Supporting Information

Table of Contents

Supporting Texts

Supporting Tables

Supporting Figures

References

Supporting Texts

Text S1: Monomer Evolutionary Energy Calculation

The monomer evolutionary energy, $E_{evoMonomer}$, is calculated as the best match between the designed sequence and the scaffold structure using the Needleman-Wunsch dynamic programming (DP) algorithm [1]. More specifically, a 2D DP matrix, $D(i,j)$, is defined where i and j are the positions along the designed and scaffold sequences, respectively. The value of $D(i,j)$ is equal to the ending value of the best path with the highest matching score towards the lattice (i,j). Here, a path in the matrix corresponds to an alignment between the designed and scaffold sequences. Thus, the $E_{evoMonomer}$ is the value at $D(L_1, L_2)$, where L_1 and L_2 are the lengths of the designed and scaffold sequences, respectively. Note, in EvoDesign L_1 and L_2 are equivalent, although it can be potentially applied to design sequences with a different length from the scaffold. The DP procedure allows gaps in the alignment between the designed and scaffold sequences, depending on the alignment score.

Given a gap penalty scheme of $w(k) = g_o + (k-1)g_e$, where k is the gap length, and g_o and g_e are the gap opening and gap extension penalties, respectively, the initialization of the DP matrix can be written as

$$\begin{cases} D(0,0) = 0 \\ D(0,j) = j * g_e \quad for\; 0 < j \; \leq L_2 \\ D(i,0) = i * g_e \quad for\; 0 < i \; \leq L_1 \end{cases} \tag{S1}$$

The remaining elements in the DP matrix are calculated by the recurrence equation:

$$D(i,j) = \max \begin{cases} D(i-1, j-1) + E_{match}(i,j) \\ \max\limits_{1 \leq k \leq i} [D(i-k, j) + w(k)] \\ \max\limits_{1 \leq k \leq j} [D(i, j-k) + w(k)] \end{cases} \tag{S2}$$

where the matching score between i and j is defined by

$$E_{match}(i,j) = M(j, aa_i) + w_1 \Theta_{SS}(i,j) + w_2 \Theta_{SA}(i,j) + w_3 \Theta_{\phi}(i,j) + w_4 \Theta_{\psi}(i,j) \tag{S3}$$

Here, aa_i is the amino acid for the i^{th} residue of the designed sequence and $M(j, aa_i)$ is the structural profile, represented by an $L_2 \times 20$ matrix, specifically, $M(j, aa_i) = \sum_{x=1}^{20} B(aa_i, x) H(j, x)$. Here, $B(aa_i, x)$ is the BLOSUM62 mutation score for mutating aa_i to amino acid x [2]. Additionally, $H(j,x) = \sum_{m=1}^{f_x^j} h(m)$, where f_x^j is the frequency with which amino acid x appears at the j^{th} position of the multiple sequence alignment (MSA) that was constructed by TM-align [3] by structurally searching the scaffold against the PDB library. Lastly, $h(m)$ is the Henikoff weight of the m^{th} template sequence in the MSA. The higher (more positive) the value of $M(j, aa_i)$, the more favorable the mutation is between residue i of the designed sequence and residue j of the scaffold protein.

The remaining terms in Eq. S3 measure the local structural similarities between the designed sequence and the scaffold protein. The secondary structure (SS), solvent accessibility (SA), and backbone torsional angles (ϕ, ψ) for the designed sequence are predicted using the fast machine learning-based methods described previously [4], while those for the scaffold structure are assigned by DSSP [5]. More specifically, these terms are defined as follows:

$$\begin{cases} \Theta_{SS}(i,j) = \begin{cases} 1, & \text{if } SS(i) = SS(j) \\ 0, & \text{else if } SS(i) \text{ or } SS(j) \text{ is coil} \\ -1, & \text{otherwise} \end{cases} \\ \Theta_{SA}(i,j) = \begin{cases} 1, & \text{if } SA(i) = SA(j) \\ 0, & \text{else if } SA(i) \text{ or } SA(j) \text{ is intermediate} \\ -1, & \text{otherwise} \end{cases} \\ \Theta_{\phi}(i,j) = \dfrac{-\min\{|\phi(i) - \phi(j)|, 360 - |\phi(i) - \phi(j)|\}}{180} \\ \Theta_{\psi}(i,j) = \dfrac{-\min\{|\psi(i) - \psi(j)|, 360 - |\psi(i) - \psi(j)|\}}{180} \end{cases} \quad (S4)$$

Here, SS is divided into three states: α-helix, β-strand or coil. Additionally, SA is categorized into three states: buried, intermediate or exposed based on its depth in the protein structure. The values for the weights w_1, w_2, w_3, and w_4 are 1.58, 2.45, 1.00, and 1.00, respectively, which are proportional to the relative accuracy of the SS, SA, and ϕ/ψ feature predictors for a set of 625 non-redundant training proteins [6].

Text S2: Interface Evolutionary Energy Pseudocount

To offset the smaller size of the interface library, a pseudocount was introduced into the evolution-based interface potential by the BindProfX approach [7]:

$$N_{pseudo}(aa_i, i) = N_{fix} + N_{gap} + N_{evo} = 25 + 15n_{gap}(i) + 5\sum_{x=1}^{20} \frac{N_{obs}(x,i)}{N_{tot}} M(x, aa_i) \quad (S5)$$

where the first term, N_{fix}, is a constant parameter whose value is set to 25. The second term, N_{gap}, is a gap dependent pseudocount that is proportional to the number of gaps, $n_{gap}(i)$, at the i^{th} position of the iMSA. The final term is the evolutionary pseudocount, N_{evo}, which takes into account amino acids that are related to the wild type and mutant residues in the interface alignment. $\frac{N_{obs}(x,i)}{N_{tot}}$ is the frequency with which an amino acid x appears at position i in the iMSA and $M(x, aa_i)$ is the interface probability transition matrix score for amino acid x mutating to residue aa_i.

Since the iMSA contains homologous sequences only from the PDB, its depth depends on the number of interface structural homologs detected. We previously found [7] that the average number of interface structural homologs was around five. This is much smaller than the size of the pseudocounts, indicating that the pseudocounts are quite important to calibrate the amino acid occurrence probabilities. The overall Pearson correlation coefficient (PCC) between experimental and predicted $\Delta\Delta G_{binding}^{WT\rightarrow mut}$ values was 0.685 for the BindProfX benchmark on the overall dataset. However, for those targets with only one or two structurally similar interfaces, the PCC was 0.207 without pseudocounts, indicating that the amino acid occurrence probabilities were unreliable when there were too few interface homologs. With pseudocounts applied, the PCC increased from 0.207 to 0.323.

Text S3: Dataset Construction and EvoEF Parameter Optimization

To compute the energy of a protein, EvoEF splits the total energy into the sum of three parts: the non-bonded atomic interactions within a residue ($E_{intraResidue}$), between different residues within the same chain ($E_{interResidueSameChain}$), and between different residues from different chains ($E_{interResidueDiffChain}$), i.e.,

$$\begin{aligned} E_{EvoEF} &= E_{intraResidue} + E_{interResidueSameChain} + E_{interResidueDiffChain} - E_{ref} \\ &= \{E_{vdw} + E_{elec} + E_{HB} + E_{solv}\}_{intraResidue} \\ &\quad +\{E_{vdw} + E_{elec} + E_{HB} + E_{solv}\}_{interResidueSameChain} \\ &\quad +\{E_{vdw} + E_{elec} + E_{HB} + E_{solv}\}_{interResidueDiffChain} \\ &\quad -E_{ref} \end{aligned} \qquad (S6)$$

where E_{vdw}, E_{elec}, E_{HB} and E_{solv} are the same as defined in Eqs. (3-8) in the main text. Overall, EvoEF uses eight energy terms each for $E_{interResidueSameChain}$ and $E_{interResidueDiffChain}$, and only six terms for $E_{intraResidue}$, as intra-residue E_{HBss} and E_{HBbb} do not exist. Thus, there are a total of 56 parameters that need to be optimized in EvoEF, including 8 weights for $E_{intraResidue}$, 14 weights for $E_{interResidueSameChain}$, 14 weights for $E_{interResidueDiffChain}$, and 20 amino acid reference energies.

