Rotamer-free protein sequence design based on deep learning and self-consistency

Yufeng Liu1,5, Lu Zhang1,5, Weilun Wang2,5, Min Zhu1, Chenchen Wang1, Fudong Li1,3, Jiahai Zhang1,3, Houqiang Li2,6, Quan Chen1,3,6 and Haiyan Liu1,3,4

Several previously proposed deep learning methods to design amino acid sequences that automatically fold into a given protein backbone yielded promising results in computational tests but did not outperform conventional energy function-based methods in wet experiments. Here we present the ABACUS-R method, which uses an encoder–decoder network trained using a multitask learning strategy to predict the sidechain type of a central residue from its three-dimensional local environment, which includes, besides other features, the types but not the conformations of the surrounding sidechains. This eliminates the need to reconstruct and optimize sidechain structures, and drastically simplifies the sequence design process. Thus iteratively applying the encoder–decoder to different central residues is able to produce self-consistent overall sequences for a target backbone. Results of wet experiments, including five structures solved by X-ray crystallography, show that ABACUS-R outperforms state-of-the-art energy function-based methods in success rate and design precision.

Computational protein design brings the capability of inventing de novo proteins12–15 to fulfill various structural and functional needs—from therapeutics3,4 to biocatalysis5,6. One challenge in computational protein design is inverse protein folding, which is to select amino acid sequences that autonomously fold into a given backbone target. Although a few existing methods to realize this have been repeatedly verified by experiments16–18, these methods still suffer from deficiencies such as low success rates11,12, high sensitivity to target structures11,15 and overly monotonic designed sequences that lack the diversity and variability of natural amino acid sequences19,20. Further method innovations are called for to overcome these limitations11,12.

Conventional methods for inverse protein folding are based on optimizing empirical energy functions with respect to the sidechain types (and the sidechain conformations19). The energy functions are either physics-based (for example, RosettaDesign1, Proteus19 and EvoEF20) or statistically learned from data (for example, ABACUS10,11 and TERM16). Invariably, these energy functions employ the approximation of treating complicated, many-body molecular interactions as linear combinations of one- and two-body terms4,10,12,18. This approximation has been a fundamental accuracy-limiting factor despite continuous improvements21 of conventional energy functions.

We previously developed a statistical energy model named ABACUS4,10 (a backbone-based amino acid usage survey). Proteins designed with ABACUS have been verified with a number of experimentally solved structures22,23–25. Although the energy terms in ABACUS were devised to consider high-order coupling between various physical factors, the non-linear integration of coupled effects was restricted to the single-residue and the residue-pair-wise levels, beyond which separately learned energy terms were combined only in a linear way. Replacing this linear combination with an approach that can integrate non-linear coupling at higher orders may substantially increase the robustness and accuracy of this data-driven method.

A suitable approach that does not depend on the linear combination approximation is deep learning4. Deep learning has already tremendously advanced protein structure prediction (that is, forward folding)12,26–28. Its application in structure-based sequence design (that is, inverse folding) was explored in several recent studies12,29,30. Although promising methods have been demonstrated to outperform conventional energy function-based approaches in computational tests, in wet experiments, deep learning methods have not yet exhibited performances comparable to established energy function-based methods (see ref. 31 for a recent review). A study using a deep learning method has only until recently reported experimentally solved structures for two sequences designed for an ideal triose-phosphate isomerase (TIM)-barrel backbone12; thus, inverse protein folding by deep learning still needs improvements to have real impacts on computational protein design.

Nevertheless, we anticipate a well-developed deep learning method to be able to outperform conventional statistical energy models such as ABACUS in both in silico tests and wet experiments, as a deep learning-based method can retain features that have been proven to work while eliminating known problems such as treating complex molecular interactions using a linear combination of one- and two-body terms. In the current study, we have moved along this direction by developing the ABACUS-R method and evaluating it by using both in silico metrics and wet experiments.

Results
Overview of the model. The method comprises two parts: an encoder–decoder network (Fig. 1a) pre-trained to infer the sidechain type of a central residue from its local environment in a
Articles

Nature Computational Science

TQW

Local environment representation

in which several attributes of the central residue besides its side-chain type have been used as decoding objectives. The decoding accuracies of Model_{eval} for the various attributes are reported in Fig. 2a–c and Supplementary Fig. 1. Similar results for Model_{final} are reported in Supplementary Table 1.

Accuracy of the encoder–decoder for individual residues. A set of non-redundant Protein Data Bank (PDB) structures has been used to train the encoder–decoder network (see Methods). We have learned two groups of network parameters by splitting the PDB structures into training and test sets in two different ways. The first group of parameters (Model_{eval}) was learned by using about 95% of the structures for training and the remaining 5% for testing, with the structures for testing belonging to single-domain topology classes (according to the CATH 4.2 classification of protein structures) in the dataset, so that none of the test structures belonged to the same CATH topology as a training structure; Model_{eval} can therefore be used for unbiased computational evaluations. The second group of parameters (Model_{final}) was learned by randomly splitting the protein structures into roughly 95% for training and 5% for testing, disregarding their CATH structure classification. As Model_{eval} was trained on more complete data, it is expected to be applied in the same CATH topology as a training structure; Model_{eval} can therefore be used for unbiased computational evaluations. As Model_{final} was trained on more complete data, it is expected to be applied in the same CATH topology as a training structure; Model_{eval} can therefore be used for unbiased computational evaluations.

Both Model_{eval} and Model_{final} are able to recover the side-chain type of the central residue with an accuracy of around 50%. Moreover, for a substantial fraction of the decoding results that did not recover the exact native side-chain types, the decoder produces side-chain types physicochemically similar to the native types (for example, lysine for arginine, phenylalanine for tyrosine and so on; see Fig. 2b). Considering attributes other than the side-chain type for training substantially improved the accuracy for decoding the side-chain type (Fig. 2a, from 45.3% to 49.3% for Model_{eval}, and Supplementary Table 1, from 47.2% to 53% for Model_{final}), probably by compensating ‘noises’ in the training (natural) sequences (that is, the native side-chain types are not always the optimum choices for the given backbone).

For the categorical side-chain type attribute, the actual output of the decoder is a vector of logits values (see Methods). To examine whether the negatives of the logits values could be interpreted as effective energies for different residue types in a specific local three-dimensional environment, we chose a dataset of protein mutants with experimentally measured changes in protein stability (the ProTherm dataset), and examined the correlation between the stability changes (as measured by ΔΔG) of different mutants with the changes in −ΔΔlogits (see Methods); the results are shown in Fig. 2d. Supplementary Table 2 shows that the correlation coefficients of around 0.52 of ABACUS-R are comparable with the results of 0.39–0.59 produced by a variety of models (those models producing higher correlation coefficients than ABACUS-R have been trained on the specific task of predicting ΔΔG). Supplementary Fig. 2 also shows that ABACUS-R produces comparable results to earlier methods for another dataset of protein stability versus sequence variations.

