listparindent=

Proteus and the design of ligand binding sites

Savvas Polydorides[1], Elena Michael[1], David Mignon[2], Karen Druart[2],
Thomas Simonson[2*] and Georgios Archontis[1*]

June 5, 2015

[1] Theoretical and Computational Biophysics Group, Department of Physics, University of Cyprus, 1678 Nicosia, Cyprus.
[2] Laboratoire de Biochimie (CNRS UMR7654), Department of Biology, Ecole Polytechnique, 91128 Palaiseau, France.

Abstract

1 Basic concepts

Computational protein design (CPD) is a collection of methods for to engineer proteins (and ligands) and to optimize molecular properties such as stability, binding affinity and binding specificity. Numerous successful CPD examples have been reported in recent years [**? ? ? ? ? ? ? ? ? ? ? ? ? ? ?**], and their impact will certainly increase with the continuous improvement in CPD tools and computational hardware.

Proteus (v. 2.0.1) [**? ?**] is a software package for computational protein and ligand design. It consists of i) a modified version of the XPLOR program [**?**], which performs the initial structural manipulations of the system under study, computes the energy matrix used in the design, and re-assesses the conformations and sequences suggested by the design, via energy and free-energy calculations and molecular dynamics simulations; ii) a library of scripts in the XPLOR command language that control the calculations; iii) the proteus program (v. 29.2), which conducts the actual search in the protein and ligand structure and sequence space; iv) a collection of Perl scripts for the analysis of the solutions provided by proteus. Shell scripts that automate the whole procedure are also available. For the sake of clarity, in the present chapter we describe a detailed design protocol, so that new users can follow it step by step.

Thermodynamic cycles The concepts of stability or specificity design, as implemented in proteus, are illustrated in the thermodynamic cycles of Figure 1. The cycle on the left compares the stabilities of two sequences A and B. The folding processes are depicted by the vertical legs; the horizontal legs display the (unphysical) transformations from sequence A into B, in the folded (N) and unfolded (U) state. The difference in the free energy changes of the horizontal (or vertical) legs yields the difference in stability of the two sequences:

$$\Delta\Delta G_f = [G(P_B^N) - G(P_A^N)] - [G(P_B^U) - G(P_A^U)] \tag{1}$$

Stability calculations seek to minimize the above free energy difference $\Delta\Delta G_f$.

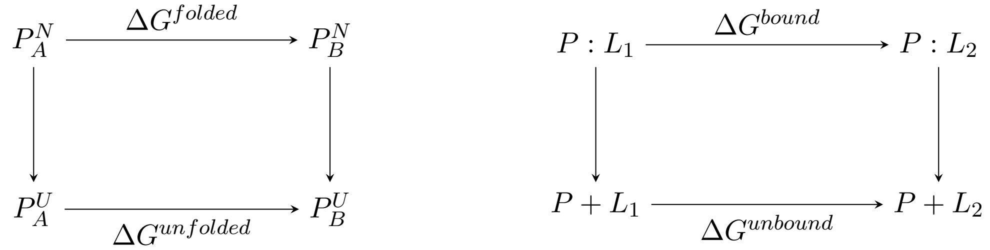


Figure 1: Thermodynamic cycles employed in CPD of stability (left) and ligand specificity (right).

Specificity calculations are illustrated by the thermodynamic cycle on the right of Figure 1. The vertical legs display the binding of two ligands L_1 and L_2 to a protein P; the horizontal legs display the (unphysical) chemical transformation between the two ligands, either in the protein complex (top leg) or in solution (bottom leg). If L_1 is a *reference* ligand and L_2 a modified analog, the calculations seek to minimize the relative binding free energy

$$\Delta\Delta G_b = [G(P:L_2) - G(P:L_1)] - [G(L_2) - G(L_1)] \tag{2}$$

The above expression assumes that the protein relaxes to the same state (P) upon dissociation of the two complexes (unlike some MM-PBSA or MM-GBSA methods [**?**]).

Energy model The free energies appearing in Eqs. (1)-(2) are computed via a physical energy function with the general form:

$$G = E_{bond} + E_{angle} + E_{dihedral} + E_{improper} + E_{vdw} + E_{coulomb} + E_{GB} + E_{SA} + E_{\rm corr} \tag{3}$$

The first six terms describe the internal and non-bonded contributions to the potential energy of the protein or ligand under consideration, and are borrowed from a molecular mechanics energy function. The parameterizations currently available within Proteus are the Charmm19 force field [**?**] and the AMBER ff99SB force field [**?**]. The next two terms capture solvent effects via a generalized Born (GB) approximation and an accessible surface area (SA) term. Simpler energy functions, that model solvent electrostatic screening via a homogeneous ("cdie") or distance-dependent ("rdie") dielectric constant are also available. The last term represents an optional "correction" energy, whose interpretation depends on the design criterion (see below).

Unfolded state The above free energies are functions of the atomic coordinates. This poses a difficulty in the case of unfolded states, for which structural models are not readily available. In stability calculations, we make the assumption that the sidechains do not interact with each other in the unfolded state, but only with nearby backbone and solvent [**?** **?** **?**]. We implement this idea by considering any sidechain X as a part of a tripeptde Ala-X-Ala. We compute the average free energy for a large number of backbone conformations of the tripeptide, using Eq. (3), and assign this value to chemical type X. An empirical correction can be added to this value, [last term of Eq. (3)], ensuring that the resulting amino acid compositions are reasonable during the design of whole protein sequences. The calculation of this term is explained in Ref. [**?**]. The total free energy of a given protein sequence in its unfolded state is the sum of the individual contributions of its constituent chemical types.