Text S3.1 Dataset construction

We used two types of experimental data, based on the mutation-induced protein stability and binding affinity changes, to train and test EvoEF. The mutation-induced protein stability change data were collected from the FoldX [8] and STRUM [9] datasets, which contain 1,056 and 3,421 mutants, respectively. After filtering out the duplicated mutants in identical structures, a total of 3,989 non-redundant mutants from 210 proteins were retained, where 3,978 were single mutations and 11 were multiple mutations. Half of the 3989 mutants were randomly selected as the training set (with 1995 mutants) and the other half as the testing set (with 1994 mutants).

Here, we note that the FoldX dataset has an overrepresentation of mutations from larger residues to smaller ones. Out of the 1,056 data samples, 1,015 are from larger-sized amino acids to smaller ones, while only 41 are from smaller to larger-sized amino acids. This trend is less obvious in the STRUM dataset, where 2,568 out of 3,421 mutation samples are from larger to smaller amino acids and 853 are from smaller to larger amino acids. The bias present in the FoldX dataset may result in overestimation of the mutation correlations. For example, the Pearson correlation coefficient between the predicted and experimental stability change data for the FoldX potential on the FoldX dataset is 0.688, which is reduced to 0.446 for the STRUM dataset.

For the second set of benchmark data, experimental mutation-induced binding affinity changes were collected from the SKEMPI v2.0 database [10], which contains 7,085 mutation entries in total. The training and test datasets were constructed as follows. First, we discarded mutants whose corresponding structures contained three or more chains. Second, we removed mutants with non-interface residues. Here, an interface residue is defined as a residue that has at least one heavy atom within 5.0 Å of the other chain in a protein complex. When there were multiple entries for the same mutant, the average $\Delta\Delta G_{binding}^{WT\rightarrow mut}$ value was calculated. After filtering the dataset, a total of 2,204 mutants from 177 protein-protein interfaces were retained. Again, half the 2,204 mutants were randomly selected as the training set (with 1102 mutants) and the other half as the test set (with 1102 mutants).

In order to predict the binding affinity and stability change upon mutation, the native structures were minimized, and the mutant models were generated using the following steps (the information for each command can be found in Text S5):

Step 1: For $\Delta\Delta G_{stability}^{WT\rightarrow mut}$ and $\Delta\Delta G_{binding}^{WT\rightarrow mut}$ predictions, we extracted the single target chain or the two target amino acid chains, respectively, from the PDB file of the crystal structure and discarded water molecules and ligands that were not amino acids.

Step 2: We optimized the structure of the wild type protein/complex using EvoEF's "RepairStructure" command as follows:

```
./EvoEF --command=RepairStructure --pdb=wildtype.pdb
```

Following this command, the minimized wild type protein/complex was output into a file named 'wildtype_Repair.pdb' and this minimized model was used as the initial structure to build the mutant model.

Step 3: We built a structural model of the mutant protein/complex using EvoEF's "BuildModel" command as follows:

```
./EvoEF --command=BuildMutant --pdb=wildtype_Repair.pdb --mutant-file=individual_list.txt
```

The file "individual_list.txt" contained the list of mutation(s). Following this command, a new file "wildtype_Repair_Mutant_1.pdb" was generated, which contained the modelled mutant structure.

Step 4: We computed the stability of the wild type and mutant proteins using EvoEF's "ComputeStability" command as follows:

```
./EvoEF --command=ComputeStability --pdb=wildtype_Repair.pdb
./EvoEF --command=ComputeStability --pdb=wildtype_Repair_Mutant_1.pdb
```

Or

We computed the binding affinity of the wild type and mutant complexes using EvoEF's "ComputeBinding" command as follows:

```
./EvoEF --command=ComputeBinding --pdb=wildtype_Repair.pdb
./EvoEF --command=ComputeBinding --pdb=wildtype_Repair_Mutant_1.pdb
```

The above steps were used to minimize/construct the models and predict either the stability or binding affinity during EvoEF's training/testing. However, to benchmark EvoEF against FoldX and to avoid potential bias in the scoring, for the FoldX tests, we minimized the structures using FoldX. The following steps were used to build the structural models and predict the stability/binding affinity energies for FoldX:

Step 1: For $\Delta\Delta G_{stability}^{WT\rightarrow mut}$ and $\Delta\Delta G_{binding}^{WT\rightarrow mut}$ predictions, we extracted the single target chain or the two target amino acid chains, respectively, from the PDB file of the crystal structure and discarded water molecules and ligands that were not amino acids.

Step 2: We optimized the structure of the wild type protein/complex using FoldX's "RepairPDB" command:

```
./foldx --command=RepairPDB --pdb=wildtype.pdb
```

After this step, the minimized wild type protein/complex was output into a file named ‘wildtype_Repair.pdb’ and this minimized model was used as the initial model to build the mutant model.

Step 3: We built a structural model of the mutant protein using FoldX’s “BuildModel” command:

```
./foldx --command=BuildModel --pdb=wildtype_Repair.pdb --mutant-file=individual_list.txt
```

Here, “individual_list.txt” was a text file that contained the specified mutation(s). After this step, two files “WT_wildtype_Repair_1.pdb” and “wildtype_Repair_1.pdb” were generated. The former file was the wildtype structure with additional structural optimization, while the latter one was the mutant structure. Normally, “WT_wildtype_Repair_1.pdb” was the same as “wildtype_Repair.pdb”, and if not, their difference were quite small.

Step 4: We computed the stability of the wild type and mutant proteins using FoldX’s “Stability” command:

```
./foldx --command=Stability --pdb=WT_wildtype_Repair_1.pdb
./foldx --command=Stability --pdb=wildtype_Repair _1.pdb
```

Or

We computed the binding affinity of the wild type and mutant complexes using FoldX’s “AnalyseComplex” command:

```
./foldx --command=AnalyseComplex --pdb=WT_wildtype_Repair_1.pdb
./foldx --command=AnalyseComplex --pdb=wildtype_Repair_1.pdb
```

The stability and binding affinity change datasets, as well as the predicted ΔΔGs by FoldX and EvoEF can be found at: https://zhanglab.ccmb.med.umich.edu/EvoDesign/EvoEFBenchmark.tar.gz.

Text S3.2 Optimization of reference energies and weights for $E_{interResidueSameChain}$ and $E_{intraResidue}$

The amino acid reference energies and the weighting factors for $E_{intraResidue}$ and $E_{interResidueSameChain}$ were determined based on the stability change data ($\Delta\Delta G_{stability}^{WT\rightarrow mut}$) of monomeric proteins upon mutation. The protein stability change due to mutation is computed by

$$\Delta\Delta G_{stability}^{WT\rightarrow mut} = \Delta G_{stability}^{mut} - \Delta G_{stability}^{WT} = E_{EvoEF}^{mut} - E_{EvoEF}^{WT} \quad (S9)$$

where the wild type and mutant structural models are required to compute the physical energies. To this end, we first performed local energy minimization on the native crystal structures using the EvoEF energy minimizer and then built mutant models based on the minimized wild type structures using the steps described above. For doing so, EvoEF first scans the wild type structure in the order of amino acid occurrence and then optimizes the amino acid side-chains one-by-one. Several minimization cycles can be performed for the sake of convergence, but the default number of cycles is set to one. Based on our test, there’s not a large difference in the minimized structures when we set the number of cycles to two or more. To remove the possible steric clashes during the minimization procedure, EvoEF searches alternative rotameric conformations from a backbone-independent rotamer library obtained from Xiang and Honig [11]. The rotamer library contains 984 rotamers for the 20 amino acid types, and 1,007 rotamers if two tautomers of histidine are considered. In the library, the hydroxyl groups of serine, threonine and tyrosine are rotated to expand their rotamers by six, six, and two folds, respectively. Asparagine, histidine and glutamine are also flipped to construct better hydrogen bonding networks during energy minimization. The details of energy minimization, model building and $\Delta\Delta G$ computation can be found in Text S5.