Convergence of the sequence design iterations. We have applied Model_{eval} and self-consistent iterations to design overall sequences for 100 target structures taken from the test set of Model_{eval}. These target structures cover three main CATH classes (Supplementary Fig. 2). The framework of the encoder–decoder model. The input of the encoder includes backbone-only structure features and the side-chain types of surrounding residues (residue neighborhood modeling). The encoder is a transformer of 12 blocks, the output of which constitutes a general, integrated representation of the input features (local environment representation). To benefit from multitask learning, this representation is decoded into several different attributes (including the side-chain type) of the central residue (decoding central node). The iterative approach for designing self-consistent overall sequences. In each iteration, the encoder–decoder network is applied to each residue in a randomly chosen subset of residues to update its side-chain type according to its current, sequence-dependent local environment.

Fig. 1 | An overview of the ABACUS-R method. a, The framework of the encoder–decoder model. The input of the encoder includes backbone-only structure features and the side-chain types of surrounding residues (residue neighborhood modeling). The encoder is a transformer of 12 blocks, the output of which constitutes a general, integrated representation of the input features (local environment representation). To benefit from multitask learning, this representation is decoded into several different attributes (including the side-chain type) of the central residue (decoding central node). b, The iterative approach for designing self-consistent overall sequences. In each iteration, the encoder–decoder network is applied to each residue in a randomly chosen subset of residues to update its side-chain type according to its current, sequence-dependent local environment.

three-dimensional protein structure, and an iterative process (Fig. 1b) of applying this encoder–decoder to update the side-chain type of each residue in a given backbone to obtain self-consistent overall sequences (starting from random initial sequences). The two parts taken together, the overall approach is effectively a backbone-based sequence generator or an encoder that encodes backbone structures into amino acid sequences (see the Methods for more detailed descriptions of the model, discussions about designing decisions and computational costs).
For each target, ten sequences have been designed by using runs starting from different random initial sequences. As the iterative approach is effectively a greedy algorithm to maximize the (predicted) probabilities of the sidechain types, we monitored the negative logarithms of these probabilities (the \(-\log P\) values) during the iterations. Figure 3a shows how the averaged per-residue \(-\log P\) value decreased and converged to a plateau. Meanwhile, the sidechain types of most residues were converging towards those of the final sequences (Fig. 3b). For all of the target structures, the iterative runs can produce self-consistent sequences that either completely converge or up to having only a very small (target dependent) number of fluctuating positions (see the inset of Fig. 3b and Supplementary Fig. 3a).

If not all positions converge, the sequence of the lowest total \(-\log P\) values recorded during a run is taken as the final design result. This choice is for convenience and is not critical, as both the number of unconverted positions and the fluctuations of \(-\log P\) are very small during the plateau stage (see the insets of Fig. 3a,b), suggesting that all of the sequences at this stage are equally acceptable.

For the same target structure, the runs starting from different random initial sequences usually lead to highly similar (although not identical) sequences (sequence identities from 0.76 to 0.89 for the 100 test targets), with the per-residue \(-\log t\)s varying only narrowly (that is, \(\pm 0.50\)); thus, the sequences designed by different runs can be considered as equally plausible.

For the 100 test targets, the self-consistent design results obtained by using Model\(_{\text{final}}\) are highly similar to those obtained by using Model\(_{\text{rand}}\) in terms of \(-\log t\)s and sequence identities (Supplementary Fig. 4); thus, ABACUS-R design results do not seem to be sensitive to which exact individual structures have been included in or excluded from the training data, as the underlying network only models local statistical distributions, to which each individual training structures should only contribute a very small fraction.

**Designed sequences compared with native sequences.** For the 100 test targets, the average identity between the designed sequences and corresponding native sequences is 43.1\% \(\pm 5.4\%\) (this identity varied from 32\% to 61\% for various individual targets). Although the native sidechain type recovery rate does not exhibit strong dependency on the CATH class (Fig. 3c), this recovery rate has the expected tendency of decreasing with increasing solvent accessibility (see Supplementary Fig. 3b). The Pearson correlation coefficient of the amino acid type compositions of the designed and the native sequences is 0.93 (Fig. 3d). Nevertheless, some sidechain types such as glutamic acid, alanine and lysine have been used more frequently in the designed rather than the native sequences, while the sidechain types of obviously reduced usages in the designed sequences include glutamine, histidine and methionine (Fig. 3d).

We have also compared the native sidechain type recovery rate of ABACUS-R with related methods. The results are summarized in Supplementary Table 4 (for single residue redesign), and Supplementary Tables 5 and 6 (for complete sequence redesign). Comparisons on the same groups of target backbones indicate that ABACUS-R produces higher native recovery rates than the model of Ingraham et al.\(^{29}\), 3DCNN\(^{17}\) and ProteinSolver\(^{30}\), and produces lower rates than two more recent models\(^{39,40}\).

We note that the native sidechain recovery rate metric, although very useful, has severe limitations for judging a sequence design method: first, for a substantial portion of residues in natural
Fig. 3 | Results of overall sequence design for natural backbones. The results have been obtained with Model$\text{ext}$ and are averages over 100 target backbones. a, Evolution of $-\log P$ per residue during the self-consistent iterations. b, The evolution of the number of residues with unconverged sidechain types (that is, the sidechain types are different from those in the final sequences) during the self-consistent iterations. The insets in a and b show enlarged views of the tail of the curves for visualizing the small fluctuations of the respective vertical values. c, The rates of recovering the native sidechain types for backbone targets in three CATH architecture classes. Center lines show the medians; box limits indicate the 25th and 75th percentiles; whiskers extend 1.5-times the interquartile range from the 25th and 75th percentiles; outliers are represented by dots. d, The distributions of sidechain types in the designed sequences and in the native sequences. e, Scattering plot of the Rosetta energies per residue of the designed sequences versus the Rosetta energies per residue of corresponding native sequences. Blue (red) points correspond to sequences designed by ABACUS-R (ABACUS2). A gray line is drawn along the diagonal. f, Scattering plot of the ABACUS-R $-\logit$ values per residue of the designed sequences versus the ABACUS-R $-\logit$ values of corresponding native sequences. Blue (red) points are for sequences designed by ABACUS-R (ABACUS2). A gray line is drawn along the diagonal.
sequences, the native sidechain types may not be optimum for the given structures; and second, a few incorrectly designed residues in a designed sequence of a high native recovery rate could disrupt protein folding (and lead to experimental failure).

The designed sequences and the native sequences have been further compared in their per-residue Rosetta energies (computed after backbone relaxation)\(^{43}\). For the same target backbone, the Rosetta energies of the designed and the native sequences are similar (Fig. 3e). We also applied the statistical energy function-based method ABACUS-R to design sequences for the 100 test backbones and considered their Rosetta energies and ABACUS-R \(-\logits\) values (Fig. 3e,f). Although ABACUS-R sequences are of systematically lower Rosetta energies than both the native and ABACUS-R sequences, they are of systematically higher \(-\logits\) than the two groups of sequences. These results indicate that the \(-\logits\) metric has constituents that are orthogonal to the Rosetta energy. This partial orthogonality is also exemplified by the moderate correlations between the per-residue \(-\logits\) values and the Rosetta energies of the designed and the native sequences (the respective Pearson correlation coefficients are 0.24 and 0.17; see Supplementary Fig. 3c).

