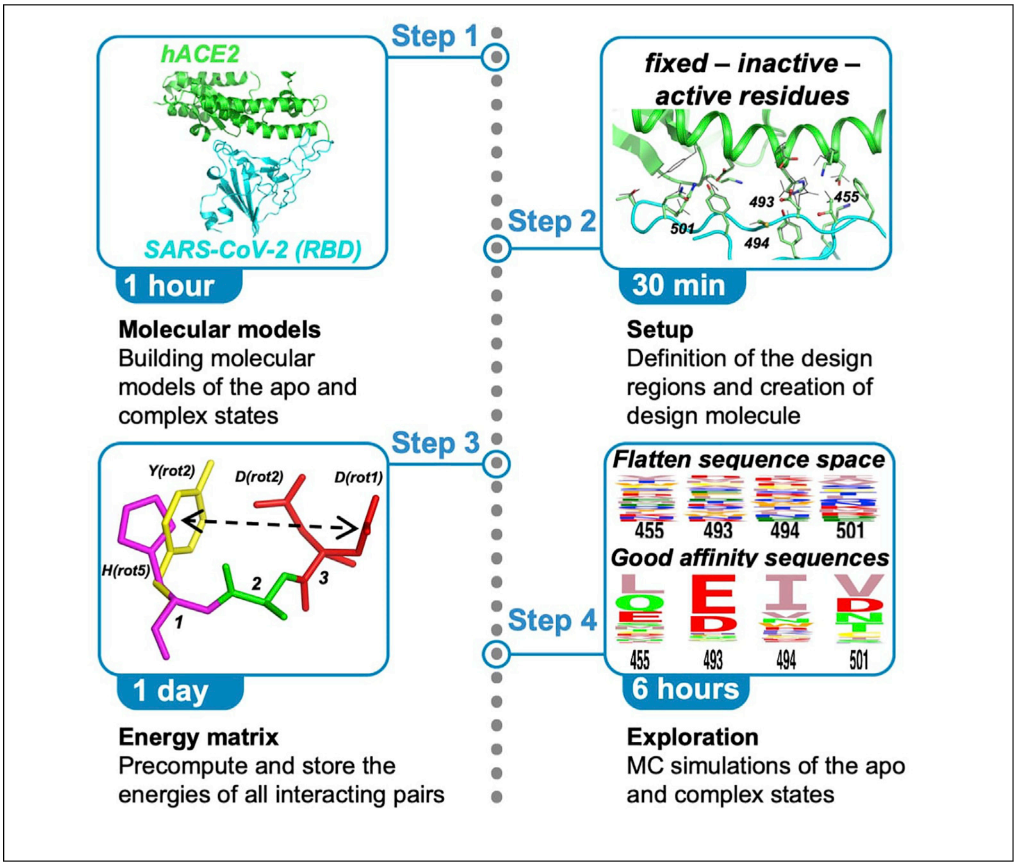
STAR Protocols

CellPress
OPEN ACCESS

Protocol

A computational protein design protocol for optimization of the SARS-CoV-2 receptor-binding-motif affinity for human ACE2


Step 1
hACE2
SARS-CoV-2 (RBD)
1 hour
Molecular models
Building molecular models of the apo and complex states
Step 2
fixed – inactive – active residues
493
455
501
494
30 min
Setup
Definition of the design regions and creation of design molecule
Step 3
Y(rot2)
D(rot2)
D(rot1)
H(rot5)
1
2
3
1 day
Energy matrix
Precompute and store the energies of all interacting pairs
Step 4
Flatten sequence space
455
493
494
501
Good affinity sequences
455
493
494
501
6 hours
Exploration
MC simulations of the apo and complex states

The present protocol describes the computational design of the SARS-CoV-2 receptor binding motif (RBD) to identify mutations that can potentially improve binding affinity for the human ACE2 (hACE2) receptor. We focus on four positions located at the interface with the hACE2 receptor in the RBD:hACE2 complex. We conduct the design with a high-throughput computational protein design (CPD) program, Proteus, incorporating an adaptive Monte Carlo (MC) protocol that promotes the selection of sequences with good binding affinities.

Savvas Polydorides, Georgios Archontis

phpgps1@ucy.ac.cy (S.P.)
archonti@ucy.ac.cy (G.A.)

