instrument used for data collection, specifying make and model number.*
Software	*Describe the software used to collect and analyze the flow cytometry data. For custom code that has been deposited into a community repository, provide accession details.*
Cell population abundance	*Describe the abundance of the relevant cell populations within post-sort fractions, providing details on the purity of the samples and how it was determined.*
Gating strategy	*Describe the gating strategy used for all relevant experiments, specifying the preliminary FSC/SSC gates of the starting cell population, indicating where boundaries between "positive" and "negative" staining cell populations are defined.*

☐ Tick this box to confirm that a figure exemplifying the gating strategy is provided in the Supplementary Information.

Magnetic resonance imaging

Experimental design

Design type	*Indicate task or resting state; event-related or block design.*
Design specifications	*Specify the number of blocks, trials or experimental units per session and/or subject, and specify the length of each trial or block (if trials are blocked) and interval between trials.*
Behavioral performance measures	*State number and/or type of variables recorded (e.g. correct button press, response time) and what statistics were used to establish that the subjects were performing the task as expected (e.g. mean, range, and/or standard deviation across subjects).*

Acquisition

Imaging type(s)	*Specify: functional, structural, diffusion, perfusion.*
Field strength	*Specify in Tesla*
Sequence & imaging parameters	*Specify the pulse sequence type (gradient echo, spin echo, etc.), imaging type (EPI, spiral, etc.), field of view, matrix size, slice thickness, orientation and TE/TR/flip angle.*
Area of acquisition	*State whether a whole brain scan was used OR define the area of acquisition, describing how the region was determined.*

Diffusion MRI ☐ Used ☐ Not used

Preprocessing

Preprocessing software	*Provide detail on software version and revision number and on specific parameters (model/functions, brain extraction, segmentation, smoothing kernel size, etc.).*
Normalization	*If data were normalized/standardized, describe the approach(es): specify linear or non-linear and define image types used for transformation OR indicate that data were not normalized and explain rationale for lack of normalization.*
Normalization template	*Describe the template used for normalization/transformation, specifying subject space or group standardized space (e.g. original Talairach, MNI305, ICBM152) OR indicate that the data were not normalized.*
Noise and artifact removal	*Describe your procedure(s) for artifact and structured noise removal, specifying motion parameters, tissue signals and physiological signals (heart rate, respiration).*

Volume censoring

Define your software and/or method and criteria for volume censoring, and state the extent of such censoring.

Statistical modeling & inference

Model type and settings

Specify type (mass univariate, multivariate, RSA, predictive, etc.) and describe essential details of the model at the first and second levels (e.g. fixed, random or mixed effects; drift or auto-correlation).

Effect(s) tested

Define precise effect in terms of the task or stimulus conditions instead of psychological concepts and indicate whether ANOVA or factorial designs were used.

Specify type of analysis: ☐ Whole brain ☐ ROI-based ☐ Both

Statistic type for inference
(See Eklund et al. 2016)

Specify voxel-wise or cluster-wise and report all relevant parameters for cluster-wise methods.

Correction

Describe the type of correction and how it is obtained for multiple comparisons (e.g. FWE, FDR, permutation or Monte Carlo).

Models & analysis

n/a	Involved in the study
☐	☐ Functional and/or effective connectivity
☐	☐ Graph analysis
☐	☐ Multivariate modeling or predictive analysis

Functional and/or effective connectivity

Report the measures of dependence used and the model details (e.g. Pearson correlation, partial correlation, mutual information).

Graph analysis

Report the dependent variable and connectivity measure, specifying weighted graph or binarized graph, subject- or group-level, and the global and/or node summaries used (e.g. clustering coefficient, efficiency, etc.).

Multivariate modeling and predictive analysis

Specify independent variables, features extraction and dimension reduction, model, training and evaluation metrics.