We applied ABACUS-R to the four target backbones considered in the 3DCNN work\(^{11}\). The native sidechain type recovery rates of ABACUS-R and 3DCNN are compared in Supplementary Table 6. Although Supplementary Table 7 shows that the ABACUS-R \(-\logits\) values of the 3DCNN sequences are clearly above those of the native sequences, which in turn are clearly above the \(-\logits\) values of the ABACUS-R sequences, Supplementary Fig. 5 shows that the groups of native residues retained by ABACUS-R and 3DCNN largely overlap, and that these residues, as well as the residue-wise variations of the \(-\logits\) values among the various sequences, are dispersed throughout the entire sequence.

**Structure prediction on designed sequences.** We applied AlphaFold2\(^{26}\) to predict the structures for all of the 1,000 sequences designed for the 100 test targets. The distributions of the template modeling scores (TM-scores) for aligning the predicted structures with the respective design targets\(^{43}\) and the predicted local distance difference test scores (pLDDT scores)\(^{33}\) of the predicted structures are shown in Supplementary Fig. 6. The predictions have been performed without using multiple sequences alignment information, and in two modes\(^{26}\): one disallowing the use of any structural templates (template-off) and another allowing the use of structural templates (template-on). For comparison, we also applied AlphaFold2 to the native sequences with the template-off mode, in which the agreements between the predicted and the target structures are poor for both the native and the designed sequences, with only a few TM-score values of above 0.6 (Supplementary Fig. 6a). We note that all of the low-TM-score predictions do not have sufficient pLDDT scores to be considered reliable (Supplementary Fig. 6b); thus, at least for the 100 targets considered here, the template-off AlphaFold2 predictions are not useful for judging sequence-structure compatibility.

In the template-on mode, 811 of the 1,000 structures predicted for the designed sequences agree well with the respective design targets, with TM-scores of above 0.6 (and mostly around 0.9). The pLDDT scores of these high-TM-score predictions are also high (Supplementary Fig. 6a), indicating high sequence-structure compatibility. For the twelve target structures for which all of the designed sequences failed to produce predicted structures with TM-scores of above 0.4, we inspected the structures and found that most of them have been determined as parts of larger complexes (the structures of these targets are shown in Supplementary Fig. 6c). These targets may therefore be dependently stabilized either by special interactions (for example, coordination with multiple metal ions) or interactions with other macromolecules; and thus could be out of reach of a general structure prediction or sequence design method.

Supplementary Fig. 5c shows the AlphaFold2 prediction results of the ABACUS-R-designed and the 3DCNN-designed sequences for the four target backbones considered by 3DCNN\(^{11}\). Although all of the predicted results have TM-scores of above 0.6, the pLDDT scores of the ABACUS-R sequences are more concentrated in the high-score region than the more varied 3DCNN sequences.

Although ABACUS-R has been learned from natural proteins, we expect it to be applicable to designed backbones, as the core network of ABACUS-R encodes the dependence of sidechain type on non-natural overall architectures\(^{43}\). The obtained sequences are far from any known natural protein sequences according to HHBlits searches\(^{44}\) (Supplementary Table 8). Applying AlphaFold2 to these sequences lead to predicted structures of high quality (as indicated by high pLDDT scores) and in close agreements with

---

**Table 1**: Overview of experiments on designed sequences.

| Target  | \(-\logits\) | Identity (%) | Identity (%) | Examined/expressed/soluble | Crystallization attempted/crystal obtained | Structure solved | HSQC spectra measured |
|---------|-------------|--------------|--------------|----------------------------|------------------------------------------|------------------|---------------------|
| 1r26    | \(-6.76 \pm 0.21\) | 53.4 ± 2.9   | 83.9 ± 5.3   | 10/10/10                   | 5/4                                      | 3                | 0                   |
| 1cy5    | \(-6.50 \pm 0.17\) | 47.1 ± 3.9   | 57.4 ± 2.2   | 10/8/7                     | 1/1                                      | 1                | 2                   |
| 1ubq    | \(-6.49 \pm 0.18\) | 40.9 ± 2.4   | 55.1 ± 2.3   | 10/7/6                     | 2/0                                      | 0                | 1                   |
| 1ubq    | \(-6.30 \pm 0.22\) | 45.1 ± 2.7   | 78.9 ± 4.8   | 7/6/6                     | 6/1                                      | 0                | 1                   |
| 1ubq    | \(-6.12 \pm 0.20\) | 43.2 ± 4.1   | 61.4 ± 2.2   | 10/10/10                   | 1/0                                      | 0                | 2                   |

Total number of experimentally examined designs: 57
Total number of designs that are soluble and well-folded based on (at least) \(^1^H\) NMR: 49
Total number of designs with HSQC spectrum measured: 6
Total number of designs with solved crystal structures: 5 (RMSD: 0.51 ± 0.88 Å)

From left to right: the columns, respectively, correspond with the PDB IDs of the target backbones; the average \(-\logits\) per residue of the designed sequences; the average identities between designed sequences; the number of examined/expressed/soluble proteins; the numbers of proteins for which crystallization was attempted/crystals were obtained; the numbers of crystal structures solved; and the numbers of proteins for which NMR HSQC spectra were measured. For each target backbone, the first and second rows correspond to designs examined in the first and second experimental batches, respectively. Each of the bottom four rows summarizes the total number of proteins in a certain group as described by the text. Note that the group of proteins with solved crystal structures do not overlap with the group of proteins with HSQC measured. RMSD, root mean square deviations of Cα atom positions of crystal structures from design models.
Articles are shown (b, 1r26-A3; c, 1cy5-A7; and a, 1ubq-A4). From left to right: results of size-exclusion chromatography experiments; 'H NMR spectra; thermal capacity versus temperature curves from DSC experiments; the experimentally solved structures (1r26-A3, 1cy5-A7) or NMR HSQC (1ubq-A4). The Cα RMSDs of the structures from the corresponding backbones are 0.57 Å for 1r26-A3 and 0.88 Å for 1cy5-A7. For 1ubq-A4, superposition of the crystal structure of 1ubq (cyan) with the AlphaFold2-predicted structure of 1ubq-A4 (pink) is shown together with the HSQC spectrum, in which ω and ω' refer to the chemical shifts of (covalently bonded) 'N and 'H nuclei, respectively.

Fig. 4 | Results of experimental analysis of designed proteins. a–c. For each of the three backbone targets considered, the results of one designed protein are shown (a, 1r26-A3; b, 1cy5-A7; and c, 1ubq-A4). From left to right: results of size-exclusion chromatography experiments; 'H NMR spectra; thermal capacity versus temperature curves from DSC experiments; the experimentally solved structures (1r26-A3, 1cy5-A7) or NMR HSQC (1ubq-A4). The Cα RMSDs of the structures from the corresponding backbones are 0.57 Å for 1r26-A3 and 0.88 Å for 1cy5-A7. For 1ubq-A4, superposition of the crystal structure of 1ubq (cyan) with the AlphaFold2-predicted structure of 1ubq-A4 (pink) is shown together with the HSQC spectrum, in which ω and ω' refer to the chemical shifts of (covalently bonded) 'N and 'H nuclei, respectively.