Highlights

SARS-CoV-2 positions 455, 493, 494, and 501 at the interface with hACE2 are designed

The design uses Proteus, a high-throughput computational protein design program

A physics-based energy function ranks sequences and conformations

An adaptive Monte Carlo protocol promotes the selection of good affinity sequences

Polydorides & Archontis, STAR Protocols 3, 101254
June 17, 2022 © 2022 The Author(s).
https://doi.org/10.1016/j.xpro.2022.101254

STAR Protocols

CellPress
OPEN ACCESS

Protocol

A computational protein design protocol for optimization of the SARS-CoV-2 receptor-binding-motif affinity for human ACE2

Savvas Polydorides[1,2,*] and Georgios Archontis[1,3,*]

[1]Department of Physics, University of Cyprus, Nicosia, Cyprus

[2]Technical contact

[3]Lead contact

*Correspondence: phpgps1@ucy.ac.cy (S.P.), archonti@ucy.ac.cy (G.A.)

https://doi.org/10.1016/j.xpro.2022.101254

SUMMARY

The present protocol describes the computational design of the SARS-CoV-2 receptor binding motif (RBD) to identify mutations that can potentially improve binding affinity for the human ACE2 (hACE2) receptor. We focus on four positions located at the interface with the hACE2 receptor in the RBD:hACE2 complex. We conduct the design with a high-throughput computational protein design (CPD) program, Proteus, incorporating an adaptive Monte Carlo (MC) protocol that promotes the selection of sequences with good binding affinities. For complete details on the use and execution of this protocol, please refer to Polydorides and Archontis (2021).

BEFORE YOU BEGIN

The protocol below describes the specific steps for the computational design of the SARS-CoV-2 receptor binding motif with improved affinity for human ACE2.

The directory tree

Timing: 5–10 min

1. The Proteus 3.0 CPD software is freely available for academic and government scientists from the website address https://proteus.polytechnique.fr. For a detailed description of Proteus and some recent applications refer to (Mignon et al., 2020). Download the software at a location on your computer. Extraction of the software creates the source directory, which contains the programs protX (a Fortran90 program for building the system and computing the interaction energies), protMC (a C Program that reads the interaction energy matrix (IEM) and performs the search in the sequence – structure space) and the following subdirectories: bin (shell, Python and Perl scripts); inp (input files to be used by protX); lib (protX stream and library files), protX/toppar (topology and parameter files for protX), rotamers (rotamer libraries), doc (documentation file) and tutorials (a set of complete applications).

The source directory path is stored in the environment variable *$CPD*. Organize the application main directory and subdirectories.

   a. Define a project directory named *$PROJ* and the subdirectories *$PROJ/APO* and *$PROJ/CMP* for the apo and complex states, respectively.
   b. Create the subdirectories *build*, *lib*, *matrix*, *protMC*, *reconstruct*, *MD* under both *$PROJ/APO* and *$PROJ/CMP*.
   c. Create the subdirectories *dat*, *out*, *err*, *local*, *local/Bsolv*, *local/Chis*, *local/EnrFltr*, *local/Mut and local/Rota* under each *matrix* subdirectory.

The starting model

⏱ Timing: 1–5 min

2. Download the crystallographic structure of the complex between the hACE2 receptor and the SARS-CoV-2 RBD from the Protein Data Bank archive, (PDB: 6M0J).

   Optional: The necessary input for the design is a structural model for the apo and the complex state, e.g., the corresponding crystallographic coordinates. An initial preparation of the system might include the following steps: i) Starting from the crystallographic coordinates, prepare, solvate, equilibrate and run one or more MD simulations in explicit water, to relax the system and produce an ensemble of conformations at thermodynamic equilibrium. ii) Use the coordinates of one or more frames from the MD production stage as a starting model. This initial preparation could be performed with the CHARMM:GUI interface (website: https://www.charmm-gui.org, (Jo et al., 2008)). CHARMM-GUI provides the option to use various force fields, including the AMBER force field used in this study (see below).

   Note: The conformations of the MD trajectories can be grouped into clusters with respect to a similarity measure (e.g., the root-mean-square-difference (rmsd) of the protein backbone coordinates). With this criterion, the backbone coordinate rmsd between conformations in the same (different) cluster is below (above) a specified threshold. Thus, different clusters identify a set of representative, structurally distinct conformations. Clustering can be done with the Wordom program (Seeber et al., 2011).

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
Crystal structure of SARS-CoV-2 spike receptor-binding domain bound with ACE2.	Lan et al. (2020)	https://www.rcsb.org/structure/6M0J PDB: 6M0J
Software and algorithms		
Proteus 3.0 for computational protein design	Mignon et al. (2020)	https://proteus.polytechnique.fr

MATERIALS AND EQUIPMENT

- *Software*
  - Python
  - Perl
  - GNU parallel shell tool
- *Hardware*
  - Workstation with Linux distribution
  - Computational cluster

STEP-BY-STEP METHOD DETAILS

Preparation of the molecular model for the design

⏱ Timing: 1–2 h

Preparation for the design involves building molecular models for the apo and complex states that are compatible with protX. As the first step, generate protein structure file (*allh_model.psf*) and a corresponding coordinate file (*allh_model.pdb*).

