Chapter 6

Proteus and the Design of Ligand Binding Sites

Savvas Polydorides, Eleni Michael, David Mignon, Karen Druart, Georgios Archontis, and Thomas Simonson

Abstract

This chapter describes the organization and use of Proteus, a multitool computational suite for the optimization of protein and ligand conformations and sequences, and the calculation of pK_α shifts and relative binding affinities. The software offers the use of several molecular mechanics force fields and solvent models, including two generalized Born variants, and a large range of scoring functions, which can combine protein stability, ligand affinity, and ligand specificity terms, for positive and negative design. We present in detail the steps for structure preparation, system setup, construction of the interaction energy matrix, protein sequence and structure optimizations, pK_α calculations, and ligand titration calculations. We discuss illustrative examples, including the chemical/structural optimization of a complex between the MHC class II protein HLA-DQ8 and the vinculin epitope, and the chemical optimization of the compstatin analog Ac-Val4Trp/His9Ala, which regulates the function of protein C3 of the complement system.

Key words Protein design, Ligand design, Monte Carlo, Implicit solvent, Generalized Born model

1 Introduction

Computational protein design (CPD) is a set of methods to engineer proteins (and ligands) and optimize molecular properties such as stability, binding affinity, and binding specificity. Many successful CPD examples have been reported in recent years [1–15], and their impact will certainly increase with the continuous improvement in CPD tools and computational hardware.

We have developed the Proteus (v. 2.1) software package for computational protein and ligand design [16–18]. It consists of (1) a modified version of the XPLOR program [19], which performs the initial setup of the system under study, computes an energy matrix used in the design, and re-assesses the conformations and sequences suggested by the design; (2) a library of scripts in the XPLOR command language that control the calculations; (3) the proteus program (v. 30.4), which conducts the actual

Barry L. Stoddard (ed.), *Computational Design of Ligand Binding Proteins*, Methods in Molecular Biology, vol. 1414,
DOI 10.1007/978-1-4939-3569-7_6,

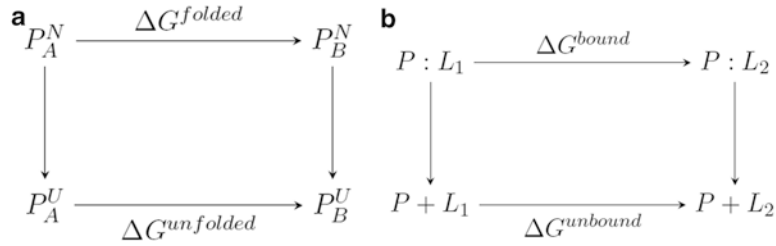

Fig. 1 Thermodynamic cycles employed in CPD of stability (**a**) and ligand specificity (**b**)

search in the protein and ligand's structure and sequence space; (4) a set of Perl scripts to help analyze the solutions provided by proteus. Shell scripts that automate the whole procedure are also available. For the sake of clarity, in this chapter we describe a detailed design protocol, so that new users can follow it step by step.

1.1 Thermodynamic Cycles

The concepts of stability or specificity design, as implemented in Proteus, are illustrated in the thermodynamic cycles of Fig. 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*) states. The difference between the free energy changes for the horizontal (or vertical) legs yields the difference in stability between the two sequences:

$$\Delta\Delta G_{\mathrm{f}} = \left[G\left(P_{\mathrm{B}}^{\mathrm{N}}\right) - G\left(P_{\mathrm{A}}^{\mathrm{N}}\right)\right] - \left[G\left(P_{\mathrm{B}}^{\mathrm{U}}\right) - G\left(P_{\mathrm{A}}^{\mathrm{U}}\right)\right] \qquad (1)$$

Stability calculations seek to minimize the above free energy difference $\Delta\Delta G_{\mathrm{f}}$.