We note that passing the AlphaFold2 structure prediction test cannot guarantee that a designed sequence will actually fold as expected. As AlphaFold2 was shown to be unable to correctly predict the disruption of folded protein structures by single mutations41, a designed sequence contains only a few folding-disrupting residues (and thus fails to fold in wet experiments) may still pass this computational test.

Experimental analysis of designed sequences. Only a few studies so far have reported experimental examinations of protein sequences designed on the basis of deep learning17,20. Proteins designed by 3DCNN17 and ProteinSolver20 were shown to have desired secondary structure contents and to fold cooperatively according to circular dichroism signatures. So far the only experimentally solved atomic structures for de novo sequences designed using a deep learning method were for two sequences designed using a de novo TIM-barrel by 3DCNN17. That backbone consisted of structural elements that were likely to be far more regular or ideal than those in natural backbones; thus, when it comes to de novo sequence design for (natural) backbones that are abundant with diverse, none-ideal structure elements, we do not know about previous validations of a deep learning method by experimentally solved atomic structures.

Here we examined the sequences designed by ABACUS-R for three natural backbones using wet experiments. These backbones (with PDB IDs: 1r26, 1cy5 and 1ubq) have been chosen as they had been used to experimentally evaluate the ABACUS model19,22,23. To assess the method in an unbiased way, the automatically generated sequences by ABACUS-R have been used as is; that is, without any post-design selection or adjustment.

Two batches of wet experiments have been carried out to examine proteins designed using two different protocols. The first protocol employs the self-consistent iterations described above to maximally converge the sidechain types of all residues. Although this protocol effectively minimizes the −logits values, the designed sequences are highly similar to each other (the identities between different sequences designed for the same backbone are around 80%; see Table 1 and Supplementary Table 9). In the second protocol, the sidechain types are sampled from distributions derived from the output of the decoders (see Methods), generating more diverse sequences from different sequence design runs, albeit with somewhat higher −logits values (see Table 1 and Supplementary Table 10). Among the 27 sequences examined in the first batch, 26 have led to successful protein expression in Escherichia coli (E. coli). All of the expressed proteins could be readily purified in soluble form. The purified proteins have been subjected to a range of experiments including size-exclusion chromatography, 'H NMR spectrum measurement, NMR heteronuclear single quantum coherence (HSQC) spectrum measurement, differential scanning calorimetry (DSC), and protein crystallization and X-ray structure determination (Table 1). Complete data from these experiments are reported in Supplementary Figs. 8–11. Representative results for
of the solved structures agree with the corresponding design targets (Supplementary Fig. 11). One high-resolution structure of a second batch protein has been solved (1r26-B4), whose structure for a second batch protein has been solved (1r26-B4), whose structure for a second batch protein has been solved (1r26-B4), whose structure for a second batch protein has been solved (1r26). The size-exclusion chromatography and NMR data (Supplementary Figs. 11–15) suggest that all of these proteins are monomeric and likely to fold into well-defined three-dimensional structures. The DSC curves for five proteins covering the three target backbones indicate that these proteins unfold cooperatively at high temperatures from 97 to 117°C (Supplementary Fig. 11). High-resolution structures of four proteins—including proteins designed using the maximum convergence protocol for the same target—are shown in Fig. 4a–c. The purified proteins have been subjected to the aforementioned experimental analysis (Table 1 and Supplementary Table 13). The identities between the different sequences designed for the same target are around 58%. Among the 30 designed proteins analyzed, 25 led to successful protein expression in E. coli, and 23 could be purified in soluble form. The results are reported in Supplementary Figs. 11–15. Again, the size-exclusion chromatography and NMR data (Supplementary Figs. 12–15) suggest that all of these proteins are monomeric and fold into well-defined three-dimensional structures. The DSC curves for five proteins covering the three target backbones indicate that these proteins unfold cooperatively at high temperatures from 85 to 118°C (see Supplementary Fig. 11). One high-resolution structure for a second batch protein has been solved (1r26-B4), whose structure for a second batch protein has been solved (1r26-B4), whose structure for a second batch protein has been solved (1r26-B4), whose structure for a second batch protein has been solved (1r26-B4), whose structure for a second batch protein has been solved (1r26-B4), whose structure for a second batch protein has been solved (1r26-B4), whose structure for a second batch protein has been solved (1r26-B4), whose structure for a second batch protein has been solved (1r26-B4), whose structure for a second batch protein has been solved (1r26). The size-exclusion chromatography results and the 1H NMR spectra have covered all of the purified proteins in the first batch (Supplementary Figs. 8–10 and Supplementary Table 11). They indicate that all of these proteins are monomers and likely to fold into well-defined three-dimensional structures. Five proteins covering the three target backbones have been measured by DSC experiments, which showed that all of these proteins unfold cooperatively at high temperatures from 97 to 117°C (Supplementary Fig. 11). High-resolution structures of four proteins—including three designed for the 1r26 backbone and one designed for the 1cy5 backbone—have been solved with X-ray crystallography. All of the solved structures agree with the corresponding design targets with high precision (the RMSDs of Ca atoms ranged from 0.51 to 0.88 Å; see Supplementary Figs. 8 and 9). For proteins designed on the 1ubq backbone, we only obtained crystals for one protein (1ubq-A4), which were twinned, and no X-ray structure could be solved; however, the NMR HSQC spectrum and the DSC results of 1ubq-A4 (Fig. 4c) prove that this protein folds into a well-defined three-dimensional structure. Taking the experimental data together with AlphaFold2 prediction results (Fig. 4c; see also Supplementary Table 12), which shows that the predicted structures have high TM-scores relative to 1ubq as well as high pLDDT scores), we expect the structures of the proteins designed for the 1ubq backbone to also fold into the target backbone with high precision.

In the second batch of experiments, we examined proteins obtained by using the modified protocol that designs more diverse sequences. Table 1 shows that the identities between the different sequences designed for the same target are around 58%. Among the 30 designed proteins analyzed, 25 led to successful protein expression in E. coli, and 23 could be purified in soluble form. The purified proteins have been subjected to the aforementioned experimental analysis (Table 1 and Supplementary Table 13). The expected sidechain types and sidechain packing have been designed by ABACUS-R. a. Crystal structures of the different proteins showing (as sticks) the sidechains of residue 24 surrounded by residues 22, 26, 36, 37, 40, 52, 54 and 75; 1r26 is the native sequence; 1r26–A3, 1r26–A6, 1r26–A7 and 1r26–B4 are ABACUS-R-designed sequences. b. Diagrams showing the sidechain types of residue 24 and its surrounding residues in the different proteins.

The orange circle indicates the central residue; the green circles indicate residues that are of the native sidechain types; the yellow circles indicate residues for which non-native sidechain types have been designed; and the circles with bold borders indicate residues whose sidechain types did not completely converge in the self-consistent iterations. c. Structural details of some sidechains forming non-native inter-residue polar interactions in the crystal structures of the designed 1cy5-A7. A superposition of the crystal structures of 1cy5 and 1cy5-A7 is shown on the left; regions of interest are marked by dashed line-encircled numbers, with corresponding enlarged views shown on the right. In each enlarged view, the structure of the native protein is shown in cyan on the left side and the structure of a designed sequence is shown in green on the right side. The text labels above the enlarged views indicate the region number and the sidechain substitutions from the native to the designed sequence.