Finally, the reference energies and parameters for $E_{intraResidue}$ and $E_{interResidueSameChain}$ were optimized by minimizing the objective function $F = \sum_i \left(\Delta\Delta G_{i,stability,pred}^{WT\to mut} - \Delta\Delta G_{i,stability,exp}^{WT\to mut}\right)^2$ over a set of experimental protein stability change data, where $\Delta\Delta G_{i,stability,pred}^{WT\to mut}$ and $\Delta\Delta G_{i,stability,exp}^{WT\to mut}$ were the predicted and experimental data for the i^{th} mutation in the dataset. More specifically, the objective function can be written as:

$$\begin{aligned} F &= \sum_i \left(\Delta\Delta G_{i,stability,pred}^{WT\to mut} - \Delta\Delta G_{i,stability,exp}^{WT\to mut}\right)^2 \\ &= \sum_i \left[\left(\sum_j \omega_j \Delta\Delta G_{i,stability,pred}^{WT\to mut}(j) + E_{ref}^{WT} - E_{ref}^{mut}\right) - \Delta\Delta G_{i,stability,exp}^{WT\to mut}\right]^2 \end{aligned} \qquad (S7)$$

where $\Delta\Delta G_{i,stability,pred}^{WT\to mut}(j)$ was the EvoEF predicted stability change upon mutation for the j^{th} energy term, not considering the reference energy. $E_{ref}(WT)$ was the summed reference energy for the wild type sequence and $E_{ref}(mut)$ was the summed reference energy for the mutant sequence. This is essentially a least squares optimization problem, which can be easily solved using simple algorithms such as least squares fitting, gradient descent and conjugated gradient methods. However, we found that the optimal weights for some terms decided by these methods could be negative and theoretically meaningless. Therefore, we implemented a Metropolis Monte Carlo procedure to re-optimize the parameters. During the procedure, the movement consisted of random changes to the parameters while the weights were restricted to be greater than or equal to zero. Parameter changes were accepted and rejected based on the Metropolis criterion, where *F* was the energy. The final reference energies and weights were chosen from the parameter set with the lowest *F* value over the training set.

Text S3.3 Optimization of weights for $E_{interResidueDiffChain}$

One of the major goals of this work is to extend the EvoDesign pipeline to design protein-protein interactions. To achieve the best performance in computing the physical interactions in protein-protein interfaces, we used experimental binding affinity change ($\Delta\Delta G_{binding}^{WT\to mut}$) data to train the weights for $E_{interResidueDiffChain}$. In EvoEF, the binding energy of a protein complex for scaffold *A* and its binding partner *B* is computed by

$$\Delta G_{binding} = E_{AB} - E_A - E_B \qquad (S8)$$

where E_{AB} , E_A and E_B are the stability scores for the complex and component monomers, respectively. The binding free energy change due to mutation is then written as

$$\Delta\Delta G_{binding}^{WT\to mut} = \Delta G_{binding}^{mut} - \Delta G_{binding}^{WT} \qquad (S9)$$

The parameters for $E_{interResidueDiffChain}$ were decided by minimizing the objective function $\sum_i \left(\Delta\Delta G_{i,binding,pred}^{WT\to mut} - \Delta\Delta G_{i,binding,exp}^{WT\to mut}\right)^2$ over the training set of experimental binding affinity change data, where $\Delta\Delta G_{i,binding,pred}^{WT\rightarrow mut}$ and $\Delta\Delta G_{i,binding,exp}^{WT\rightarrow mut}$ were the predicted and experimental data, respectively, for the i^{th} mutation in the SKEMPI training set described above. The same Metropolis Monte Carlo procedure was used to decide the parameters for $E_{interResidueDiffChain}$ as was used to train the parameters for $E_{interResidueSameChain}$ and $E_{intraResidue}$ as well as the reference energies. During the Monte Carlo search, the 42 previously optimized parameters were fixed.

A list of the optimized weight parameters and reference energies are given in Tables S1 and S2 separately.

Text S4: Decoy Discrimination

In order to further validate EvoEF, we assessed its ability to discriminate native structures from decoy structures for the 200 non-redundant monomeric proteins in the 3DRobot Decoy Set [12]. For each protein, 300 decoys were generated by 3DRobot. The root mean squared deviations (RMSDs) of the structural decoys to the native ranged from 0 to 12 Å. We did two types of decoy discrimination tests: (1) discriminating the native structures from decoys according to the folding stability energy, and (2) discriminating near-native decoy structures (low RMSD decoys) from those with high RMSDs.

In the first test, EvoEF correctly ranks the native protein as the lowest energy for each of the 200 individual decoy sets, while FoldX does so in 198 cases. We also computed the Z-score of the native structure in each decoy set:

$$Z_{native} = \frac{\langle E\rangle - E_{native}}{\delta E} \qquad (S10)$$

where E_{native} is the energy of the native structure, and $\langle E\rangle$ and δE are the average and standard deviation of the energy function for all the structures in the decoy set. For EvoEF, the Z-score has value ranging from 2.25 to 8.09 for the 200 structures, while the Z-score for FoldX ranges from 2.41 to 7.60. Since the Z-score has a high significance (>2) for all the cases, both FoldX and EvoEF were able to discriminate the native from non-native decoys with a sufficient gap for all the decoy sets, although EvoEF has a slightly larger variation range in Z-score.

For the second test, we computed the Z-score of the near-native decoy structures, i.e., those with low RMSDs:

$$Z_{nnative} = \frac{\langle E\rangle_h - \langle E\rangle_l}{\delta E_h} \qquad (S11)$$

where $\langle E\rangle_l$ is the average energy for the 10% of decoys that have the lowest RMSD; $\langle E\rangle_h$ and δE_h are the average and standard deviation of the energy function for all the rest of the structures in the decoy set. Since both low and high RMSD decoys are generated *in silico* and thus have similar local structural errors, it is much harder to recognize the near-native structures than to recognize the native structure that was determined experimentally and usually has idealized local structural features and side-chain packing. The average $Z_{nnative}$ for EvoEF was 1.959 with values ranging from 0.32 to 3.57, while that for FoldX was 1.844 with values ranging from 0.40 to 2.83. In 198 cases, EvoEF has a $Z_{nnative} > 1$, while FoldX has a $Z_{nnative} > 1$ in 193 of the cases. These data suggest that EvoEF has a relatively better ability to recognize near-native structures from high RMSD structural decoys.

Here, the decoys datasets were taken directly from the work of Deng et al. [12], which can be downloaded at: https://zhanglab.ccmb.med.umich.edu/3DRobot/decoys/. The decoy recognition data for EvoEF and FoldX can be found at: https://zhanglab.ccmb.med.umich.edu/EvoDesign/EvoEFBenchmark.tar.gz.

Text S5: Commands in EvoEF

Several commands have been implemented in EvoEF, such as “RepairStructure”, “BuildMutant”, “ComputeStability”, “ComputeBinding”, and “OptimizeHydrogen”, to facilitate the use of the force field.

Generally, these commands are performed using the following syntax:

```
EvoEF --command=commandName --pdb=your.pdb [other options]
```

Here, we describe the details of each of these commands.

Text S5.1 Energy Minimization

Energy minimization in EvoEF is performed using the command “RepairStructure”. Usually, the user-provided structural model or even the crystal structure can have steric clashes or bad hydrogen bonding networks. Moreover, sometimes side-chain atoms can be missing from the structural model. Therefore, it is important to fix the structure and do energy minimization to optimize the rotameric side-chain conformations for the clashed amino acids. Essentially, the global optimization of the amino acid side-chain conformations requires complete repacking of the side-chains, but this is not trivial. Instead of doing full side-chain repacking, EvoEF does fast local optimization of the initial model, either a crystal structure or a model predicted by structure modeling software, to remove steric clashes as much as possible. To do so, EvoEF first scans the user-input structure in the order of amino acid occurrence and then optimizes the amino acid side-chains one-by-one. To remove the possible steric clashes in the user-provided structural model, EvoEF searches rotameric conformations from a backbone-independent rotamer library obtained from the work of Xiang and Honig [11]. The rotamer library contains 984 rotamers for 20 amino acid types, and 1,007 rotamers if two tautomers of histidine are considered. In the library, the hydroxyl groups of serine, threonine and tyrosine are rotated to expand their rotamers by six, six, and two folds, respectively. The asparagine, histidine and glutamine amino acids are also flipped for better hydrogen bonding evaluation. Several cycles of energy minimization can be performed for the sake of convergence and the best minimization results, but the default number of cycles is set to one. Based on our benchmarking, the difference between one or two or more minimization cycles is not that significant. The syntax to do energy minimization in EvoEF is:

```
EvoEF --command=RepairStructure --pdb=model.pdb
```

Successful execution of this command will generate a new structure file named “*model_Repair.pdb*”. In the minimized model, the optimized polar hydrogen coordinates are also shown.