Ligand titration In the case of binding calculations, the contribution of the free protein cancels out in relative binding free energies, as explained above. The free energies of the unbound ligands can be averaged over single or multiple structures, obtained from experiments or simulations; alternatively, it may be assumed that the ligands (and possibly the protein) maintain the same conformations in solution and in the complexes. A correction [last term of Eq. (3)] can be added to the energy of the unbound ligand L, to express the dependence of binding free energies on the ligand concentrations:

$$E^{L}_{\text{corr}} = +k_B T \ln[L] \tag{4}$$

with k_B Boltzmann's constant, T the temperature, and $[L]$ the ligand concentration (a variable parameter). The ratio of concentrations of two complexes is given by the equation

$$\frac{[PL_2]}{[PL_1]} = \exp[-\beta(\Delta\Delta G_b - k_B T \ln([L_2]/[L_1])] \tag{5}$$

It is possible to vary the ligand concentration ratio $[L_2]/[L_1]$ progressively during ligand design, and monitor the ratio of predicted concentrations $[PL_1]$, $[PL_2]$; the binding free energy difference $\Delta\Delta G_b$ is then obtained as $k_B T \ln([L_2]/[L_1])$, for a concentration ratio $[L_2]/[L_1]$ that yields equal concentrations $[PL_1] = [PL_2]$.

Proton binding The thermodynamic cycle on the right of Figure 1 can also describe proton binding (or release) by titratable protein residues (e.g. Asp $\rightarrow$ AspH). This can be of use to determine sidechain protonation states and prepare a system for design or other simulations. Proton binding in the protein environment is described by the top, and in solution by the bottom horizontal leg. The solution state is a model compound, typically a single amino acid X with blocked terminal ends (ACE-X-NME). The free energy change upon protonation is:

$$\Delta\Delta G_p = [G(P-XH) - G(P-X)] - [G(XH) - G(X)] \tag{6}$$

and corresponds to the pK_a shift of a titrating site X in the protein, relative to the model compound. As in ligand optimization, in titration calculations we add a correction energy term to the free energy of the model compound in its protonated state, to account for the proton concentration $[\text{H}^+]$:

$$E^{X}_{corr} = 2.303 k_B T [pH - pK_a^{model}(X)] \tag{7}$$

with $pK_a^{model}(X)$ the experimental pK_a value for model compound X [**? ?**]. The fraction f of protonated states at different pH values is described by the following titration curve:

$$f = \frac{[XH]}{[X] + [XH]} = \frac{1}{1 + 10^{n(pH - pK_a(X))}} \tag{8}$$

To apply the above equation, titration calculations are conducted for different pH values. The pK_a of residue X is the pH for which the protonated and unprotonated states are equiprobable. The Hill coefficient n represents the maximum slope of the curve, which occurs iat the titration mid-point.

Multi-objective optimization As described above, the Proteus suite is a CPD multi-tool, which is applicable to typical sequence/structure optimization calculations, but also to more refined pK_a and relative binding affinity calculations. Its physical scoring function, with the addition of appropriate correction energy terms, can be easily adjusted to describe different situations. Eqs. (1) and (2) can be decomposed into protein-ligand intramolecular and intermolecular energy contributions, which can be enhanced or diminished during energy minimization via appropriate weight factors (positive, negative or zero); and combined to produce more sophisticated multi-objective optimization functions, as follows:

$$\widetilde{G} = w_1 \cdot G(P) + w_2 \cdot G(P:L) + w_3 \cdot G(L) + w_4 \cdot G_{dc}(P) + w_5 \cdot G_{dc}(L) \tag{9}$$

The subscript "dc" denotes duplicate copies of the protein and ligand groups, which maintain the amino acid sequence, but sample different conformations during optimization. Energy threshold values can also be included in Eq. (9) to refine the optimization outcome.

Energy matrix The design begins by separating the protein (and ligand, if present) into groups (residues), which can contain backbone and sidechain moieties. Part of the system, typically the backbone and selected sidechains, is classified as "frozen"; i.e., it retains its conformation and chemical composition during the calculation. Other parts can change both chemical identity and conformation ("active"), or only conformation ("inactive"). Sidechain conformations are taken from a rotamer library [**?**]. Multiple backbone conformations can also be specified [Eq. (9)]. We then precompute and store in a matrix the interaction energies for all intra- and intermolecular residue pairs, taking into account all chemical types and conformations compatible with the classification of each residue. This calculation is done by XPLOR and a library of command scripts, using the energy function of Eq. (3). The GB and SA terms of the energy function are not rigorously pairwise-decomposable; i.e., even though they can be expressed as contributions from particular residue pairs, each contribution depends on the geometry of the entire molecule. To solve this problem, we employ a "Native Environment Approximation" (NEA) for the GB term, and a "sum over atom-pairs" approximation for the SA term; more details are supplied below and in Ref. [**?**].

The entries of the resulting interaction matrix correspond to distinct rotamer orientations of the active and inactive parts, and to a given conformation of the "frozen" part. Often, it is

desirable to take into account multiple conformations of the frozen part (e.g., several backbone conformations from an MD trajectory). Separate interaction matrices can be constructed for each of these conformations, and employed in the design.

Sequence/structure exploration The interaction energy matrices are read by the C program proteus, which performs the optimization in structure and sequence space. Three exploration methods are available in proteus; a heuristic protocol, first introduced by Wernisch et al. [**?**], a mean-field approach [**?** **?**] and a Monte Carlo (MC) method [**?** **?**]. The Monte Carlo method can use a single "walker", exploring a single trajectory. Alternatively, it can use multiple walkers, which have distinct temperatures, explore distinct trajectories, and occasionally exchange their temperatures. The multi-walker variant corresponds to a "replica exchange" Monte Carlo simulation, which we refer to as REMC.

All the exploration methods output multiple "solutions", sampled along the trajectory or heuristic exploration. Each solution or timestep is described by a list of chemical types and rotamers for all the active and inactive positions. Subsequently, the corresponding conformations can be reconstructed, and subjected to energy minimization and/or MD simulations with the same force field used in the design. Average binding free energies can be obtained from the resulting trajectories, and/or post-processed using an MM-GBSA or MM-PBSA approximation, as a further test of the design.