The size-exclusion chromatography results and the 1H NMR spectra have covered all of the purified proteins in the first batch (Supplementary Figs. 8–10 and Supplementary Table 11). They indicate that all of these proteins are monomers and likely to fold into well-defined three-dimensional structures. Five proteins covering the three target backbones have been measured by DSC experiments, which showed that all of these proteins unfold cooperatively at high temperatures from 97 to 117°C (Supplementary Fig. 11). High-resolution structures of four proteins—including those successfully designed using the maximum convergence protocol for the same target—are shown in Fig. 4a–c.
targets (see the DSC curves of the first and second batch proteins in Supplementary Fig. 11; note that there are exceptions to this observation).

Closer inspection of the crystal structures indicates that ABACUS-R is able to design alternative sidechain combinations that form equally well-packed structures. As examples, Fig. 5a,b show how a group of core residues of different sidechain type combinations are packed in native 1r26, and in the crystal structures of the four ABACUS-R designed proteins; Fig. 5c shows the structural details for some of the sidechains that form (non-native) inter-residue polar interactions in the structures of the designed 1c5y-A7 (in Supplementary Fig. 16, the crystal structures of the native 1r26 and of the designed 1r26 proteins are compared in similar detail). The ABACUS-R model has therefore learned how to find sidechain type combinations with good packing qualities without explicitly going through atomic sidechain coordinates. As shown in Supplementary Fig. 17, the inter-residue (pseudo)coevolution information derived using Direct Coupling Analysis on ABACUS-R-designed sequences is correlated with inter-residue distances, similar to the coevolution information from natural sequences.

Discussion

A number of previous studies have investigated deep learning-based approaches to inverse protein folding19–25. Among the various models, the encoder–decoder part of ABACUS-R is the most similar conceptually to those using deep learning networks to predict the sidechain type of a central residue on the basis of its local environment17,29.

Compared with earlier methods, several features of ABACUS-R are unique and may be important for overcoming some of the main difficulties in solving the inverse folding problem. Notably, ABACUS-R is rotamer-free and does not require the explicit modeling of sidechain conformations. This eliminates the use of approximately reconstructed sidechains by using discrete rotamers3, or through the sampling of continuous distributions17. Although using atomically detailed sidechains could be advantageous for certain molecular modeling tasks, for de novo sequence design, accurate coordinates of sidechain atoms are not available from input. When sidechains have to be approximately reconstructed and fed to a model trained on experimentally determined atomic structures, the overall accuracy of the method may degrade.

Another unique feature of ABACUS-R is that complete sequences are designed through self-consistent iterations. This is possible as sidechains are not explicitly modeled. Should sidechain conformations have to be constructed and updated, the number of degrees of freedom would be substantially increased and the energy landscape would have become too rugged for the iterative updating algorithm to converge meaningfully. Moreover, on a rugged, high-dimensional energy landscape, it can be hard to produce a sequence that is free of structure-disrupting local interactions. We note that such design errors may not affect much computational metrics such as native recovery rates or even AlphaFold2 prediction results16. However, they will considerably degrade the success rates (or robustness) in wet experiments. Another property of the self-consistent iterative approach is that the sidechain type of every residue will be repetitively updated according to the evolving environmental contexts. This is different from a ‘one-pass’ sequence generation process, in which any improperly chosen residue in an early generation step will be accumulated and propagated in subsequent steps, which can lead to an exponentially decaying probability of generating truly foldable overall sequences with increasing sequence length16.

An encouraging finding of the current study is that in experimental tests on the three natural backbones, ABACUS-R has exhibited much higher success rates and design precision than those reported for the energy function-based models ABACUS1,10 and RosettaDesign14. For example, more than half of the original ABACUS designs on these three targets could not be expressed or purified; only after introducing mutations through directed evolution, structures of ABACUS designs for the 1r26 and 1c5y targets had been successfully solved3,16. Marin and colleagues14 have experimentally examined proteins designed for eight different backbones of thioredoxin proteins (the 1r26 target used here is also a thioredoxin), and reported that in the most favorable case (the target backbone for which RosettaDesign exhibited the highest success rate), only three out of eight experimentally examined designed proteins could be purified as monomers. In an earlier study of these authors, the crystal structure of one protein designed by Rosetta for a thioredoxin backbone had been determined with RMSDs of 1.8 to 2.0 Å from the target backbone16. There could be multiple reasons for the improved performance of ABACUS-R over energy function-based methods. One of which is that as a deep learning method, ABACUS-R avoided the use of linear combinations of energy terms. A second reason could be that ABACUS-R does not explicitly use sidechain structures and thus it does not suffer from the incompatibility between the inaccurately reconstructed positions of sidechain atoms (for example, by using rotamers) and the high sensitivity of some of the energy terms (for example, the van der Waals repulsion terms) to atomic positions.

There are still several limitations of our approach. One is that the current decoder predicts the sidechain torsional angles with only moderate accuracies (Fig. 2a and Supplementary Fig. 1); thus, atomic models of sidechain structures—if they are needed after sequence selection—have to be constructed with other tools. Another limitation is that ABACUS-R by itself does not take into account the requirements on sidechain types for properties other than the protein structure, such as protein function. In principle, restraints on sidechain type choices for protein function or other desired properties may be derived before sequence selection from other considerations or analyses (for example, the prediction of functionally important protein residues from evolutionary information), and then imposed during ABACUS-R sequence selection. The application and validation by wet experiments of ABACUS-R in the sequence design for proteins with functions is still to be carried out in future studies.

Methods

Modeling the three-dimensional local environment. For the encoder–decoder network of ABACUS-R (see Fig. 1c), the input comprises the sidechain types and three-dimensional backbone structure information of all of the structurally neighboring residues of a central residue. We emphasize that the sidechain type of the central residue is not used as input. Neither are the sidechain conformations of the neighboring residues.

The set of residues that form the three-dimensional local environment of a central residue is defined to be its k-nearest neighbor residues according to the Ca–Ca distances. As input of the encoder, we have considered the following three types of features for each neighboring or surrounding residue: its location and orientation relative to the central residue (X_{\text{neighbor}}), its sequence position relative to the central residue (X_{\text{position}}), and its sidechain type (X_{\text{RSP}}). The raw input for the different types of information are defined as below.

The components of X_{\text{neighbor}} are defined using (1) the Ca–Ca distance from the environment residue to the central residue; (2) the Cartesian coordinates of the environment residue’s Cα in a local coordinate frame defined using the backbone N, Ca, and C atoms of the central residue (in this frame, Cα is at the origin, the Ca–N direction is along the x axis, and the N–Ca–C plane corresponds to the x–y plane); and (3) the rigid body rotation that aligns the orientations of the two local coordinate frames of the central and the environment residues. The Ca–Ca distance is mapped to a vector of 16 components using a set of Gaussian radial basis functions centered around 16 distance values ranging from 0 to 20 Å; the Cartesian coordinates are used as is; the rigid body rotation is mapped to a three-dimensional vector whose direction corresponds to the axis of the rotation and whose length corresponds to the magnitude of the rotated angle. After these mappings, X_{\text{neigh}} for each residue is a vector of 22 real-valued components. The components of X_{\text{position}} are defined based on the difference between the position indices of the neighboring residues and the central residue, that is, Δ = X_{\text{position}}. The X_{\text{RSP}} is a one-hot vector that encodes 129 states, with each integer value of Δi from −64 to +63 corresponding to one of 128 states, and the remaining one state encoding whether the corresponding neighboring residue is not on the same peptide chain as the central residue.
The \( X_{\text{seq}} \) is a one-hot vector that encodes the 20 sidechain types plus a special MASKED type (for the central residue, see below).