Text S5.2 Model Builder

To compute the protein stability and binding affinity changes due to mutation, we need the experimental structure and the mutant model. Here, the experimental structure should be minimized as mentioned in the above section. We build a mutant model starting from the minimized wild type structure and mutate the amino acids at the specified positions one-by-one. During the mutation process, the amino acid side-chain conformations within 6 Å of each mutated position are repacked to alleviate possible steric clashes and optimize the local energies. The rotameric conformations for repacking and mutation are also taken from the above Xiang and Honig rotamer library [11]. The mutated structure is first built, then three cycles of local energy minimization are performed. The syntax to build mutant models in EvoEF is:

```
EvoEF --command=BuildMutant --pdb=model_Repair.pdb --mutant-file=individual_list.txt
```

where "*model_Repair.pdb*" is the minimized initial model, and "*individual_list.txt*" is a text file that specifies the desired mutation(s). In "*individual_list.txt*", the mutations must be presented in the following format:

```
CA171A,DB180E;
```

Each mutation is written in one line ending with ";", and multiple mutants are divided by ",". Note that there are no gaps/spaces between single mutations. For each single mutation, the first letter is the native amino acid, the second is the identifier for the chain that the amino acid appears on, the number is the amino acid's position in the chain, and the last letter is the mutant amino acid. Running the command successfully should generate a new structure file named "*model_Repair_Mutant_1.pdb*". In the mutant model, the optimized polar hydrogen coordinates are also shown.

Text S5.3 Energy Computation

In EvoEF, the protein stability energy can be calculated using the following command:

```
EvoEF --command=ComputeStability --pdb=your.pdb
```

Furthermore, binding affinity for protein-protein complexes can be calculated using the command:

```
EvoEF --command=ComputeBinding --pdb=complex.pdb
```

The energies for each term and the total energy will be output if the command is run successfully.

Text S5.4 Other commands

In the initial protein structures, such as the crystal structure or models obtained by different structure modelling software, polar hydrogens are usually not provided. However, the positions of polar hydrogens are important to model and calculate hydrogen bonding energy, which is crucial for the structural specificity that underlies protein folding, function, and interactions. Although backbone or side-chain polar hydrogens of some amino acid types can be determined by standard topologies from force fields such as CHARMM19 [13] and AMBER [14], the hydroxyl groups of serine, threonine and tyrosine are rotatable and the hydrogen positions cannot be decided by the topologies. In EvoEF, we implemented another command "OptimizeHydrogen" to find the hydrogen positions that optimize the hydrogen bonding network. Specifically, we build the rotamers for serine, threonine and tyrosine using their native conformations and expand the number of rotamers considered by rotating the hydroxyl groups.

Supporting Tables

Table S1. Weight Parameters for the EvoEF Force Field.

Classification	**Energy terms**	**Weight**
Intra-residue interaction	Van der Waals attractive	0.0000
	Van der Waals repulsive	0.1200
	Coulomb's electrostatics	0.0000
	desolvP	0.0000
	desolvH	0.0000
	HBsb_dist	0.0000
	HBsb_theta	0.0400
	HBsb_phi	0.0012
Inter-residue interactions in the same chain	Van der Waals attractive	0.6076
	Van der Waals repulsive	0.4968
	Coulomb's electrostatics	0.0000
	desolvP	0.3008
	desolvH	0.0322
	HBbb_dist	0.3554
	HBbb_theta	0.0000
	HBbb_phi	0.0000
	HBsb_dist	0.5452
	HBsb_theta	0.2080
	HBsb_phi	0.1600
	HBss_dist	0.3036
	HBss_theta	0.0800
	HBss_phi	0.0000
Inter-residue interactions in different chains	Van der Waals attractive	0.6384
	Van der Waals repulsive	0.7904
	Coulomb's electrostatics	0.0000
	desolvP	0.4048
	desolvH	0.3432
	HBbb_dist	0.0000
	HBbb_theta	0.7600
	HBbb_phi	0.5000
	HBsb_dist	0.0000
	HBsb_theta	0.6552
	HBsb_phi	0.5236
	HBss_dist	0.7200
	HBss_theta	0.6720
	HBss_phi	0.5040

Table S2. Amino Acid Reference Energies Decided for EvoEF Force Field.

Amino acid	**Reference energy**	**Amino acid**	**Reference energy**
ALA	1.20	MET	1.00
CYS	0.60	ASN	1.00
ASP	1.20	PRO	0.96
GLU	1.00	GLN	1.40
PHE	2.00	ARG	0.72
GLY	2.00	SER	1.20
HIS	2.20	THR	0.80
ILE	-0.12	VAL	0.24
LYS	1.20	TRP	2.60
LEU	0.00	TYR	1.60