Flowcharts The above calculations are summarized in the flowcharts of Figure 2. The top flowchart portrays a structure/sequence optimization of a complex, which starts from an initial conformation taken from an MD trajectory. A related example, described in the Methods section, involves the redesign of the cyclic 13-residue peptide compstatin, which regulates the function of the complement system protein C3. Binding of this molecule and related analogs has been the subject of numerous experimental and computational studies in recent years [**?** **?** **?** **?** **?**].

The lower flowchart describes the preparation of an X-ray structure for MD simulations. A related example in Methods describes the chemical and structural optimization of a complex between the MHC class II protein HLA-DQ8 and the vinculin epitope.

2 Materials

To carry out a complete protein design calculation with Proteus, the user needs the Proteus 2.0.1 CPD platform. The appropriate files can be downloaded from the BIOC group site: *http://biology.polytechnique.fr/biocomputing/proteus.html*. In what follows, we refer to spe-

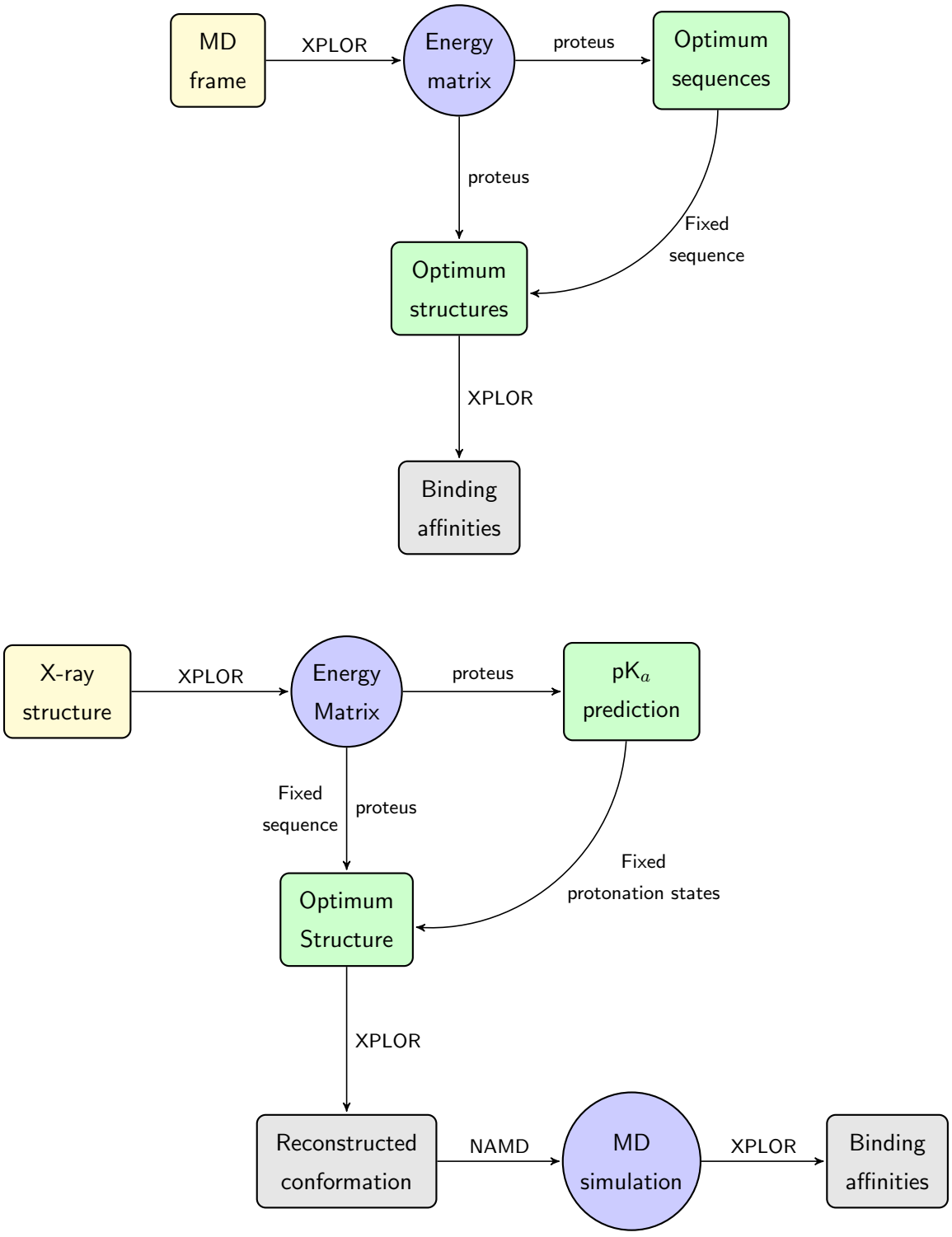


Figure 2: Calculation flow chart diagrams for the test cases: (top) ligand redesign, and (bottom) preparation of a structure for MD simulations.

cific files from this distribution. Furthermore, the user needs an initial structural model for the molecule (or complex) under optimization.

3 Methods

3.1 Structure preparation

1. Split the PDB file into separate files for each protein segment (e.g. multiple chains), the ligand, and the crystallographic waters. Rename atoms and residues to match the Amber or Charmm force field. Renumber residues of each segment starting from 1000 for chain A, 2000 for chain B, etc., to avoid duplicate residue numbers; name the various segments "PROA", "PROB", "PROC" or "LIGA" and "XWAT" (see Note 1).

2. Use the XPLOR script *build.inp* to generate a protein structure file (*system.psf*) which describes the topology of the protein:ligand system and a coordinate file (*system.pdb*) in XPLOR pdb format (see Note 2).