1. Edit the coordinate file to:
   a. Remove any remote metal ions, crystallographic water molecules and other molecules, from the binding site.
   b. Build any missing heavy atoms using the CHARMM-GUI interface (Jo et al., 2008).
   c. Rename atoms, residues, disulfide bonds, titratable groups and terminal groups to match the Amber ff99SB force field (Hornak et al., 2006) troubleshooting 1.
   d. Name the hACE2 chain (segment identifier) "PROA" and the SARS-CoV-2 chain "PROB".
   e. Renumber residues to ensure unique residue numbers for each chain. For example, numbering could start at 1,000 for PROA and 2,000 for PROB (PROA contains less than 1,000 residues).
   f. Copy the modified coordinate *file initial_structure.pdb* to the *$PROJ/APO/build* and *$PROJ/CMP/build* subdirectories.
2. Copy the file *$CPD/lib/parameters.str* to the subdirectory *$PROJ/lib.*
3. Edit the file *parameters.str* to specify the main model parameters.
   a. Select the Amber ff99SB force field (Hornak et al., 2006).
   b. Set the Generalized Born (GB) flag and select the Hawkins-Cramer-Truhlar approximation of the GB solvation model (Hawkins et al., 1995). The GB is a many-body function since the GB atomic solvation radii depend on the entire protein geometry. The GB energy can become pairwise-decomposable via the "Native Environment Approximation," which computes the solvation radius of a particular residue with the rest of the molecule kept in its wildtype sequence and native conformation. An exact treatment, referred to as the "Fluctuating Boundary Method" (FBM), is also available (Villa et al., 2017; Archontis and Simonson, 2005).
   c. To optimize the performance of the solvation energy, it is important to combine the selected solvation model with an appropriate value for the protein dielectric constant ε_p. Optimized values can be found in references (Gaillard and Simonson, 2017; Michael et al., 2017). In the present work we set ε_p to 6.8.
   d. Retain the default values of all other parameters, at this stage.

   Note: Instead of Amber ff99SB, the Charmm force field CHARMM19 (Neria et al., 1996) can be employed. A combination of the GB model with the Lazaridis-Karplus (LK) description of the nonpolar solvation contribution is also available.

4. Make copies of the protX input file *build.inp* and shell script *build.sh* from *$CPD/tutorials/tutorials/tuto_PDZ/build* to *$PROJ/build* subdirectory.
5. Edit *build.inp* to set specific parameters of the system.

```
! parameters for the specific application

eval ($name = ``model'')

eval ($pdbfile = ``initial_structure.pdb'')

eval ($chaina = ``PROA'')

eval ($chainb = ``PROB'') !this is chain E of the pdb file 6M0J

! Define disulphide bridges between residues 379 and 432 (PDB

numbering)

patch DISU refe = ``-'' = (segid PROB and resid 379) refe = ``+'' =

(segid PROB and resid 432) end
```

6. Run protX using the script file *build.sh* to produce the coordinate file *allh_model.pdb* and the protein structure file *allh_model.psf* files in protX format.

```
protX < build.inp > build.out
```

Definition of the design regions and creation of a design molecule

⏱ Timing: 30 min

At this stage, we setup the interaction energy matrix (IEM) calculation of the apo and complex systems. We partition each system into three groups: i) ''fixed'' residues, maintaining the same chemical type and conformation during design; ii) ''inactive'' residues that explore conformations from a discrete rotamer database; iii) ''active'' residues that change both chemical type and conformations (this is the target of the design). The partitioning is performed in the following steps:

7. In the *$PROJ/lib* subdirectory
   a. Edit the file *sele.str* to define the sequence/structure space of the design.
      i. Set the native rotamer flag in the *parameters.str* file to ''1'', to include in the rotamer library of active and inactive groups the sidechain conformation(s) seen in the initial structural model (''native rotamers'').
      ii. Define the active region to include positions 455, 493, 494 and 501 in the CoV-2 RBD.
      iii. Select SARS-CoV-2/hACE2 residues within 15 Å of any atom hACE2/SARS-CoV-2 (excluding prolines and glycines) as the inactive region. With the crystallographic structure and the 15 Å cutoff, the resulting inactive region contains 53 SARS-CoV-2 and 108 hACE2 residues.
      iv. By default, the entire protein backbone, terminal groups, cysteine residues forming disulphide bridges, glycine and proline residues and any remaining atoms not included in the previous selections are included in the fixed region.
   b. Edit the file *mutation_space.dat* to define the list of amino acid types sampled at each active position. By default, we allowed positions 455, 493, 494 and 501 to sample 18 natural amino acid types (including the initial wild type): A, I, L, V, M, K, R, D, E, N, Q, C, S, T, F, Y, W and $H(N_{\delta})$. At a later stage we excluded bulky sidechains F, Y, W, H at position 501 due to steric repulsions troubleshooting 2.
   c. Edit the file *phia.str* to set the atomic surface area coefficients of nonpolar, polar, ionic, and aromatic atoms and the protein dielectric constant ε_p. We employ the following values (Gaillard and Simonson, 2017; Michael et al., 2017): $\varepsilon_p = 6.8$, $\sigma_{alk} = -5$ cal/mol, $\sigma_{pol} = -8$ cal/mol, $\sigma_{ion} = -9$ cal/mol, $\sigma_{aro} = -12$ cal/mol.