Specificity calculations are illustrated by the thermodynamic cycle on the right of Fig. 1. The vertical legs represent the binding of two ligands L_1 and L_2 to a protein *P*; the horizontal legs represent 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

$$\Delta\Delta G_{\mathrm{b}} = \left[G\left(P : L_2\right) - G\left(P : L_1\right)\right] - \left[G\left(L_2\right) - G\left(L_1\right)\right] \qquad (2)$$

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 [20, 21]).

1.2 Energy Model

The free energies appearing in Eqs. 1–2 are computed via a physical energy function with the general form:

$$G = E_{\mathrm{bond}} + E_{\mathrm{angle}} + E_{\mathrm{dihe.}} + E_{\mathrm{impr.}} + E_{\mathrm{vdW}} + E_{\mathrm{coul.}} + E_{\mathrm{GB}} + E_{\mathrm{SA}} + E_{\mathrm{corr}} \qquad (3)$$

The first six terms describe the internal and nonbonded contributions to the potential energy of the protein or ligand under study, and are borrowed from a molecular mechanics energy function. The parameterizations currently available in Proteus are the Charmm19 force field [22] and the Amber ff99SB force field [23]. 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*).

1.3 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 [24–26]. We implement this idea by considering any sidechain X as a part of a tripeptide 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 (*see* last term of Eq. 3), chosen so that the resulting amino acid compositions are reasonable during the design of whole protein sequences. The calculation of this term can be done ahead of time and is explained in Ref. 18. The total free energy of a given protein sequence in its unfolded state is the sum of the individual contributions of its constituent residue types.

1.4 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 (*see* 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:

$$E_{\mathrm{corr}}^{\mathrm{L}} = +k_{\mathrm{B}}T\ln\left[L\right] \tag{4}$$

with k_{B} the Boltzmann's constant, T the temperature, and $[L]$ the ligand concentration (set by the user). The ratio of concentrations of two complexes obeys the equation

$$\frac{\left[PL_2\right]}{\left[PL_1\right]} = \exp\left[-\beta\left(\Delta\Delta G_{\mathrm{b}} - k_{\mathrm{B}}T\ln\left(L_2 / L_1\right)\right)\right] \tag{5}$$

One can 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 the concentration ratio $([L_2]/[L_1])$ that yields equal concentrations $[PL_1]=[PL_2]$.

1.5 Proton Binding

The thermodynamic cycle on the right of Fig. 1 can also describe proton binding (or release) by titratable protein residues (e.g., Asp → 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 upper horizontal leg, and in solution by the lower leg. The solution state is a model compound—typically a single amino acid X with blocking terminal groups (ACE-X-NME). The free energy change upon protonation in the protein, relative to the model compound in solution, is:

$$\Delta\Delta G_p = \left[G(P-XH)-G(P-X)\right]-\left[G(XH)-G(X)\right] \quad (6)$$

and corresponds to the pK_α difference between the sidechain in the protein and the model compound. In titration calculations, as in ligand optimization, we add a correction term to the free energy of the model compound in its protonated state to account for the proton concentration $\left[\ ^{+}\right]$:

$$E_{corr}^{X} = 2.303 k_B T\left(pH - pK_a^{model}\right) \quad (7)$$

where pK_α^{model} is the experimental pK_α value for model compound [27, 28]. The fraction f of protonated states at different pH values can usually be described by the following titration curve:

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

To apply the above equation, titration calculations are conducted for different pH values. The pK_α 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 at the titration mid-point.

1.6 Multi-Objective Optimization

As described above, Proteus is a multitool CPD suite, which is applicable to typical sequence/structure optimization calculations, but also to more refined pK_α and relative binding affinity calculations. Its physical scoring function, with the addition of appropriate correction 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 weighting factors (positive, negative, or zero); and

combined to produce more sophisticated, multi-objective energy, or cost functions, as follows:

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

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

1.7 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 their chemical identity and conformation ("active"), or only their conformation ("inactive"). Sidechain conformations are taken from a rotamer library [29]. Multiple backbone conformations can also be specified (*see* Eq. 9). We then pre-compute 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 (active or inactive). 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-additive; 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. 30.