Supporting Figures

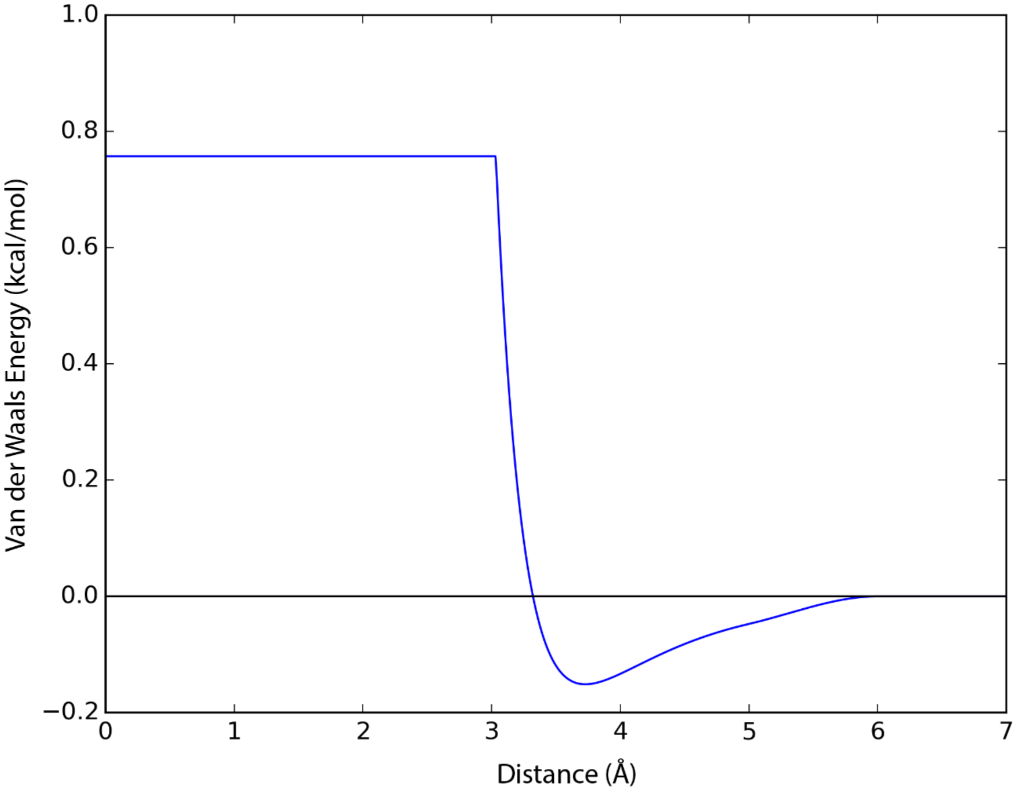
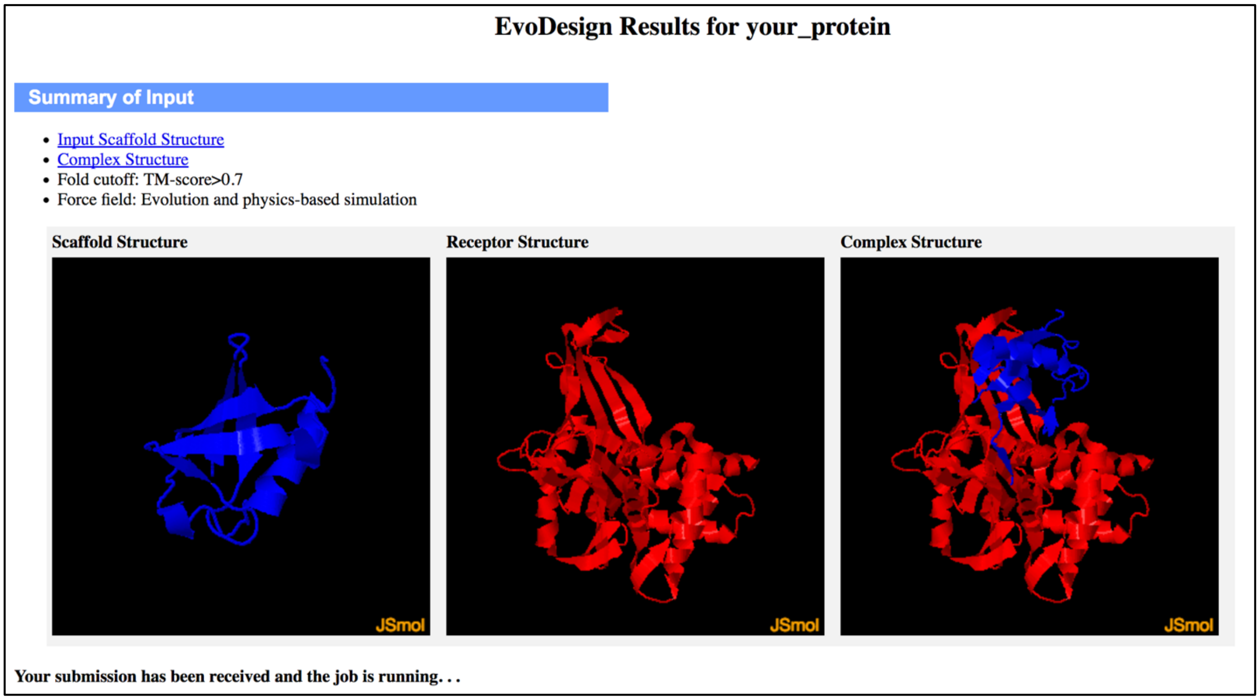


Figure S1: Shape of the Van der Waals Energy Between an Amide N and a Carbonyl C. The van der Waals radii for the amide N and carbonyl C are 1.7632 and 1.9649 Å, respectively, while their corresponding well-depths are 0.1617 and 0.1418 kcal/mol, respectively.



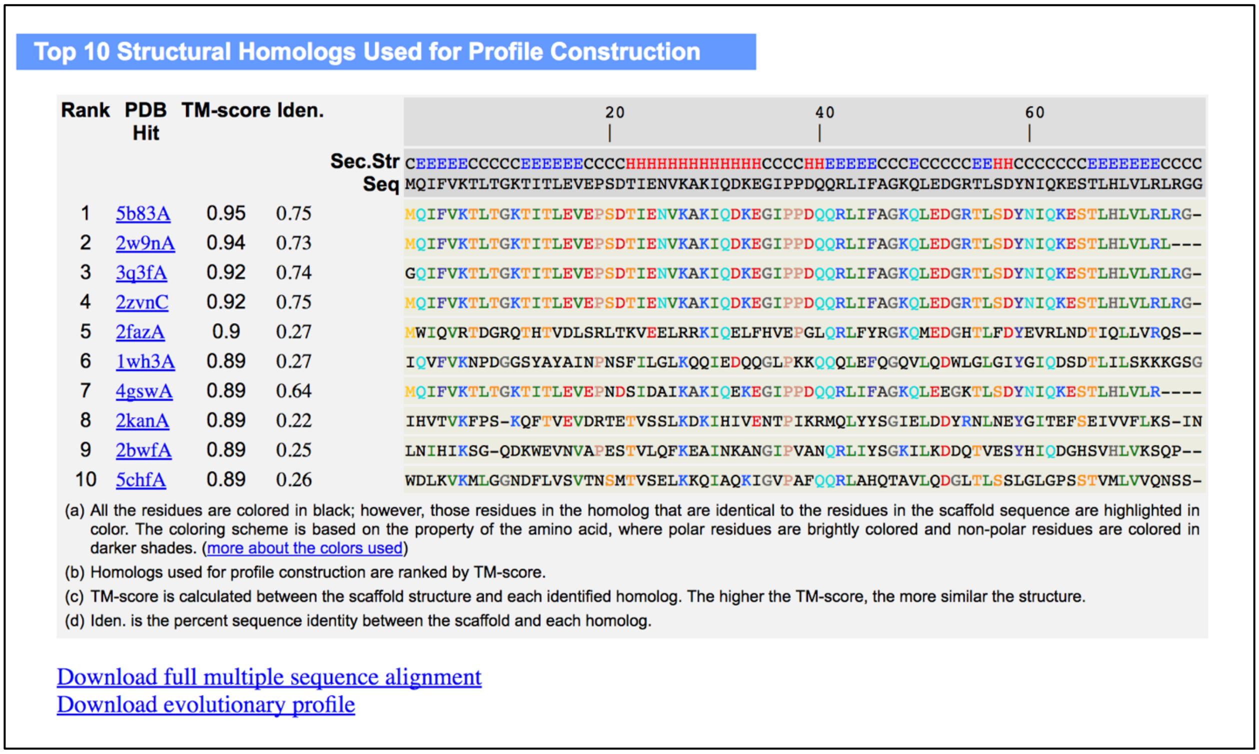
EvoDesign Results for your_protein

Summary of Input

- Input Scaffold Structure
- Complex Structure
- Fold cutoff: TM-score>0.7
- Force field: Evolution and physics-based simulation

Your submission has been received and the job is running. . .

Figure S2: Illustration of the Input Summary Section in the EvoDesign Output Page. The first section of the EvoDesign results page is a summary of the input, which is created immediately after submission of a job. It contains a description of the structural similarity cutoff (TM-score) used during the evolutionary profile construction along with a description of the force field used by the design simulation. Additionally, a link is provided to download the input scaffold structure and, in the case of interface design, the complex structure. If the user opts to upload the scaffold and receptor structures separately and dock them together, the complex structure will be available to download upon completion of docking. For monomer design, if the input scaffold structure is a Cα trace, a full-atomic model will be generated using REMO [15]. The full-atomic model is then uploaded to the server, replacing the initial Cα trace model. The scaffold, receptor and complex structures are visualized using the interactive JSmol applet.