Note: To generate a more comprehensive IEM, the user can include a larger set of positions in the active and inactive regions at this stage. Later (in the design phase), the user can restrict sampling of sequences and/or conformations to a subset of the initially specified active and inactive residues.

The rotameric library contains a specified number of rotamer conformations for each amino acid type (i.e., one rotamer for alanine, eleven rotamers for asparagine, etc). When the system is partitioned into fixed/inactive/active groups, each position is associated with the appropriate number of rotamers. Setting the native rotamer flag to ''1'' (in the file *parameters.str*) includes the rotamer(s) encountered in the initial structural model, as explained above.

The interaction energy matrix

⏱ Timing: 1 day

Construction of the IEM necessitates the computation and storage of the sidechain-backbone and sidechain-sidechain interaction energies, for all possible pairs of active and inactive chemical types and rotamers (defined in the setup stage). To perform this calculation:

8. Make copies of files *project.sh*, *setup.sh*, *setup.inp* from the source directory to the subdirectory matrix.
   a. Edit *project.sh* to set the environment variable *$CPD* to your local Proteus directory (source directory).

```
> export CPD = /home/username/Programs/Proteus_3.0
```

   b. Edit *setup.inp* to replace the segment identity (segid) definitions "A" and "B" by "PROA" and "PROB", respectively.

```
> ... and (segid PROA or segid PROB)
```

   c. Execute the shell script *setup.sh.* This invokes protX to produce the following necessary files:
      i. The protein structure file *setup.psf* for a "giant molecule" that contains at each active position a number of sidechains representing all chemical types sampled at this position.
      ii. The corresponding coordinate file *setup.pdb.*
      iii. File *position_list.dat* contains the list of active and inactive positions of the system (the total number of pair interactions involving each position is also reported).
      iv. Individual files for all active and inactive positions, listing the allowed amino acid types at each position. These files are stored under the subdirectory *$PROJ/matrix/local/Mut.*
      v. The Generalized Born atomic radii coefficients b_i for the fixed part of the system are precalculated and stored at *matrix/local/Bsolv/bsolv.pdb.*
9. Make copies of the shell scripts *runI.sh* and *runIJ.sh* to the matrix directory.
   a. Execute the shell script *runI.sh*. This invokes protX to compute the diagonal IEM terms (containing the interaction energy of each sidechain with itself and the fixed part). The calculation can be performed sequentially using one CPU core (default), or in parallel using multiple CPU cores (set the desired number of cores as the command line argument). A prerequisite for the parallel calculation is the GNU parallel shell tool.

```
> runI.sh [number of CPU cores]
```

      i. Generalized-Born atomic-radii coefficients (b_i) are precalculated at each position, for each chemical type and rotamer conformation sampled at that position, with the rest of the system kept at the native sequence and conformation (native environment approximation). These coefficients are stored, along with the atomic coordinates of the corresponding rotamer, to distinct pdb files at *matrix/local/Rota.*
      ii. The diagonal IEM elements are computed. The total interaction energies and their decomposition into molecular mechanics terms (bonded, Coulombic, van der Waals) and solvation terms (GB and solvent accessible surface area) are stored in data files at *matrix/dat.*
      iii. The shell script *make_rotamer_space.sh*, invoked by *runI.sh*, checks the diagonal terms of the matrix to identify high-energy rotamer conformations at active and inactive positions. The exclusion of unfavorable rotamers reduces the conformational space of the system (and facilitates convergence of the adaptive flattening procedure during the second step). At position 501, all rotamers of chemical types F, Y, W and H failed to satisfy the energy threshold due to steric clashes and were excluded troubleshooting 2.

iv. A list of all residue pairs to be computed in the next step is created (*pair_list.dat*).

b. Execute the shell script $PROJ/matrix/runIJ.sh to compute the off-diagonal IEM terms via protX.

```
> runIJ.sh [number of CPU cores or PBS queue] pair_list.dat
```

i. At this step, protX computes the sidechain – sidechain interaction energies for all chemical types/rotamers of the various pairs of active and inactive residues. The total interaction energies and the individual molecular-mechanics, Generalized-Born and solvent accessible surface-area energy terms are stored in data files at *matrix/dat.*

ii. After the energy calculation is completed, all data files are concatenated into a big file (*matrix.dat*) and then split into two files containing the diagonal terms (*matrix.bb*) and the off-diagonal terms (*matrix.pw*) to be read in the second step of the protocol.