3.2 System setup

3. The XPLOR stream file *parameters.str* contains important information about energy calculations. Edit the file to select between the Amber "ff99SB" [**?**] and Charmm "toph19" [**?**] force fields. These two force fields are consistent, respectively, with the GB/HCT [**?**] and GB/ACE [**?**] implicit solvent models. Add a surface area term to the energy function to account for the non polar contribution to the solvation energy. Include X-ray sidechain conformations ("native rotamers") to the rotamer library, and choose the number of minimization steps before the computation of pairwise interaction energies. Set the protein dielectric constant and define parameters employed by the solvation model and the corresponding non-bonded energy terms.

4. Modify the XPLOR stream file *sele.str* to define the sequence and conformation space. Select the modifiable residues (active), the flexible sidechains (inactive), the ligand (active or inactive) and the fixed part (backbone plus any glycines, prolines, cysteines in disulfide bonds, and crystallographic waters/ions).

5. The file *mutation_space.dat* lists the amino acid types available for each active position. The mutation space includes up to 26 amino acid types, including all natural amino acids (except glycine and proline), three histidine tautomers (protonated on N_δ, N_ϵ, or

both), and the uncommon protonation states of titratable residues Lys, Asp, Glu, Tyr, Cys.

6. The system setup is completed via the XPLOR script *setup.inp*, which prepares the system for residue pairwise energy calculations. The structure file *setup.psf* defines each active residue, including its crystallographic backbone and a set of sidechains corresponding to all considered mutations (defined in *mutation_space.dat*). Entries of these amino acid sidechains at each modifiable position are included in the coordinate file *setup.pdb*, with initialized coordinates ($x = y = z = 9999.0$). The B-factor column of the coordinate file labels the corresponding residue as active ($b = 2.00$), inactive ($b = 1.00$), or fixed ($b = 0.00$). The Q-factor column labels buried ($q = 0.00$) and exposed ($q = 1.00$) residues, via their exposure fraction. At this point the backbone atomic GB solvation radii are computed and stored in the file *bsolv.pdb*.

7. The Perl script *make_position_list.pl* reads the file *setup.pdb*, and lists in *position_list.dat* the active, inactive and ligand positions, including the number of all possible pairwise interactions to be computed at each position.

8. The Shell script *make_mutation_space.sh* creates individual files for each active, inactive and ligand position, listing the compatible amino acid types at each position. These files are stored locally and read later by the XPLOR scripts during the residue pairwise interaction calculations.

3.3 Interaction Energy Matrix

9. First we compute the diagonal terms of the interaction energy matrix, using *matrixI.inp*. The calculation is usually executed sequentially over all non frozen positions; it is also possible to run the separate positions simultaneously on multiple cores. For each position I, we loop over its compatible amino acid types (depending on whether it is active, inactive, frozen, or part of the ligand). For each amino acid type we loop over rotamer states taken from a rotamer library [**?**]. We also include the native orientation as a separate rotamer. At this stage, we also compute and store GB radii for all residues in the Native Environment Approximation (NEA). In a standard GB formulation, the GB energy function is not pairwise-decomposable, as the GB radius of each atom depends on the position and chemical type of all other atoms in the molecule. To render the GB function pairwise-decomposable, we assume during the GB radii calculation that each residue is surrounded by the native sequence and conformation. Namely, for each rotamer we compute the solvation GB radii in the presence of residue I, the whole

backbone (fixed part) and all remaining portions of the molecule, further than 3.0 Å away from sidechain *I*, considered in their native sequence and structure. The GB 3.0 Å cutoff distance excludes native sidechain atoms that might overlap with sidechain *I* in its new rotamer; this cutoff can be adjusted to a different value by *parameters.str*. Importantly, to alleviate possible clashes of a sidechain in a particular rotamer conformation with the backbone, we perform 15 steps of Powell minimization (see Note 3), keeping everything else (except sidechain *I*) fixed. If a resulting solvation radius is too large (due to overlap of the residue under consideration with the rest of the molecule), it is reset to a maximum value (999.0 Å). After the minimization, we update the solvation radii of the sidechain, restore those of the backbone (from *bsolv.pdb*) and compute the interaction energy of sidechain *I* with itself and the backbone.

The energy function describes bond, angle, dihedral, improper, van der Waals, Coulomb, GB and SASA energies. The sidechain coordinates after minimization and the corresponding solvation radii are stored in a local pdb file (*matrix/local/Rota/ 1025_ARG_5.pdb*) to be used in step 11. The results are printed in local files (*matrix/dat/matrix_I_1025.dat*) and can be displayed either in standard or enriched format. The basic identification information for each position is printed with standard format: residue number (1025), amino acid type (ARG), one letter code (R), rotamer index number (2) followed by four energy values: the unfolded state (or unbound ligand) energy (estimated by Eq. (3)), the bonded terms plus vdW, the electrostatic including GB, and the surface area term. A further decomposition of individual energy terms is displayed when the "enriched format" is specified in *parameters.str*.

10. Use the Shell script *make_rotamer_space.sh* to examine the rotamer van der Waals energies and exclude those exceeding a locally defined threshold value. Excluding "bad" rotamers for each amino acid type at each position reduces the conformational space.