Features of the central residue are mapped to vectors of the same dimensions as the neighboring residues, only that all components of its \( X_{\text{seq}} \) are zeros, and its \( X_{\text{seq}} \) value is mapped to the MASKED type. Aside from these, the backbone conformation centered around the central residue (\( X_{\text{seq}} \)) is also considered. The components of \( X_{\text{seq}} \) are the 15 backbone torsional angles \( \Phi \sim \Psi \sim \omega_1 \sim \cdots \sim \Phi \sim \Psi \sim \omega_{15} \), in which \( i \) is the position index of the central residue. The angles are divided by \( \pi \) to output values in \([-1,1]\). To maintain the same input dimensions for vectors encoding the central and the neighboring residues, the \( X_{\text{seq}} \) vectors are also formally considered for all of the neighboring residues, but the values of all of their components are zero.

The above input features of the encoder have been defined to include complete information of relative backbone positions between the central the surrounding residues while remaining invariant with respect to three-dimensional translations and rotations. Moreover, the transformer architecture of the encoder (see below) is invariant to permutation of the ordering of neighboring residues.

To produce input for the transformer encoder from the above raw input feature vectors, the one-hot-encoded feature vectors \( X_{\text{seq}} \) and \( X_{\text{seq}} \) are linearly transformed, whereas the feature vectors \( X_{\text{seq}} \) and \( X_{\text{seq}} \) which are of real-valued components, are transformed using a linear layer. The resulting intermediate vectors are noted, respectively, as \( E_{\text{seq}} \), \( E_{\text{seq}} \), and \( E_{\text{seq}} \), which are simply concatenated together to form the overall input vector \( E \), namely

\[
E_{\text{seq}} = [\begin{matrix} W_{\text{seq}} X_{\text{seq}} & E_{\text{seq}} & E_{\text{seq}} \end{matrix} _{\text{seq}}],
\]

where the matrices \( W_{\text{seq}} \) and \( W_{\text{seq}} \) and the biasing vectors \( b_{\text{seq}} \) and \( b_{\text{seq}} \) comprise trainable elements, which are shared between all residues.

**Architecture of the transformer-based encoder.** We note the set of \( E \) vectors for all residues, as \( [E_i^n, n=0,1,\ldots,n], \) in which \( E_i \) is the vector for the central residue. These set of vectors are fed to a transformer of 12 blocks, as shown in Fig. 1. Each block of the transformer is composed sequentially of a multihd head self-attention module\(^a\), a layer normalization transformation\(^b\), a feed-forward module and a layer normalization transformation, as formulated below,

\[
F_i^{(n)} = E_i,
Q_i^{(n)} = W_{q} F_i^{(n)} + b_{q},
K_i^{(n)} = W_{k} F_i^{(n)} + b_{k},
V_i^{(n)} = W_{v} F_i^{(n)} + b_{v},

L_i^{(n)} = \text{LN}(F_i^{(n)} + \text{MHA}(Q_i^{(n)}, K_i^{(n)}, V_i^{(n)}, \text{softmax} = \sigma_{0,1,2,k})),

F_i^{(n+1)} = \text{LN}(F_i^{(n)} + \text{FFN}(L_i^{(n)})�),
\]

where the residue index \( i \) is from 0 to \( k \), the block index \( n \) is from 1 to 12, the \( \text{MHA}(\sigma) \), \( \text{LN}(\cdot) \) and \( \text{FFN}(\cdot) \) operations refer to the multihd self-attention module, the feed-forward module and the layer normalization transformation, respectively. From the output of the above transformer, the vector \( F_i^{(n)} \) is taken as a general representation of the local environment of the central residue, which is to be decoded into various attributes of the central residues.

The decoders and training losses. The encoder-decoder network has been trained by using a multitask learning strategy\(^b\), in which multiple attributes of the central residue have been simultaneously considered as decoding objectives, including the sidechain type, the secondary structure state, the solvent accessible surface area (SASA), the sidechain torsional angles (that is, \( \chi \) and \( \psi \)), and the crystallographic B factor of the sum of the mainchain atoms. We note that for secondary structure states, both the three-state and the eight-state schemes as used by the DSSP program\(^\dagger\) have been simultaneously considered. For sidechain torsions, we have only considered \( \chi \) and \( \psi \) but not \( \chi \) or \( \psi \), or \( \chi \) or \( \psi \) should be useful for encoding sidechain structures only when they will be associated with correct and accurate \( \chi \) and \( \psi \).

It seems that the decoding accuracy for \( \chi \) and \( \psi \) of our current model (see the 'Results' section) cannot guarantee the additional consideration of \( \chi \) or \( \psi \) to be beneficial.

Our main reason for using multitask learning is that the so-called ground-truth native sidechain type labels in the training natural sequences are actually quite noisy, as the native sidechain types are not always the optimal choices for the given structures. The indusive biases introduced by the auxiliary tasks can help to reduce the encoder-decoder network's Rademacher complexity or its ability to fit these noises\(^b\). The secondary structure state and SASA auxiliary tasks are expected to introduce useful inductive biases as they are known to strongly affect backbone type preferences, and they have been considered as key features in our previous (experimentally validated) ABACUS method\(^b\). The sidechain conformations of the central residues have been considered as auxiliary tasks as they are expected to correlate with the conformations of the surrounding sidechains. Thus they may implicitly help more structural information about the local environment to be retained in the output representation. The B-factors has been included for a similar reason.

Different attributes of the central residues are decoded by passing \( F_i^{(n)} \) through different perceptron networks (decoders), each decoder network having a single hidden layer.

For the sidechain type decoder, its output layer contains twenty perceptron nodes, the output values (the logits) are transformed using Softmax to yield normalized probabilities predicted for the 20 sidechain types. Namely, for sidechain type \( a \),

\[
p_a = \exp(y_a) / \sum_{a=1}^{20} \exp(y_a),
\]

The cross-entropy loss function is used to measure the decoding loss, namely, for each training item,

\[
L_{\text{aux}} = -\sum_{a=1}^{20} y_a \log(p_a),
\]

where \( a \) is sidechain type index, \( y = (y_1, y_2, \ldots, y_{20}) \) is the one-hot vector produced by the ground-truth label (that is, the native sidechain type) and \( p = (p_1, p_2, \ldots, p_{20}) \) is the vector of the predicted probabilities.

The decoders and losses for the secondary structure state are defined in the same way as that for the sidechain type. Both the three-state and the eight-state categorization schemes are used at the same time. The cross-entropy losses are respectively noted as \( L_{\text{SS3}} \) and \( L_{\text{SS8}} \).