Note: This calculation is computationally demanding and should be performed in multiple CPU cores. In our case, a cluster of 43 nodes with 424 (Intel Xeon 2.4 GHz) CPU cores was employed. By assigning each pair interaction to one CPU core, the entire step was completed in less than a day.

⚠ CRITICAL: Repeat the IEM calculation for the SARS-CoV-2 RBD apo state. The next step of the protocol involves the independent explorations of the sequence – structure space of the complex and the apo states.

Exploration of sequence/conformation space

⏲ Timing: 6 h

The exploration of the sequence/conformation space is performed in two stages: i) First, we conduct a set of design simulations of the apo state (the free SARS-CoV-2 RBD domain). We use these simulations to derive biasing potentials that render all chemical types at each active position approximately equiprobable. The procedure is analogous to the Wang-Landau approach and has been described in (Villa et al., 2018; Opuu et al., 2020; Mignon et al., 2020). The resulting biasing potentials are approximately equal to the negative folding energies of the apo state sequences. ii) Using these biasing potentials, we conduct biased "production design" simulations of the complex. The biasing potentials subtract the apo-state folding free energies from the complex-state folding free energies, promoting thereby the selection of sequences with good binding affinities. iii) We also conduct a new set of biased simulations for the apo state, with the same biasing potentials. These simulations produce biased probabilities of the various sequences in the apo state. The binding free energy of a sequence, relative to a reference sequence, is computed from the sequence probabilities in the complex and apo state (see Equation (9), Polydorides and Archontis, 2021).

10. Adaptive flattening of the SARS-CoV-2 RBD apo state:
    a. Make copies of the configuration files "*.conf" from *$CPD/tutorials/adaptive_MC/apo/protMC* to *$PROJ/protMC/apo.*
    b. Edit the adaptive configuration file (*adapt.conf*) troubleshooting 3.
        i. Define the energy directory location of the apo state energy files (matrix.bb/pw).
        ii. Set the length of the MC trajectory to 3 × 10^8 steps.
        iii. Employ all active positions (455, 493, 494, 501) and all possible position pairs in the adaptive procedure, and update the calculated single and double bias terms every 1,000 steps. Use the parameters e_0 = 0.2 kcal/mol and E^0 = 100 kcal/mol for single position biases, and the values 0.1 kcal/mol and 40 kcal/mol for two-positions biases. (The values

of these parameters can be optimized for a particular system via short biasing simulations).

iv. Set the temperature via the thermal energy kT at 0.6 kcal/mol and the protein dielectric constant at 6.8.

c. Run protMC to conduct the adaptive-flattening simulation that computes and stores the biasing potentials in the output file *bias.dat*. The values are updated every 1,000 steps. The sampled sequences (*adapt.seq*) and their energies (*adapt.ener*) are also stored.

```
> $CPD/protMC/protMC.exe < adapt.conf > adapt.log
```

d. When the adaptive flattening is completed, manually copy the final set of values of the biasing potentials to the file *bias.in*, that is used in the next step.

11. Set up and run the biased "production design" simulation of the sequence – conformation space of the apo state:

a. Edit the MC configuration file (*MC.conf*) troubleshooting 3

i. Use a replica exchange Monte Carlo (REMC) exploration method, with 4 replicas at temperatures (0.6 kcal/mol, 0.9 kcal/mol, 1.3 kcal/mol, 1.8 kcal/mol).

ii. Include the file containing the biasing potentials.

iii. Reduce the conformational space, by restricting distant inactive residues (beyond 8 Å from the interface) to their native conformation.