11. The energy matrix calculation continues with the off-diagonal terms, by computing the interaction between sidechains *I* and *J*. Only the lower triangle of the matrix ($I > J$) is needed. The fastest approach for this part of the calculation evaluates single residue pairs *I*-*J* simultaneously, on multiple cores. It is also possible to calculate all the residue pair interactions sequentially. For each residue pair, we loop over the sidechain type/rotamer space of residue *I*; we retrieve the coordinates and atomic solvation radii of the current sidechain from the rotamer PDB file (*matrix/local/Rota/1025_ARG_5.pdb*), created in step 9. For each rotamer we loop over all residues $J < I$ and apply a first distance filter. Residues that are too far from *I* (e.g., C_β–C_β distance > 30 Å) are omitted. For each residue *J* within the first distance filter, we loop over the sidechain

type/rotamer space of residue J and read the coordinates and solvation radii from the corresponding rotamer PDB files. For both residues I and J we employ only the "good" rotamers, determined in the previous step. With the current sidechains in place, we apply a second distance filter, where interactions between sidechains are ignored if the minimum distance between the two sidechains exceeds 12 Å. The interaction energies of sidechain pairs that pass the second distance filter are computed. Recall that the final coordinates of two sidechains are produced via the independent minimization of each sidechain in the presence of the fixed backbone. Consequently, it is possible that the two sidechains overlap, for some rotamer combinations. If the minimum sidechain-sidechain distance is smaller than a cutoff (3 Å), we perform 50 steps of Powell minimization (see Note 3) to improve the sidechain geometry and alleviate bad contacts. During this minimization, everything except the two sidechains is kept fixed, and the two sidechains interact with each other and the backbone. The results are stored in local files (*matrix/dat/matrix_IJ_1025_1022.dat*). The standard display format consists of a headline indicating the residue numbers and names of a given pair (1025 ARG 1022 VAL), followed by a list of entries for each computed rotamer pair, for the given pair of amino acid types. Each entry reports the two rotamer numbers, the vdW interaction term, the sum of electrostatic and GB terms, and the surface area term. Similarly to step 9, an "enriched format" option is possible, which prints a more detailed output file.

12. Finally, run the shell script *concat_matrix.sh* to join all energy elements in a global matrix file *matrix.dat*, to be read by the proteus exploration program.

3.4 Protein design

3.4.1 Sequence optimization

The sequence exploration is done by the proteus program, controlled by setting various options in an input script, *proteus.conf*.

13. Choose the same protein dielectric constant value in *proteus.conf* as the one used in the energy matrix calculations [defined in *parameters.str*]. To use a different value, first use the Perl script *modify_matrix.pl* to modify the original matrix (see Note 4).

14. During the energy matrix construction (Subsection 3.3), a large set of active and inactive positions can be defined. During sequence exploration, we may want to limit ourselves to a smaller set. For this, in *proteus.conf*, the sequence/conformational space of selected protein and/or ligand residues can be restricted to particular types and/or rotamers.

For example, in the redesign of the compstatin peptide, in the energy matrix calculation, we chose all 15 ligand positions to be active and all protein sidechains to be inactive; subsequently, in proteus, we optimized the sequence of just a two-residue extension. The default option corresponds to a full scale exploration of all possible amino acid types and rotamers for each active and inactive position (see Note 5).

15. Choose among the mean field, heuristic, and Monte Carlo sequence/structure exploration methods, and assign the relevant parameters. For example, if the MC method is employed, we might use a high initial temperature (given in kT units) to overcome local energy barriers, and run several long simulations [millions of steps; (see Notes 5)]. By default, the simulation starts from a random sequence/structure combination and uses the Metropolis criterion to evaluate the successive moves in sequence and rotamer space. The exploration is performed using single and/or double moves, enabling the sampling of coupled sidechains. The frequency of each type of move during the simulation is also controlled by the occurrence probability of each mutation type; a small sequence/structure move ratio (1:10 or 2:10) allows the system to relax in the presence of the new amino acid type (see Notes 6).

16. All exploration parameters mentioned in steps 14 and 15 are set up via a simple, user-editable configuration file (*proteus.conf*), which is read as the standard input by the proteus executable.

17. Running proteus in post-processing mode converts the resulting solutions into a more readable (fasta-like) form. The output file *proteus.rich* reports each solution by the sequence of: (a) amino acid types (b) residue numbers (c) rotamer numbers. The Perl script *analyze_proteus_sequences.pl* sorts the solutions (combinations of sequences and rotamers) by their frequency of occurrence and calculates the minimum, maximum and average folding free energies.

3.4.2 Structure optimization

After large-scale sequence exploration, it can be desirable to do more extensive rotamer exploration for selected sequences.

18. Repeat the above steps for a chosen subset of designed sequences. Keep each protein and ligand sequence invariant, and explore their conformational space through rotamer optimization. Compute the statistical average of the folding free energy over all sampled conformations, to improve the energy estimate for the chosen sequences.

19. Use the Perl script *rot_distrib_proteus.pl* to compute the rotamer distribution of all residues from the pseudo-trajectory obtained during optimization, to characterize the flexibility of each sidechain.

20. Cluster the protein and ligand conformations based on selected sidechains, and reconstruct the minimum energy conformation of each cluster to get a set of "good" conformations.

3.5 pK$_a$ calculations

In some cases, we wish to determine sidechain protonation states through pK$_a$ calculations. For each titratable sidechain, the energy will include a pH-dependent term, $E^X_{\rm corr}$, where X is the sidechain type.

21. First, compute the correction energy term $E^X_{\rm corr}$ at pH = 7 (see Equation (7)), by evaluating the energy G^{model}_X of the model compound in solution with Eq. (3), and replace the values representing the unfolded state energy from the diagonal matrix elements with $-G^{model}_X$.

22. Modify the proteus configuration file to restrain the mutation space of each active-titratable residue to its two or three ionization states (ASP/ASH, GLU/GLH, CYS/CYM, HID/HIE/HIP, TYR/TYD, LYS/LYN); restrict the other positions to their native type (or make them inactive during the energy matrix calculation).

23. Run a proteus MC simulation, to identify optimum combinations of sequences (protonation states) and structures at the specified pH. Start with 1 million equilibration steps at high temperature (k_BT = 1 kcal/mol), extract the final state and continue with 10 million production steps at room temperature; use a relatively small sequence-to-structure move ratio (1:10), to allow the system to relax after protonation moves.

24. At the end of the MC simulation, compute the probabilities of each protonated state at each active, titratable position (see Note 7).