Top 10 Structural Homologs Used for Profile Construction

Rank	PDB Hit	TM-score	Iden.	20 40 60
			Sec.Str	CEEEEECCCCCEEEEEECCCCHHHHHHHHHHHHHCCCCHHEEEEECCCECCCCCEEHHCCCCCCCEEEEEEECCCC
			Seq	MQIFVKTLTGKTITLEVEPSDTIENVKAKIQDKEGIPPDQQRLIFAGKQLEDGRTLSDYNIQKESTLHLVLRLRGG
1	5b83A	0.95	0.75	MQIFVKTLTGKTITLEVEPSDTIENVKAKIQDKEGIPPDQQRLIFAGKQLEDGRTLSDYNIQKESTLHLVLRLRG-
2	2w9nA	0.94	0.73	MQIFVKTLTGKTITLEVEPSDTIENVKAKIQDKEGIPPDQQRLIFAGKQLEDGRTLSDYNIQKESTLHLVLRL---
3	3q3fA	0.92	0.74	GQIFVKTLTGKTITLEVEPSDTIENVKAKIQDKEGIPPDQQRLIFAGKQLEDGRTLSDYNIQKESTLHLVLRLRG-
4	2zvnC	0.92	0.75	MQIFVKTLTGKTITLEVEPSDTIENVKAKIQDKEGIPPDQQRLIFAGKQLEDGRTLSDYNIQKESTLHLVLRLRG-
5	2fazA	0.9	0.27	MWIQVRTDGRQTHTVDLSRLTKVEELRRKIQELFHVEPGLQRLFYRGKQMEDGHTLFDYEVRLNDTIQLLVRQS--
6	1wh3A	0.89	0.27	IQVFVKNPDGGSYAYAINPNSFILGLKQQIEDQQGLPKKQQQLEFQGQVLQDWLGLGIYGIQDSDTLILSKKKGSG
7	4gswA	0.89	0.64	MQIFVKTLTGKTITLEVEPNDSIDAIKAKIQEKEGIPPDQQRLIFAGKQLEEGKTLSDYNIQKESTLHLVLR----
8	2kanA	0.89	0.22	IHVTVKFPS-KQFTVEVDRTETVSSLKDKIHIVENTPIKRMQLYYSGIELDDYRNLNEYGITEFSEIVVFLKS-IN
9	2bwfA	0.89	0.25	LNIHIKSG-QDKWEVNVAPESTVLQFKEAINKANGIPVANQRLIYSGKILKDDQTVESYHIQDGHSVHLVKSQP--
10	5chfA	0.89	0.26	WDLKVKMLGGNDFLVSVTNSMTVSELKKQIAQKIGVPAFQQRLAHQTAVLQDGLTLSSLGLGPSSTVMLVVQNSS-

(a) All the residues are colored in black; however, those residues in the homolog that are identical to the residues in the scaffold sequence are highlighted in color. The coloring scheme is based on the property of the amino acid, where polar residues are brightly colored and non-polar residues are colored in darker shades. (more about the colors used)

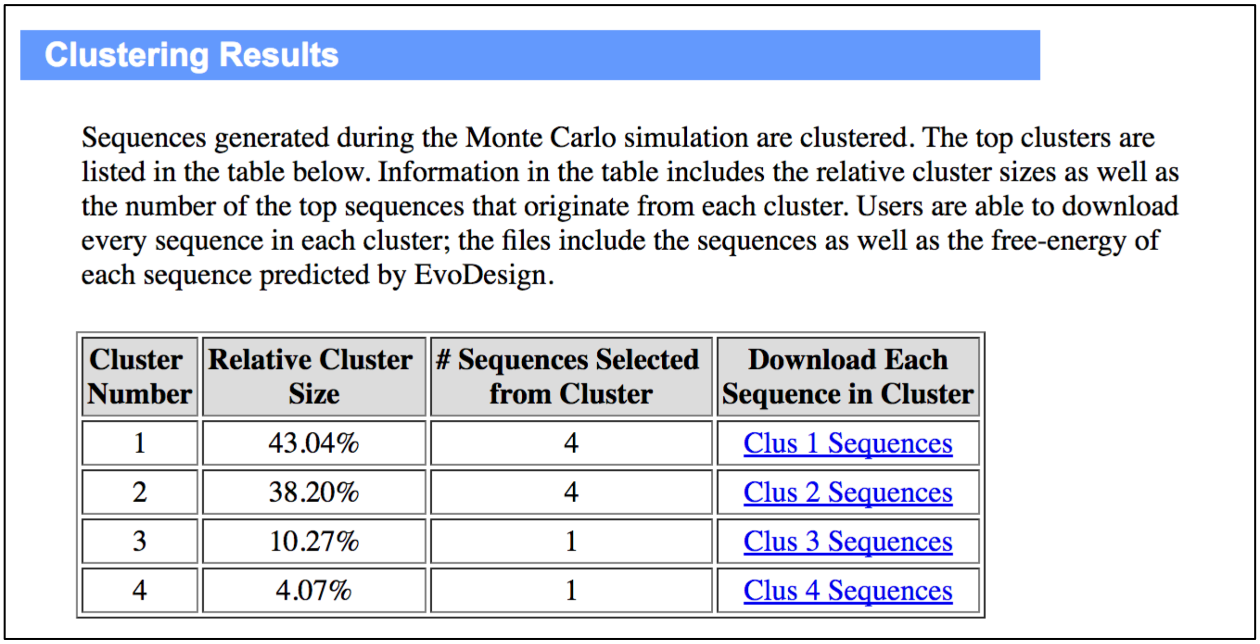
(b) Homologs used for profile construction are ranked by TM-score.

(c) TM-score is calculated between the scaffold structure and each identified homolog. The higher the TM-score, the more similar the structure.

(d) Iden. is the percent sequence identity between the scaffold and each homolog.

Download full multiple sequence alignment
Download evolutionary profile

Figure S3: Summary of the Top Homologs Used for Profile Construction. To generate the evolutionary profiles, structural and interface homologs are identified from the PDB and protein interaction libraries. Although all homologous proteins with TM-scores higher than the specified cutoff are used for monomer profile construction, only the top ten structural homologs, which are sorted by TM-score to the scaffold structure, are displayed in this section. The information displayed for each homolog includes: (*i*) the homolog PDB ID and the link to download the structure, (*ii*) the TM-score and sequence identity to the scaffold, and (*iii*) the alignment between the scaffold and the homolog. Moreover, links are provided to download the full multiple sequence alignment and the evolutionary profile used to guide the design simulation.

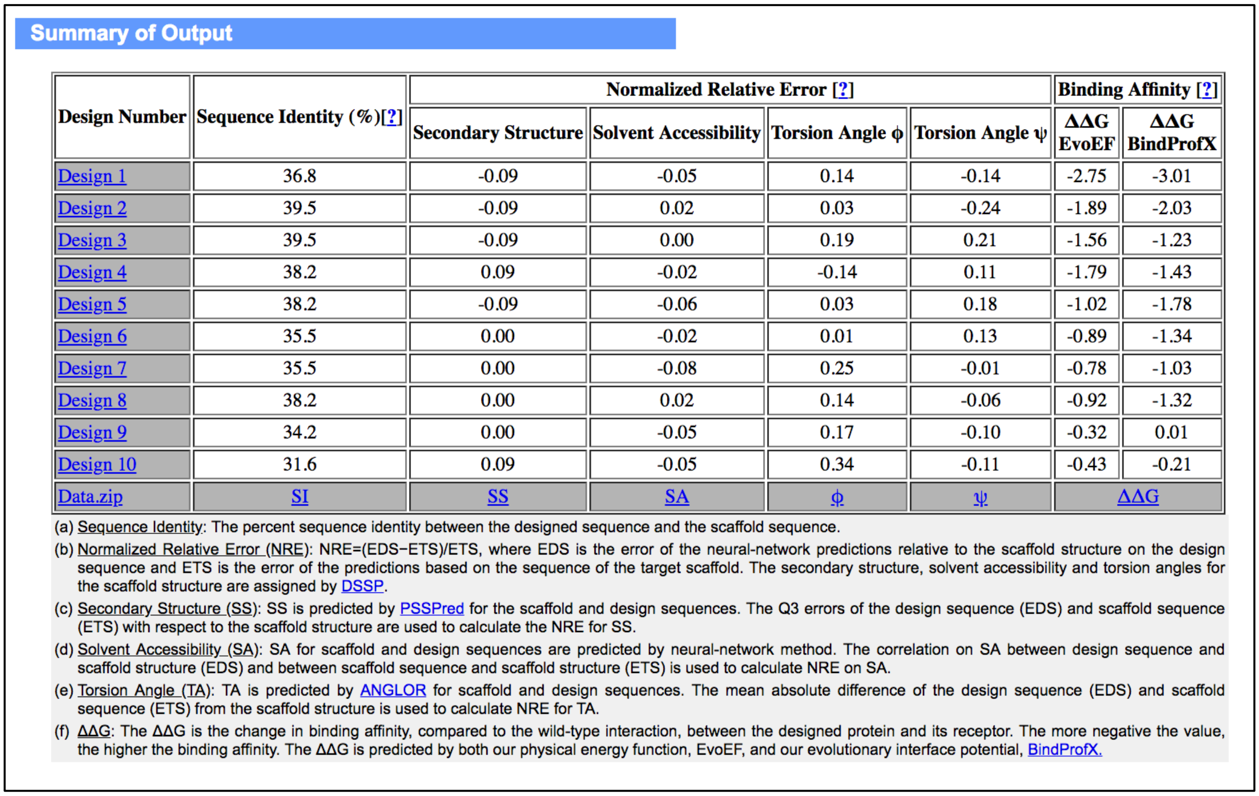


Clustering Results

Sequences generated during the Monte Carlo simulation are clustered. The top clusters are listed in the table below. Information in the table includes the relative cluster sizes as well as the number of the top sequences that originate from each cluster. Users are able to download every sequence in each cluster; the files include the sequences as well as the free-energy of each sequence predicted by EvoDesign.