iv. Set the trajectory length to 10^9 steps.

```
<Replica_Number> 4 <\Replica_Number>

<Temperature> 1.8 1.3 0.9 0.6 <\Temperature> #kT units

<Bias_Input_File> bias.in <\Bias_Input_File>

<Space_Constraints>

2417 LYS # all rotamers of lysine are available

2437 ASN{12} # only the native rotamer is available

<\Space_Constraints>

<Trajectory_Length> 1000000000 <\Trajectory_Length>
```

b. Run protMC to conduct the biased production simulation of the apo SARS-CoV-2 RBD. All sequence – structure combinations visited by each replica and their energies are stored in individual files (*proteus.seq*, *proteus.ener*), indexed 0, 1, 2, 3 according to the temperature of the replica (from the highest to the lowest).

```
> $CPD/protMC/protMC.exe < MC.conf > MC.log
```

c. Edit the postprocess configuration file *POST.conf* to include only the designed sequences sampled at the room temperature trajectory.

```
<Seq_Input_File> proteus.seq_3 <\Seq_Input_File>

<Fasta_File> proteus.rich_3 <\Fasta_File>
```

d. Run protMC to convert the stored sequences (*proteus.seq_3*) in a human-readable enriched form (*proteus.rich_3*)

```
> $CPD/protMC/protMC.exe < POST.conf > POST.log
```

   e. Run the python script *$CPD/bin/analyze_seq.py* to count the occurrence of the various sequences identified by the design. For each sequence the script reports, its occurrence and percentage probability, along with the minimum, maximum and average energy (over all rotameric conformations identified by the design).

```
> $CPD/bin/analyze_seq.py proteus_seq.3 proteus_rich.3
1000000000 $proj/matrix/active.list > proteus.dat
```

   f. A total of 81,643 out of 81,648 possible sequences were sampled in the simulation of the SARS-CoV-2 RBD apo state. The wild-type sequence L455-S493-Q494-N501 (“LQSN”) was selected 15,651 times. A graphical representation of the chemical type abundancies at each position can be constructed (Polydorides and Archontis, 2021).

12. Repeat the previous step to set up and run the biased production simulation of the sequence – conformation space of the complex state, using the same biasing potentials (*bias.in*). A total of 29,668 out of 81,648 possible sequences were sampled in the simulation of the complex. The wild-type sequence “LQSN” was selected 6,347 times. Run the following from the *$PROJ/complex*. troubleshooting 4

```
> $CPD/protMC/protMC.exe < MC.conf > MC.log
> $CPD/protMC/protMC.exe < POST.conf > POST.log
> $CPD/bin/analyze_seq.py proteus_seq.3 proteus_rich.3
1000000000 $proj/matrix/active.list > proteus.dat
```

13. Filter designed sequences according to their binding affinity.
   a. For sequences encountered very infrequently during the design, the computed sequence probabilities and the resulting binding free energies may be associated with significant errors. For this reason, we exclude sequences with smaller occurrence frequency than a minimum number, set here to 5000. In the simulations of the apo state and the complex, 60,660 and 6,350 sequences appeared, respectively, more than 5,000 times. Of those, a total of 5,596 sequences were common in both states. Extract these common sequences in two new files (i.e., proteus_5000.dat); store those files in the apo and complex directories.
   b. Run the python script $CPD/bin/affinity.py to calculate the binding free energy of each sequence with respect to the wild-type, using the biased probabilities in the apo and the complex state. The output file lists those sequences with improved binding affinity relative to the wild type, in descending order. A total of 4,704 out of the 5,596 had better affinity than the wild-type sequence.

```
> $CPD/bin/affinity.py $PROJ/CMP/protMC/proteus_5000.dat
$PROJ/APO/protMC/proteus_5000.dat bias.in -rf LQSN -p 455
493 494 501 > results.dat
```

14. Further analysis of potent sequences
   a. From the top (strongest-affinity) 50 to 100 sequences, choose a small number ~10 of sequences with chemical diversity.

b. Run the shell script *$CPD/bin/reconstruct.sh* from *$PROJ/complex/reconstruct* to create 3D structure models of selected sequences – conformations sampled by the selected replica (proteus_rich.3).

c. We reconstructed the most probable conformers of the highest-affinity sequence LEIV and two more interesting sequences, EEIV and LQWN. We tested further these sequences by MD simulations in explicit water, using the NAMD program (Phillips et al., 2020).

   i. Create a new subdirectory *$PROJ/comp/MD* and make copies of the coordinate (*rec.1.pdb*) and structure (*rec.1.psf*) files of reconstructed sequence.

   ii. Run the shell script *$CPD/bin/dihe_mult.sh* to correct the protein structure file from multiple dihedral entries, readable by NAMD.