25. Run a full pH scan by increasing progressively the pH from 0 to 15 and repeating steps 21 to 24.

26. Fit the fractional occupancy of the protonated state to the modified Hill equation (Eq. (8) for each titratable sidechain using the Perl script xxx.pl; extract the pK$_a$ value with the corresponding Hill coefficient at the mid-point of the sigmoidal curve.

Table 1 (adapted from Ref. [**?**]) shows pK_a calculations for 9 proteins and 130 titratable groups with sufficient sidechain type diversity (35 Asp, 34 Glu, 13 Tyr, 28 Lys and 20 His). Overall, the agreement with experiment is good, with an rms deviation of just 1.1 pH units, for reasonable protein dielectric constants 4 and 8. For sidechains with large pK_a shifts, ≥ 2, the rms error with our method is 1.8, compared to 2.6 with the Null model (and 1.1 with the specialized PROPKA program).

Table 1: Comparing large and small pK_a shifts

Experimental range	Number of sidechains	[a]Null model	[a]MC $\epsilon_p = 4$	[a]MC $\epsilon_p = 8$	[a]PROPKA3
$\|\Delta pK_a\| < 1$	85	0.5	0.9	1.0	0.6
$2 > \|\Delta pK_a\| \geq 1$	45	1.7	1.3	1.2	1.0
$\|\Delta pK_a\| \geq 2$	11	2.6	1.8	1.8	1.1
All[a]	130	1.1	1.1	1.1	0.8

[a]Rms deviations between computed and experimental pK_a's.

An application example involves the chemical and structural optimization of a complex between the MHC class II protein HLA-DQ8 and the vinculin epitope [**?**]. Since the structure of the specific complex was not known, we started from the X-ray structure of the HLA-DQ8 complex with an insulin peptide. MHC class II proteins bind various peptides in the endosome where the pH ranges from 4.5 to 6.0; therefore, in the initial setup we determined the ionization state of titrating groups by pK_a calculations with Proteus. The binding site (residues within 8 Å of the peptide) contains 23 titrating sidechains (3 Lys, 3 His, 2 Asp, 6 Glu and 9 Tyr residues, out of 98 residues). Arginines were excluded, since they titrate well outside the pH range of interest ($4.0 \leq pH \leq 7.0$). We focused on a group of residues near the first anchor position (P1) of the binding groove, where αGlu31, βGlu86, αHis24 and αArg52 form a strong interaction network. Between αGlu31, αHis24 and P1 there is also an important crystallographic water. The two gluatamic acids are 4.1 Å apart ($C_\delta - C_\delta$) and their titrating behaviour is coupled. The net charge of this group of residues was not verified by the X-ray crystallography [**?**], and was a matter of discussion in further studies of HLA-DQ8 and MHC class II proteins [**?** **?**]. We performed pK_a calculations with two dielectric constants $\epsilon_p =$ 4 and 8, both in the absence and the presence of the vinculin peptide; and compared our results with the empirical Propka model. For extracellular pH values around 7 proteus calculations with $\epsilon_p = 4$ and Propka predict a neutral histidine and a protonated αGlu31. The other glutamic βGlu86 is overestimated by proteus, but becomes better

at $\epsilon_p = 8$. Similar pK_a values are obtained for the complex and the free protein. Figure 3 shows a superposition of the reconstructed optimum conformation (vinculin) and the template X-ray structure (insulin). Setting the appropriate ionization state for αGlu31 promotes a successful sidechain placement of all key role residues that take part in binding (Figure 3). Structure preparation as performed by preliminary pK_a calculations and sidechain placement is an important byproduct of Proteus.

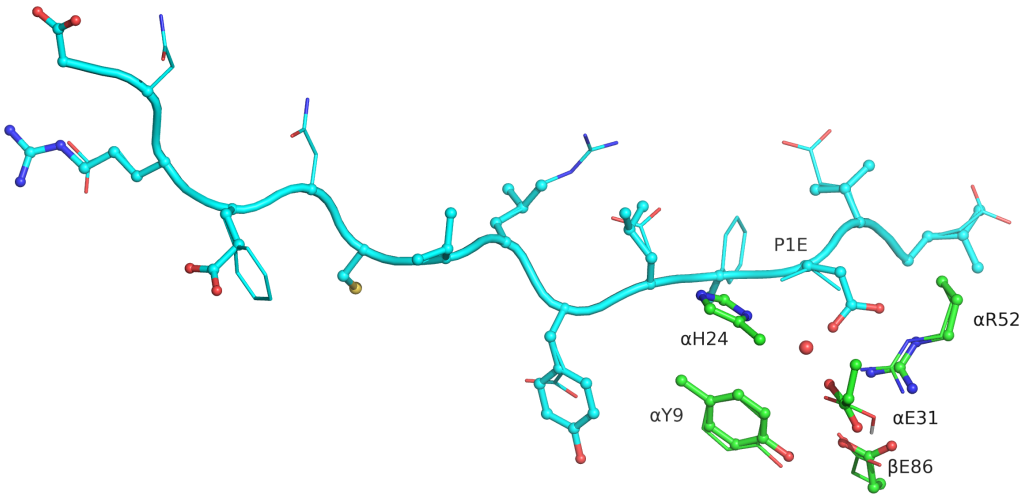


Figure 3: Superposition of the starting X-ray structure of the insulin complex (ball-and-stick view) and the optimized conformation of the vinculin complex (thick lines).

3.6 Specificity calculations by ligand titration

In many applications, we want to discover sequences that favor one ligand over another, and design for specificity. One approach is to make two or more ligands compete for a single binding site. By gradually increasing the concentration of one ligand, we gradually displace the other(s), and can extract the relative binding free energy from the titration curve.

27. Set all or part of the ligand to be active, with two or more types; say, X_{nat} (natural ligand) and X_{mut} (alternative, or "mutant" ligand). The protein and any remaining ligand positions are inactive. To speed up the calculation, constrain the rotamer space of distant residues (those situated further than 8 Å from the active position) to their native conformation (see Note 5).