Cluster Number	Relative Cluster Size	# Sequences Selected from Cluster	Download Each Sequence in Cluster
1	43.04%	4	Clus 1 Sequences
2	38.20%	4	Clus 2 Sequences
3	10.27%	1	Clus 3 Sequences
4	4.07%	1	Clus 4 Sequences

Figure S4: Clustering Results. During the Monte Carlo simulation, many designed sequences are generated. After the simulaion is completed, the generated sequences are clustered using SPICKER [16] based on the distance scaled by their BLOSUM62 sequence similarity. For each target, EvoDesign outputs ten designed sequences, which are selected after clustering. The number of sequences selected from each cluster depends on the cluster size. For example, if 70% of the sequence decoys are contained in the first cluster, 7 sequences would be selected from the first cluster. For each cluster, the sequence in the cluster center is selected first, followed by selection of the non-redundant and lowest-energy sequences. Here, the non-redundant sequence identity cutoff is equal to 70%. The first column of the table lists the cluster number. The relative size of each cluster and the number of sequences selected from each cluster are displayed in columns 2 and 3, respectively. The last column contains links to download text files containing each sequence in the cluster. The files contain the sequences and the EvoDesign calculated energy of each sequence.



Summary of Output

Design Number	Sequence Identity (%)[?]	Normalized Relative Error [?]				Binding Affinity [?]	
		Secondary Structure	Solvent Accessibility	Torsion Angle ϕ	Torsion Angle ψ	ΔΔG EvoEF	ΔΔG BindProfX
Design 1	36.8	-0.09	-0.05	0.14	-0.14	-2.75	-3.01
Design 2	39.5	-0.09	0.02	0.03	-0.24	-1.89	-2.03
Design 3	39.5	-0.09	0.00	0.19	0.21	-1.56	-1.23
Design 4	38.2	0.09	-0.02	-0.14	0.11	-1.79	-1.43
Design 5	38.2	-0.09	-0.06	0.03	0.18	-1.02	-1.78
Design 6	35.5	0.00	-0.02	0.01	0.13	-0.89	-1.34
Design 7	35.5	0.00	-0.08	0.25	-0.01	-0.78	-1.03
Design 8	38.2	0.00	0.02	0.14	-0.06	-0.92	-1.32
Design 9	34.2	0.00	-0.05	0.17	-0.10	-0.32	0.01
Design 10	31.6	0.09	-0.05	0.34	-0.11	-0.43	-0.21
Data.zip	SI	SS	SA	ϕ	ψ	ΔΔG	

(a) Sequence Identity: The percent sequence identity between the designed sequence and the scaffold sequence.

(b) Normalized Relative Error (NRE): NRE=(EDS−ETS)/ETS, where EDS is the error of the neural-network predictions relative to the scaffold structure on the design sequence and ETS is the error of the predictions based on the sequence of the target scaffold. The secondary structure, solvent accessibility and torsion angles for the scaffold structure are assigned by DSSP.

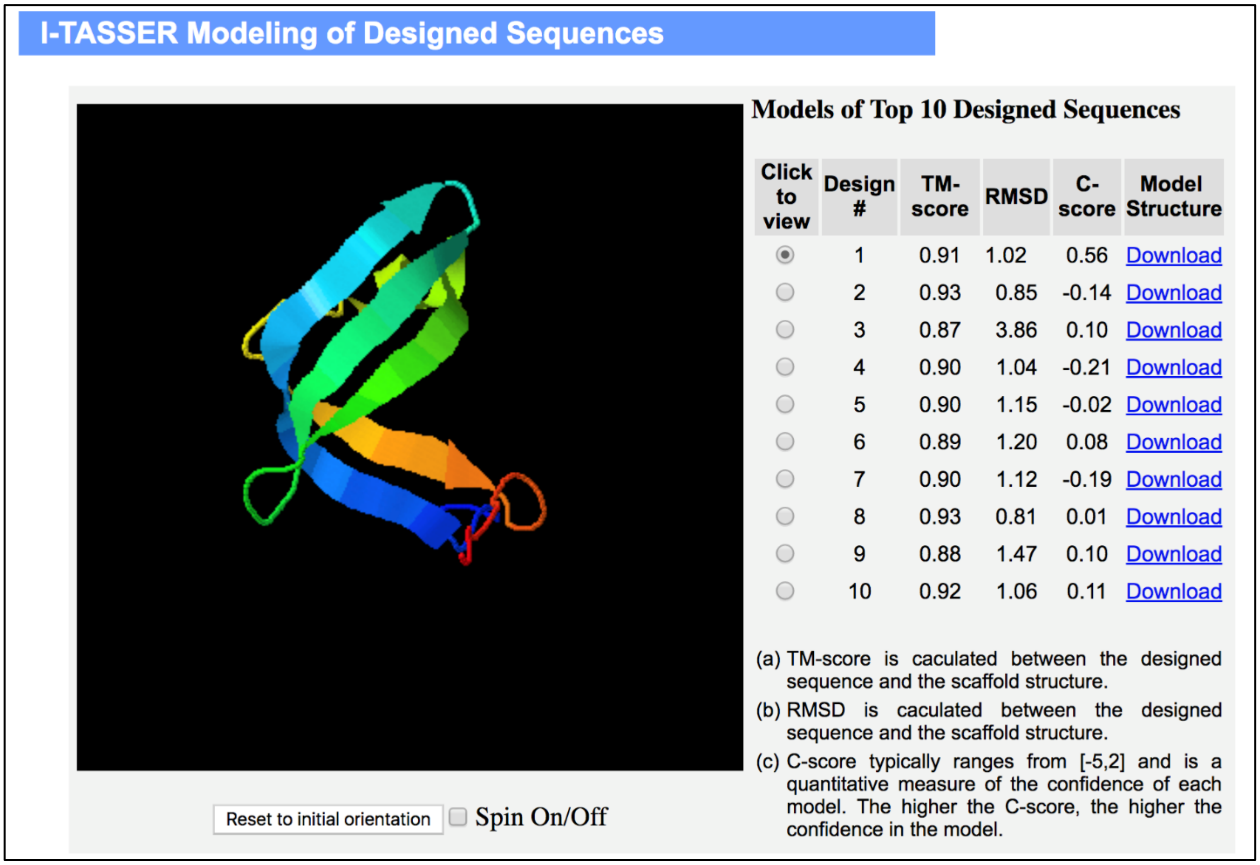
(c) Secondary Structure (SS): SS is predicted by PSSPred for the scaffold and design sequences. The Q3 errors of the design sequence (EDS) and scaffold sequence (ETS) with respect to the scaffold structure are used to calculate the NRE for SS.

(d) Solvent Accessibility (SA): SA for scaffold and design sequences are predicted by neural-network method. The correlation on SA between design sequence and scaffold structure (EDS) and between scaffold sequence and scaffold structure (ETS) is used to calculate NRE on SA.