⚠ CRITICAL: Employ a sufficient number of adaptive flattening steps to achieve reasonable convergence of biasing potentials. The simulation length depends on the size of the sequence space. For 4 active positions and 18 chemical types per position, there are 18^4 ($\sim 10^5$) possible sequences. Each sequence can be associated with a large number of rotamer conformations. Thus, a MC length of at least 10^9 steps would be expected). Note that perfect flattening of the energy landscape is not necessary. The MC output files *proteus.seq*, *proteus.ener* can be extremely large, depending on the trajectory length. We usually employ sequences from the room temperature replica, thus the output files storing solutions from higher-temperature replicas can be shrunk to zero using the Linux truncate command.

Note: As explained above, we set a minimum sequence occurrence threshold to exclude insufficiently sampled sequences during design. The threshold depends on the size of the sequence space. For a total of $\sim 10^5$ possible sequences, explored by 10^9 MC steps, we ignore sequences sampled less than 5,000 times.

EXPECTED OUTCOMES

The computation of biasing potentials does not consider correlations beyond pairs of active positions. Therefore, the biasing potentials do not render the sequences of the apo state strictly equiprobable. For this reason, the biased apo-state simulation with the converged biased potentials should have an appropriate length so that a large fraction of all possible sequences be sampled.

The bias potentials are approximately equal to the folding free energies of the apo state. In the biased simulation of the complex state, the bias potentials are subtracted from the folding free energies of the complex state, promoting sequence with low binding free energies (high binding affinities). Efficient sampling of the apo and complex energy landscapes should result in a large number of common sequences with high frequency occurrence and low binding free energies.

LIMITATIONS

The performance of the protocol depends on the accuracy of the energy function, the effectiveness of the bias potentials, and the quality of the sampling. Proteus employs a physics-based energy function to describe the protein-protein interaction energies and models solvent effects by implicit solvation models. A careful selection of solvation model and optimum parameters (protein dielectric constant, atomic surface-area coefficients, Lazaridis-Karplus (LK) parameters, etc.) should be made during the setup step. The exploration of the vast sequence – conformation space becomes tractable via Boltzmann weighted MC simulations, sampling billions of sequences/conformations in a short time. The quality of the sampling is improved via replica-exchange MC (REMC) simulations and optimum bias potentials. The combination of these techniques allows sampling of a large number of sequences with good binding affinities. The number of replicas, their temperature range, and the trajectory length should be decided with respect to the size of the combinatorial space. The biasing potentials are derived through an iterative procedure; a sufficient number of steps should be taken to ensure that convergence is reasonably achieved (i.e., the chemical types at each position are approximately equiprobable). Sidechain conformations with extremely high energies, arising from steric interactions, prevent convergence and should be excluded from the adaptive procedure.

During design, a part of the molecule remains fixed at the conformation specified by the initial structural model. Even though the missing backbone flexibility is partly taken into account via the protein dielectric constant, the results may depend on the chosen structural model troubleshooting 5.

TROUBLESHOOTING

Problem 1

Errors during the system build and setup steps occur if: (i) the atom/residue names are not compatible to the selected force field (Amber), (ii) the coordinates file (pdb) doesn't end with the "END" keyword and (iii) the application employs a molecule with unknown topology. Paths to files longer than 80 characters cause the termination of protX.

Potential solution

Edit the structure files properly and use additional files (patches) for the topology and parameters of any unknown molecules. Avoid large file names.

Problem 2

The fixed-backbone approximation occasionally results in the exclusion of bulky sidechains or specific rotamers at a given position. In that case, none of the available conformations fits and the interaction energies remain very large due to steric clashes, even after minimization. These unfavorable energies may cause a convergence problem during the adaptive flattening step.

Potential solution

Check the list of allowed rotamers for each position under *matrix/local/EnrFltr*. Employ suitable mutation-space constraints in the configuration files, to exclude amino acid types with solely unfavorable conformations. To allow excluded conformers, one could alleviate steric clashes during the initial preparation of the starting structural model or consider alternative starting models.

Problem 3

Syntax errors in the configuration files (*MC.conf, adapt.conf, POST.conf*) terminate the simulations and report the message "error in the optimization configuration". An incomplete interaction energy matrix will also cause the termination of the simulation indicating errors in the matrix files (*matrix.bb, matrix.pw*) with missing types or rotamers "Can not find type … ", "Can not find rotamer".

Potential solution

Keep the necessary syntax of the configuration files, and use compatible space constraints between the calculation steps. Check the interaction energy matrix files for missing chemical types or rotamers at a given position. Errors during the IEM calculation by protMC are reported at the *matrix/err* subdirectory for each diagonal and off diagonal term. Check for errors and recompute only the necessary pairs.

Problem 4

Using an unrestricted sequence / conformation space of the hACE2 in the complex state, limits sampling of the active positions and their neighboring inactive positions at the interface of the two proteins.

Potential solution

Keep the same sequence / conformation space constraints of SARS-CoV-2 in the apo and complex state and include additional space restrictions hACE2, within the MC configuration file (*MC.conf*).