For the numerical attributes including the SASA and the X-ray B factor, we first normalized the attributes using the means and standard variations computed from the training data. The corresponding decoders regress the normalized attributes. The losses \( L_{\text{SSA3}} \) and \( L_{\text{SSA8}} \) are defined as the L1 errors of the regressions.

The total loss for the multitask learning task is given by

\[
L = L_{\text{aux}} + \lambda_1 (L_{\text{SS3}} + L_{\text{SS8}}) + \lambda_4 L_{\text{SSA3}} + \lambda_4 L_{\text{SSA8}} + \lambda_1 L_{\text{Bfactor}} + \lambda_4 (L_{\text{神器}} + L_{\text{神器}}),
\]

where \( \lambda_i \) are the trade-off parameters to balance the different losses. As the final decoding accuracies are not sensitive to the exact values of these parameters (see also the ablation results in Fig. 2a) and also because of the computational costs (training the entire model from scratch takes about 15h on a single GeForce RTX 3090 GPU), we did not carry out extensive explorations on the various combinations of the trade-off parameters. Instead, we have separately and gradually increased the individual \( \lambda \) parameters from small starting values until the decoding accuracy for the corresponding attributes maximized while the decoding accuracy for the sidechain type did not decrease. The finally used values for \( \lambda_i \) are 0.2, 0.2, 0.2 and 0.5, respectively. Sufficient training (and validation) curves of the various losses. Although all of the losses decreased and were gradually plateauing, the sidechain type validating loss plateaued first during the training.

The self-consistent iterations. To design complete sequences for a given target backbone, we start from an initial sequence with randomly chosen sidechain types for all residues. In each iteration, we randomly select a number of residues, consider each of them as the central residue, apply the pre-trained encoder-decoder network to its current three-dimensional local environment, and update the sidechain type of the central residue according to the output of the (sidechain type) decoder. Within one iteration, the calculations for all of the selected residues are executed in parallel to take advantage of parallel computation. The initial number of residues selected for parallel sidechain type updating is set to be 80% of the sequence length, and this number is gradually reduced in subsequent iterations until the final iterations in which the residues are considered one by one.

During the iterations, we monitor the total \(-\log P\) value of the evolving sequence, which is defined as

\[
-\log P_{\text{total}} = \sum_{i=1}^{L} -\log P_{i},
\]

where \( L \) is the length of the sequence, \( i \) is the residue index, \( a_i \) is the index of the sidechain type at position \( i \) in the current sequence, and \( P_{i} \) is the probability of the corresponding sidechain type predicted by the encoder-decoder network according to the current local environment of residue \( i \) (which depends on the current sidechain types of \( i \)'s neighboring residues). Please note that these \(-\log P\) values are not related to the \( P \)-values considered in common statistical hypothesis tests. For simplicity, we have used a fixed number of 1,000 self-consistent iteration steps to perform all of the design runs in this study. It has been retrospectively confirmed that in all of the design runs, the lowest \(-\log P\) values were always
DNA sequences of designed proteins were constructed into the Ndel and XhoI sites of pET-22b (+) by General Biotech and TsingKe Biotech. Plasmids encoding the designs were transformed into E. coli BL21 (DE3) cells. Protein expression was induced at OD₆₀₀nm=0.7 with 1 mM IPTG for 20 h at 16 °C. For 1H, 15N HSQC-NMR study, uniformly 15N-labeled proteins were prepared by growing the bacteria in inorganic medium (24 g/l KH₂PO₄, 5 g/l NaOH, 0.5 g/l NH₄Cl, 2.2 mM MgSO₄, 0.1 mM CaCl₂, and 2.5 g/l glucose) using 1H,15N-Cl as isotope source. Cells were harvested and sonicated in buffer containing 20 mM Tris and 500 mM NaCl at pH 7.8. The solute supernatant was purified by Ni⁺⁺ affinity chromatography and concentrated in buffer containing 20 mM Tris, 500 mM NaCl and 1 mM EDTA, pH 7.8 and then subjected to size-exclusion chromatography in a Superdex 75 column with the ÄKTA purifier system (GE Healthcare).

The thermal stability of designed proteins were evaluated by nano differential scanning calorimetry (nanoDSC) (TA Instruments). The protein samples were prepared in buffer containing 300 mM NaCl, 20 mM Tris, 1 mM EDTA, pH 8.0 with a protein concentration of 3–6 mg/ml. The protein melting temperatures were monitored in a 0.33 ml cell from 25 to 125 °C at a heating rate of 1 °C/min. Responses of proteins were measured by heating and scanning for three times. The nanoDSC scans were background-corrected and analyzed with Launch NanoAnalyze software.

All NMR data were collected at 298 K on a Bruker DMRX600 spectrometer equipped with triple resonances, self-shielded z-axis gradient probes. Two-dimensional HSQC-NMR samples typically contained 0.35–0.5 mM 15N-labeled proteins, 25 mM NaH₂PO₄, 150 mM NaCl, 2 mM EDTA (pH 6.9) and 10% (v/v) D₂O. Data were processed using the programs NMRDraw/NMRPipe and SPARKY. Crystallographic screening was conducted at 298 K by the hanging-drop vapor diffusion method using various screening kits. Purified and concentrated proteins (15–20 mg/ml) were used for crystallization. The crystals for data collection were obtained in 24–48 h in PEG 1500 for 1r26-A3, 2.5 M ammonium sulfate and 0.1 M Bis-Tris propane, pH 7.0 for 1r26-A6, 2% v/v 14-dioxane, 0.1 M Bicine pH 9.0, 10% w/v PEG 2000 for 1r26-A7, 27% w/v PEG 2000 MME and 0.1 M sodium cacodylate, pH 6.5 for 1r26-B4. Crystals of 1Cy5-A7 appeared in seven days in buffer containing 12% w/v polyethylene glycol 3,350 and 4% v/v Tacsimate, pH 7.0. All crystals were shortly soaked in reservoir solution supplemented with 30% glycerol (v/v) and then flash frozen in liquid nitrogen. The diffusion data were collected on BL19U1 and BL02U1 beamlines to 0.9785 to 0.9791 Å. Datasets were processed by using HKL2000 program for 1r26-A3, 1r26-A6, 1r26-A7 and XDS program for 1Cy5-A7 and 1r26-B4. The designed structures served as search model for molecular replacement with MOLREP. The final structures were refined by PHENIX. The statistics for data collection and structural refinement are summarized in Supplementary Table 14.

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

Data availability
The following data are available from Zenodo: complete lists of proteins for training and testing the models; the amino acid sequences designed for the 1000 pairs by Model₁; the amino acid sequence and DNA sequences of the experimentally examined proteins. The experimentally solved protein structures have been deposited in the PDB under accession codes: 7VQL (1r26-A3, 10.2210/pdb7VQL/pdb); 7VQQ (1r26-A6, 10.2210/pdb7VQQ/pdb); 7VQW (1r26-A7, 10.2210/pdb7VQW/pdb); 7VYT (1Cy5-A7, 10.2210/pdb7VYT/pdb); 7VUX (1r26-B4, 10.2210/pdb7VUX/pdb). Source Data are provided with this paper.