(e) Torsion Angle (TA): TA is predicted by ANGLOR for scaffold and design sequences. The mean absolute difference of the design sequence (EDS) and scaffold sequence (ETS) from the scaffold structure is used to calculate NRE for TA.

(f) ΔΔG: The ΔΔG is the change in binding affinity, compared to the wild-type interaction, between the designed protein and its receptor. The more negative the value, the higher the binding affinity. The ΔΔG is predicted by both our physical energy function, EvoEF, and our evolutionary interface potential, BindProfX.

Figure S5 Results Table. A summary of the results of the local structural analysis is provided for the top ten designed sequences in tabular form. Here, the secondary structure, solvent accessibility and torsion angles of the designed and scaffold sequences are predicted using three neural-network programs, namely, PSSpred [17], Solve [18], and Anglor [19]. The normalized relative errors (NRE) are provided for each of the features (Columns 3-6) to give an approximate assessment of the local structural quality of the designed sequences. The NRE is an error measure for the local structural feature predictors for each designed sequence relative to the scaffold sequence: $NRE = (EDS - ETS)/ETS$. EDS (EDS stands for 'error of designed sequence') is the error of prediction for the designed sequence relative to that assigned by the DSSP program [5] on the scaffold structure, and ETS (ETS stands for 'error of target sequence') is the error of the prediction for the target scaffold sequence. Thus, a small (or negative) NRE value indicates that the designed sequence has a relatively small (or even smaller) prediction error than the scaffold sequence, while a large NRE usually signals a bad design due to the large prediction error relative to the scaffold sequence. Links are provided at the bottom of the table to download the secondary structure, solvent accessibility, and backbone torsional angle prediction data for the designed sequences. The first column of the table contains links so that users can download each of the designed sequences. In addition, the binding energy change of the designed proteins compared to the native scaffold are calculated by EvoEF and BindProfX and are listed in the results table. This should help provide information on how the altered interfaces affect the binding affinities compared to the wild type proteins. All of the information can be downloaded as a compressed file under the link to 'Data.zip'.



I-TASSER Modeling of Designed Sequences

Models of Top 10 Designed Sequences

Click to view	Design #	TM-score	RMSD	C-score	Model Structure
●	1	0.91	1.02	0.56	Download
○	2	0.93	0.85	-0.14	Download
○	3	0.87	3.86	0.10	Download
○	4	0.90	1.04	-0.21	Download
○	5	0.90	1.15	-0.02	Download
○	6	0.89	1.20	0.08	Download
○	7	0.90	1.12	-0.19	Download
○	8	0.93	0.81	0.01	Download
○	9	0.88	1.47	0.10	Download
○	10	0.92	1.06	0.11	Download

(a) TM-score is caculated between the designed sequence and the scaffold structure.

(b) RMSD is caculated between the designed sequence and the scaffold structure.

(c) C-score typically ranges from [-5,2] and is a quantitative measure of the confidence of each model. The higher the C-score, the higher the confidence in the model.

Figure S6: I-TASSER Modeling of the Top Ten Designed Sequences. If the user selects the option, the structures of the top ten designed sequences are modeled with the I-TASSER pipeline. For each model, a confidence score (C-score) of the folding simulation is calculated by $C\text{-}score = \ln\left(\frac{M/M_{tot}}{\langle RMSD\rangle}\cdot\frac{1}{N}\sum_{i=1}^{N}\frac{Z(i)}{Z_{0(i)}}\right)$, where M/M_{tot} is the fraction of the structure decoys generated by I-TASSER in the largest structure cluster, and $\langle RMSD\rangle$ is the average RMSD of the decoys to the cluster center. This term corresponds to the degree of convergence of the folding simulations. $Z(i)/Z_0(i)$ is the normalized significance score of the templates by the i^{th} threading program, where there are a total of *N* threading program used by I-TASSER for template identification. The C-score is normally in the range of [-5, 2], and a C-score >-1.5 usually indicates that the I-TASSER model has a correct fold with a TM-score >0.5 [20]. Since not all designs can be folded by I-TASSER with a high confidence, the C-score can be used as an approximate assessment of the foldability of the designed sequences. In a large-scale experiment that examined the folding of designed sequences [4], it was shown that there is a strong correlation between the C-score of I-TASSER simulations and the folding rate of designed proteins, where 80% (or 100%) of designed sequences are experimentally foldable for sequences with an I-TASSER C-score >0 (or >0.8). In this figure, the TM-scores and RMSDs are between the I-TASSER model and the starting scaffold. Users are able to download the I-TASSER models from the provided links.

References

[1] Needleman SB, Wunsch CD. A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology. 1970;48:443-53.

[2] Henikoff S, Henikoff JG. Amino-Acid Substitution Matrices from Protein Blocks. P Natl Acad Sci USA. 1992;89:10915-9.

[3] Zhang Y, Skolnick J. TM-align: a protein structure alignment algorithm based on the TM-score. Nucleic Acids Res. 2005;33:2302-9.

[4] Mitra P, Shultis D, Zhang Y. EvoDesign: De novo protein design based on structural and evolutionary profiles. Nucleic acids research. 2013;41:W273-80.

[5] Kabsch W, Sander C. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers. 1983;22:2577-637.

[6] Mitra P, Shultis D, Brender JR, Czajka J, Marsh D, Gray F, et al. An Evolution-Based Approach to De Novo Protein Design and Case Study on Mycobacterium tuberculosis. PLoS computational biology. 2013;9:e1003298.

[7] Xiong P, Zhang C, Zheng W, Zhang Y. BindProfX: Assessing Mutation-Induced Binding Affinity Change by Protein Interface Profiles with Pseudo-Counts. Journal of molecular biology. 2017;429:426-34.

[8] Guerois R, Nielsen JE, Serrano L. Predicting Changes in the Stability of Proteins and Protein Complexes: A Study of More Than 1000 Mutations. Journal of Molecular Biology. 2002;320:369-87.

[9] Quan L, Lv Q, Zhang Y. STRUM: structure-based prediction of protein stability changes upon single-point mutation. Bioinformatics. 2016;32:2936-46.

[10] Jankauskaite J, Jimenez-Garcia B, Dapkunas J, Fernandez-Recio J, Moal IH. SKEMPI 2.0: An updated benchmark of changes in protein-protein binding energy, kinetics and thermodynamics upon mutation. Bioinformatics. 2018.

[11] Xiang Z, Honig B. Extending the accuracy limits of prediction for side-chain conformations. J Mol Biol. 2001;311:421-30.

[12] Deng H, Jia Y, Zhang Y. 3DRobot: automated generation of diverse and well-packed protein structure decoys. Bioinformatics. 2016;32:378-87.

[13] Brooks BR, Bruccoleri RE, Olafson BD, States DJ, Swaminathan S, Karplus M. CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. Journal of Computational Chemistry. 1983;4:187-217.

[14] Case DA, Cheatham TE, Darden T, Gohlke H, Luo R, Merz KM, et al. The Amber biomolecular simulation programs. Journal of computational chemistry. 2005;26:1668-88.

[15] Li Y, Zhang Y. REMO: A new protocol to refine full atomic protein models from C-alpha traces by optimizing hydrogen-bonding networks. Proteins. 2009;76:665-76.

[16] Zhang Y, Skolnick J. SPICKER: A clustering approach to identify near-native protein folds. J Comput Chem. 2004;25:865-71.

[17] Yan R, Xu D, Yang J, Walker S, Zhang Y. A comparative assessment and analysis of 20 representative sequence alignment methods for protein structure prediction. Sci Rep. 2013;3:2619.

[18] Chen H, Zhou HX. Prediction of solvent accessibility and sites of deleterious mutations from protein sequence. Nucleic acids research. 2005;33:3193-9.

[19] Wu S, Zhang Y. ANGLOR: a composite machine-learning algorithm for protein backbone torsion angle prediction. PloS one. 2008;3:e3400.

[20] Yang J, Yan R, Roy A, Xu D, Poisson J, Zhang Y. The I-TASSER Suite: protein structure and function prediction. Nature Methods. 2015;12:7-8.