28. Assign a correction term to the mutant ligand [Eq. (4)], to reflect a low initial, relative concentration. This term has two parts. The first part is $k_B T \ln([L_{X_{\text{mut}}}]/[L_{X_{\text{nat}}}])$. The

second part is the energy difference between the two unbound ligands, computed with Eq. (3). The first contribution can be set to -5 kcal/mol; this corresponds to the case where the native ligand is represented in the mixture at a much higher concentration than the mutant type, favoring the native ligand binding.

29. Run a short stage of equilibration (500,000 steps) at high temperature, followed by a long production stage (10 million steps) at room temperature starting from the final state of equilibration.

30. Count the number of steps with the mutant ligand present and deduce the population fraction with a bound mutant ligand.

31. Repeat steps 27 to 30 while gradually increasing the relative concentration term of the two competing ligands from -5 to +5 kcal/mol. As we gradually increase the concentration of the unbound ligand $L_{X_{\mathrm{mut}}}$, it slowly displaces $L_{X_{\mathrm{nat}}}$ from the binding site and takes its place.

32. Fit the data to the appropriate titration curve [adapted from Eq. (5)] and obtain the binding free energy difference from the mid-point of the sigmoidal curve, where the populations of the mutant and native ligand are equal.

A ligand titration example A related example involves the redesign of the cyclic 13-residue peptide compstatin, which regulates the function of the complement system protein C3. We and our collaborators have studied extensively by computational and experimental methods the binding of compstatin and its analogs to C3 [**? ? ? ?**]. In recent work [**? ?**], we explored the addition of a two-residue extension [XY] to the N-terminal end of the compstatin double mutant Ac-Val4Trp/His9Ala ([XY]W4A9). MD simulations had suggested that this extension may increase the number of contact residues with the protein. Using a snapshot from MD simulations of the C3 complex with [RS]W4A9, we searched for extension sequences that optimized ligand binding. To determine the amino acid type preference of the two-residue extension of compstatin, we computed the binding free energy difference [Eq. (2)] of each amino acid type X with respect to Ala at each position of the extension. Binding affinities (relative to Ala) for various amino acid substitutions at positions -2 and -1 are summarized in Table 2. Columns 2 and 6 contain the results from design calculations at extension positions -2 and -1, respectively, in which all amino acid types are allowed to compete simultaneously; the resulting affinities are computed from the individual amino acid frequencies in the resulting solutions. Columns 3 and 7 contain the results of calculations, in which

only one amino acid at a time competes with Ala; the corresponding relative affinities are computed from Eq. (5). The results of the two methods agree closely. Experimentally, positions -2 and -1 can tolerate various amino acid types, without large differences in the corresponding binding free energies[**?**]. The design favors a positively charged Arg residue at position -2. MD simulations of the [RS]W4A9 complex with C3 suggest that an Arg residue at position -2 forms a strong electrostatic interaction with proximal residue Glu372 (see Figure 3); this interaction is captured by the proteus design. Position -1 is predicted no to have a strong amino acid propensity; it somewhat disfavours 14 out of 18 types, especially bulky hydrophobic sidechains. This can be explained by the fact that sidechains placed at position -1 are oriented towards the solvent.

33. It can be useful to reassess the designed sequences by additional calculations. In the compstatin redesign study, we performed rotamer optimization on the designed sequences and clustered the resulting conformations (based on the rotamer states of all sidechains within 8 Å of the extension). For each sequence, we reconstructed representative conformations from the ten most populated clusters, and subjected them to 100 steps of energy minimization with the Powell conjugate gradient method. During minimization, we kept the backbone fixed, to facilitate comparison with the raw design results. We then computed the binding free energy of each conformation at the end of minimization with the MM-GBSA approximation, as the difference between the free energy of the complex and the isolated ligand and protein. The results, averaged over the ten conformations, are included in columns 4 and 8 of Table 2; the values are expressed relative to alanine. Some bulky amino acid types (Trp, Lys, Met, His, Tyr, Leu, Val, Ile) become slightly preferred at position -2 after minimization, due to enhanced van der Waals interactions with Val375 (see Figure 3). At position -1, Arg still represents the optimum sidechain after reconstruction and minimization. These predictions may still change after MD simulations of the same complexes.

4 Notes

1. The ligand can be a polypeptide segment (chain C), like the insulinB 14-mer bound to HLA-DQ8, which we treat the exact way as the protein, or a non-peptidic molecule like heme in haemoglobin. In that case we need to define the topology of the new molecule and specify the necessary parameters. The new segment must be named "LIGA".

2. The file *build.inp* must be modified to match the segment names defined by the user. The file reads the amino acid sequence of each chain according to its segment name,

the number of water molecules, and adds disulphide bonds and terminal group patches, to generate the corresponding molecular structure. The coordinates of any missing hydrogens are assigned, and the structures are grouped in the system.psf and system.pdb files.

3. The minimization steps performed prior to the energy calculations in steps 9 and 11 balance to some extent the suboptimal orientations available to the sidechains through the discrete rotamer space. The number of minimization steps can be adjusted for specific cases. For several systems, extending the minimization to more than 50 steps was shown to increase computational cost without significant improvement in results.

4. The protein dielectric constant is an empirical parameter. Its value depends on the type of calculation and the solvation model used. For CPD applications with a GBSA implicit solvent model, we found that low dielectric values 4–8 give reasonable results. pK_a calculations on a large data set of titrating sites showed improved accuracy for $\epsilon_p = 8$ [**?**]. For whole protein designs, a higher value such as $\epsilon_p = 16$ may give better results [**? ? ?**].