Code availability
The source code is available from Code Ocean at https://doi.org/10.24433/CO.3351944.v1.

Received: 27 December 2021; Accepted: 7 June 2022; Published: 21 July 2022.

References
1. Kuhlman, B. & Bradley, P. Advances in protein structure prediction and design. Nat. Rev. Mol. Cell Biol. 20, 681–697 (2019).
2. Huang, P.-S., Boyken, S. E. & Baker, D. The coming of age of de novo protein design. Nature 537, 320–327 (2016).
3. Silva, D.-A. et al. De novo design of potent and selective mimics of IL-2 and IL-15. Nature 565, 186–191 (2019).
4. Klein, L. et al. De novo design of covalent SARS-CoV-2 mini-spike inhibitors. Science 370, 426–431 (2020).
5. Siegel, J. B. et al. Computational design of an enzyme catalyst for a stereospecific bimolecular Diels–Alder reaction. Science 329, 309–310 (2010).
6. Cui, Y. et al. Development of a versatile and efficient C–N lyase platform for asymmetric hydroamination via computational enzyme redesign. Nat. Catal. 4, 364–373 (2021).
7. Kuhlman, B. et al. Design of a novel globular protein fold with atomic-level accuracy. Science 302, 1364–1368 (2003).
performed experimental analyses under the supervision of Q.C. and H.Y.L. M.Z., C.C.W. and F.D.L. analyzed the crystallographic data. J.H.Z. collected and processed NMR data. Y.F.L., W.L.W. and H.Y.L. wrote the paper with input from all of the other authors.

**Competing interests**
H.Y.L. Q.C., H.Q.L., Y.F.L. and W.L.W. have filed patent application (no. 202210091553.7) relating to rotamer-free protein sequence design in the name of University of Science and Technology of China. The other authors declare no competing interests.

**Additional information**
The online version contains supplementary material available at https://doi.org/10.1038/s43588-022-00273-6.
Corresponding author(s): Houqiang Li; Quan Chen; Haiyan Liu
Last updated by author(s): Jun 2, 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.

- 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
- 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
Protein structures for training the models have been downloaded from the Protein Data Bank. PISCES server (http://dunbrack.fccc.edu/Guoli/pisces_download.php) was used to select non-redundant data with a resolution cutoff of 2.5 Angstrom and a sequence-identity cutoff of 50%. NMR data were collected using Bruker’s standard NMR software TopSpin 4.0.9. The X-ray diffraction data were collected on BL19U1 and BL02U1 beamlines at 0.9785 to 0.9791 Angstrom.

Data analysis
NMR data were processed using the software NMRDraw/NMRPipe (Version 8.2) and SPARKY 3.115. Crystallographic data of 1r26-A3, 1r26-A6 and 1r26-A7 were processed using the HKL2000 software (Version 718). Data sets of 1cy5-A7 and 1r26-B4 were processed using XDS (Version Feb 5, 2021). Model rebuilding was performed in MOLREP (Version 11.7.03) and the final structures were refined by PHENIX 1.14-3260. Structure figures were made with PyMOL version 1.8. The DSC spectra data were processed with Launch NanoAnalyze software (Version 3.8.0). Training of the neural network models have been performed with the PyTorch machine learning package version 1.7. Rosetta energy calculations were performed using the Rosetta software suite version 3.12.
For structure prediction, source code for the AlphaFold model, trained weights, and inference script were used (https://github.com/deeptmind/alphafold).
We used HHBlits from hh-suite v3.0-beta.314/07/2017 (https://github.com/soedinglab/hh-suite).
Source codes and pre-trained parameter of the home-made programs [ABACUS-R Version 1.0] for protein sequence design described in the manuscript are available from https://doi.org/10.24433/CO.3351944.v1 for free non-commercial use.

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.
Protein structures for training or testing the models have been downloaded from the Protein Data Bank. The protherm dataset were downloaded from http://gibk26.bse.kyutech.ac.jp/jouhou/Protherm/protherm.html. Complete lists of proteins for training and testing model in comparison of sequence recovery rate with Ingraham et al. and Anand et al. were respectively downloaded from http://people.csail.mit.edu/ingraham/graph-protein-design/data and https://console.cloud.google.com/storage/browser/seq-des-data.

The experimentally solved protein structures have been deposited in the Protein Data Bank with accession codes 7VQL (1r26-A3), 7VQV (1r26-A6), 7VQW (1r26-A7), 7VTY (1cy5-A7) and 7VU4 (1r26-B4).

The following additional data is available from https://doi.org/10.5281/zenodo.6592054: source data for Figures 2 to 4; PDB-format X-ray structures for 1r26-A3, 1r26-A6, 1r26-A7, 1cy5-A7 and 1r26-B4, which have been used to produce figures of protein structures in Figure 5 and in Supplementary Information; complete lists of proteins for training and testing the models; the amino acid sequences designed for the 100 targets by Model_eval; the amino acid sequences and DNA sequences of the experimentally examined proteins; source data for Supplementary Figures 1a, 1b, 1c, 2a, 2b, 3a, 3c, 5a, 7a, 7b, 7c and 11.

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 statistical methods were used to predetermine sample sizes for experimental testing. The sample sizes have been mainly determined based on available human capacity and experimental resources. The first batch of experiments examined 27 proteins designed for natural backbone (the PDB IDs of the targets are 1r26, 1cy5, and 1ubq). The size exclusion chromatography results and the 1H NMR spectra have covered all the purified proteins in the first batch. Five proteins covering the three target backbones have been measured by DSC experiments. High resolution structures of four proteins, including three designed for the 1r26 backbone and one designed for the 1cy5 backbone, have been solved with X-ray crystallography. Among the 30 designed proteins analyzed in the second batch, 25 led to successful protein expression in E.coli, and 23 could be purified in soluble form. The purified proteins have been subjected to the same types of experimental analysis as the first batch protein. One high resolution structure for a second batch protein has been solved.

Data exclusions

No data were excluded.

Replication

Protein expression and solubility was tested once or twice. All attempts to replicate expression and solubility screening experiments for further experimental characterization were successful. The X-ray structures were determined based on single crystals using standard procedures which have internal statistical validations.

Randomization

There was no randomized sample allocation in this work. All tested protein designs received identical treatment.

Blinding

Blinding is not relevant to our study because there is no group allocation.

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.

Ecalorical, 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 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

- Antibodies
- Eukaryotic cell lines
- Palaeontology and archaeology
- Animals and other organisms
- Human research participants
- Clinical data
- Dual use research of concern

Methods

- 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

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 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.
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 | Specify voxel-wise or cluster-wise and report all relevant parameters for cluster-wise methods. |
| (See Eklund et al. 2016) | | | |
| 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 and predictive analysis |
|---------------------------|-----------------------------------------|----------------|---------------------------------------------|
|                           | Report the measures of dependence used and the model details (e.g. Pearson correlation, partial correlation, mutual information). |
|                           | 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.). |
|                           | Specify independent variables, features extraction and dimension reduction, model, training and evaluation metrics. |