Problem 5

The designed sequences may depend on approximations of the protocol, such as the fixed-backbone treatment and the discretization of conformational space. As an example, with the crystallographic structure as a model for the fixed backbone, the calculations predict the negatively charged residues glutamic and aspartic acid as the main solutions at position 493 (for a sequence logo see Figure 3B of (Polydorides and Archontis, 2021)). With a frame from the all-atom MD simulation of the complex as the fixed-backbone structural model, the polar analogues asparagine and glutamine are also inserted at that position.

Potential solution

Design calculations could be performed with a small number of different initial structural models (X-ray, some MD snapshots) and the results could be combined.

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Georgios Archontis (archonti@ucy.ac.cy).

Materials availability

Results of the application describing the protocol are reported in Polydorides and Archontis (2021).

Data and code availability

This study did not generate new code. The source code (Proteus3.0) is available at https://proteus.polytechnique.fr.

ACKNOWLEDGMENTS

S.P. thanks the Cyprus Ministry of Education, Culture, Sports, and Youth for a leave of absence.

AUTHOR CONTRIBUTIONS

Both authors contributed equally to this work.

DECLARATION OF INTERESTS

The authors declare no competing interests.

REFERENCES

Archontis, G., and Simonson, T. (2005). A residue-pairwise generalized born scheme suitable for protein design calculations. J. Phys. Chem. B *109*, 22667–22673.

Gaillard, T.,., and Simonson, T. (2017). Full protein sequence redesign with an MMGBSA energy function. J. Chem. Theor. Comput. *13*, 4932–4943.

Hawkins, G.D., Cramer, C.J., and Truhlar, D.G. (1995). Pairwise solute descreening of solute charges from a dielectric medium. Chem. Phys. Lett. *246*, 122–129.

Hornak, V., Abel, R., Okur, A., Strockbine, B., Roitberg, A., and Simerling, C. (2006). Comparison of multiple Amber force fields and development of improved protein backbone parameters. Proteins *65*, 712–725.

Jo, S., Kim, T., Iyer, V.G., and Im, W. (2008). CHARMM-GUI: a web-based graphical user interface for CHARMM. J. Comput. Chem. *29*, 1859–1865.

Lan, J., Ge, J., Yu, J., Shan, S., Fan, S., Zhang, Q., Shi, X., Wang, Q., Zhang, L., and Wang, X. (2020). Structure of the SARS-CoV-2 spike receptor-binding domain bound to the ACE2 receptor. Nature *581*, 215–220.

Michael, E., Polydorides, S., Simonson, T., and Archontis, G. (2017). Simple models for nonpolar solvation: parameterization and testing. J. Comput. Chem. *38*, 2509–2519.

Mignon, D., Druart, K., Michael, E., Opuu, V., Polydorides, S., Villa, F., Gaillard, T., Panel, N., Archontis, G., and Simonson, T. (2020). Physics-based computational protein design: an update. J. Phys. Chem. A *124*, 10637–10648.

Neria, E., Fischer, S., and Karplus, M. (1996). Simulation of activation free energies in molecular systems. J. Chem. Phys. *105*, 1902–1921.

Opuu, V., Nigro, G., Gaillard, T., Schmitt, E., Mechulam, Y., and Simonson, T. (2020). Adaptive landscape flattening allows the design of both enzyme: substrate binding and catalytic power. Plos Comput. Biol. *16*, e1007600.

Phillips, J.C., Hardy, D.J., Maia, J.D.C., Stone, J.E., Ribeiro, J.V., Bernardi, R.C., Buch, R., Fiorin, G., Henin, J., Jiang, W., et al. (2020). Scalable molecular dynamics on CPU and GPU architectures with NAMD. J. Chem. Phys. *153*, 044130.

Polydorides, S., and Archontis, G. (2021). Computational optimization of the SARS-CoV-2 receptor-binding-motif affinity for human ACE2. Biophys. J. *120*, 2859–2871.

Seeber, M., Felline, A., Raimondi, F., Muff, S., Friedman, R., Rao, F., Caflisch, A., and Fanelli, F. (2011). J. Comput. Chem. *32*, 1183–1194.

Villa, F., Mignon, D., Polydorides, S., and Simonson, T. (2017). Comparing pairwise-additive and many-body generalized Born models for acid/base calculations and protein design. J. Comput. Chem. *38*, 2396–2410.

Villa, F., Panel, N., Chen, X., and Simonson, T. (2018). Adaptive landscape flattening in amino acid sequence space for the computational design of protein:peptide binding. J. Chem. Phys. *149*, 072302.