5. To obtain adequate sampling, we restrict the sequence/conformation space depending on the application. For the compstatin redesign, we focussed on the area surrounding the peptide extension. The two extension residues are allowed to sample all amino acid types and rotamers without any restrictions, while every sidechain within 10 Å from any atom of the extension changes only conformation. The remaining residues are kept fixed, together with the backbone, at the X-ray conformation.With these "local" space restrictions, the exploration converged within 10 million steps. The quality of the sampling can be assessed by repeating the calculation with different seed values and different random number generators, or by performing both backward and forward pH/ligand concentration scans (see Eqs. (4) and (7)). The convergence of the method can be tested with additional simulations of increasing length.

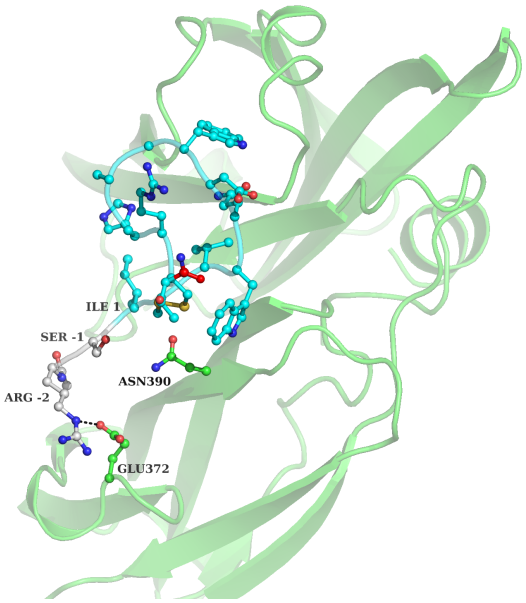
6. With MC exploration, the relative frequency of mutation and rotamer moves (both single and double) can be adjusted by the user in the proteus.conf configuration file to match the needs of a given calculation [**?**]. Conformational changes are usually less drastic than amino acid type changes (i.e. Ala → Arg); therefore, it is generally preferred to allow more rotamer than type moves, to allow the system to relax after a mutation.

7. To calculate correctly the fractional occupancies from the Monte Carlo simulation, both accepted and rejected moves should be accounted for, since a move rejection signifies a

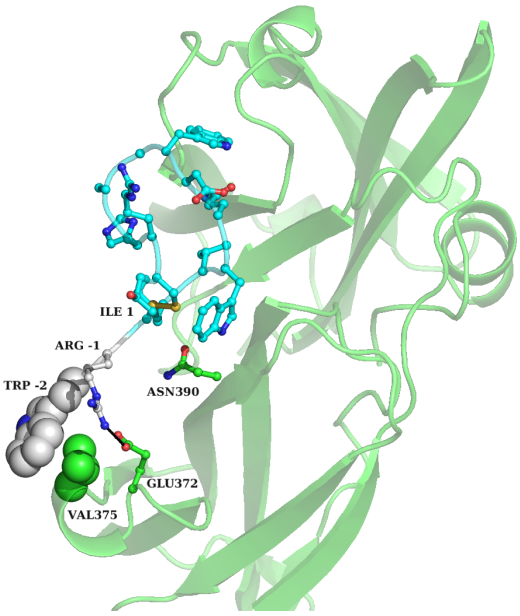
preference for the previously occupied state.

Acknowledgements

References



[[RS]W4A9]



[[WR]W4A9]

Figure 4: 3D structure of the cyclic 13-residue peptide compstatin analogue W4A9 (cyan) and a two-residue extension added to the N-terminal end (white) in complex with the protein C3 (green). (a) Starting structure used by Proteus (b) minimized structure of a predicted mutant.

Table 2: Sequence optimization, affinity and specificity calculations on the compstatin extension.

Extension residues							
position -2				position -1			
aa type	$\Delta\Delta G^{[a]}$	$\Delta\Delta G^{[b]}$	$\Delta\Delta G^{[c]}$	aa type	$\Delta\Delta G^{[a]}$	$\Delta\Delta G^{[b]}$	$\Delta\Delta G^{[c]}$
	(kcal/mol)				(kcal/mol)		
R	-0.9	-2.0	-1.4	R	-0.4	0.0	-1.4
Y	-0.1	0.0	-1.7	S	0.0	0.0	-0.4
A	-	-	-	A	-	-	-
M	0.0	0.0	-1.9	N	0.0	0.0	-0.4
C	0.0	0.0	-0.6	C	0.1	0.0	-0.1
K	0.1	0.0	-1.1	T	0.3	0.5	0.2
N	0.1	0.0	-0.8	Q	0.4	0.8	-0.1
V	0.1	0.0	-0.8	M	0.5	0.9	-0.5
Q	0.1	0.0	-1.2	V	0.5	1.9	-0.3
S	0.2	0.0	0.0	K	0.5	1.3	0.0
I	0.2	0.3	-1.4	Y	0.6	1.0	-0.6
F	0.2	0.4	-0.3	W	0.7	1.5	-0.3
W	0.4	0.5	-3.4	H(N_ϵ)	0.8	1.5	0.0
T	0.4	0.5	0.0	H(N_δ)	0.8	1.5	-0.2
H(N_δ)	0.4	0.5	-1.8	E	0.8	1.3	-0.1
H(N_ϵ)	0.4	0.5	-0.7	D	0.8	1.3	-0.2
L	0.5	1.0	-1.3	F	2.0	0.9	-0.8
E	0.6	1.1	-0.8	I	1.1	2.0	-0.5
D	0.9	1.5	0.0	L	1.1	2.0	-0.3

All binding affinities computed relative to Alanine (A) .

[*a*] Estimated from the frequency of the solutions with the corresponding amino acid in target position -2 or -1.

[*b*] Estimated from the titration curves.

[*c*] Estimated after reconstruction and minimization of the resulting solutions for a 100 steps with a fixed backbone. The results are averaged over the 10 most populated rotamer conformations, taking into account all sidechains within 8 Å from the extension.