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.

1.8 Sequence/ Structure Exploration

The interaction energy matrices are read by the C program proteus, which performs the exploration (or "optimization") in structure and sequence space. Three exploration methods are available in proteus; a heuristic protocol, first introduced by Wernisch et al. [26], a mean-field approach [31, 32], and a Monte Carlo (MC) method [33, 34]. 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

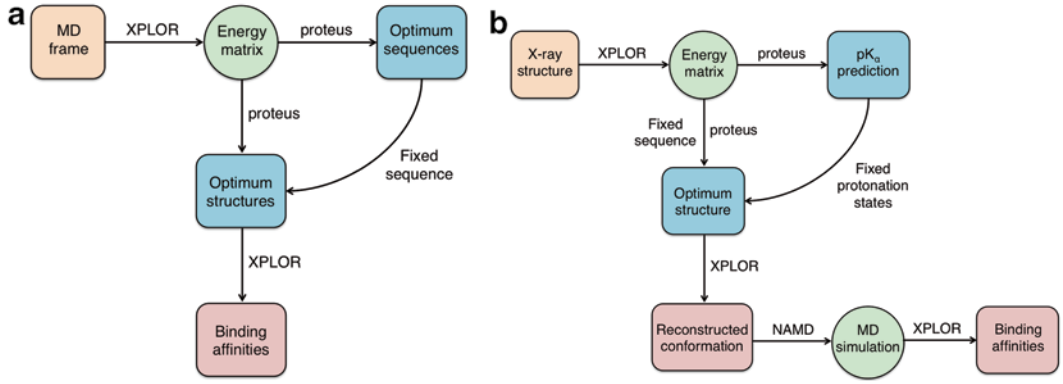

Fig. 2 Calculation flowchart diagrams for the test cases: (**a**) ligand redesign, and (**b**) preparation of a structure for MD simulations

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 MC trajectory or the heuristic exploration. Each solution or time-step 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 a GBSA or PBSA approximation, as a further test of the design.

1.9 Flowcharts

The above calculations are summarized in the flowcharts of Fig. 2. The left 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 protein C3 of the complement system. Binding of this molecule and related analogs has been the subject of numerous experimental and computational studies in recent years [35–39]. The right 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: Software and Data Files

To carry out a complete protein design calculation with Proteus, the user needs the Proteus 2.1 CPD package. The appropriate files can be downloaded from *http://biology.polytechnique.fr/biocomputing/*

proteus.html. In what follows, we refer to specific files from this distribution. Furthermore, the user needs an initial structural model for the molecule (or complex) under study.

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 ensure unique 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

1. The XPLOR stream file *parameters.str* contains important information about the energy calculation setup. Edit the file to select between the Amber "ff99SB" [23] and Charmm "toph19" [22] force fields. These two force fields are consistent, respectively, with the GB/HCT [40] and GB/ACE [41] implicit solvent models. Add a surface area term to the energy function to account for the nonpolar contribution to the solvation energy. Include X-ray sidechain conformations ("native rotamers") in 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 nonbonded energy terms.
2. 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).
3. 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 minor protonation states of titratable residues Lys, Asp, Glu, Tyr, Cys.
4. The system setup is done via two XPLOR scripts. The first one, *setup.inp*, 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 arbitrary 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 frozen $(b = 0.00)$. The Q-factor column labels buried $(q = 0.50)$ and exposed $(q = 1.00)$ residues, with $q = 0.00$ for hydrogens. At this point the GB solvation radii of the backbone atoms are computed and stored in the file *bsolv.pdb*.

5. 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.
6. 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.
7. The second XPLOR script for system setup is *setupI.inp*. For each position *I*, we loop over its allowed 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 [29]. We also include the native orientation as a separate rotamer. At this stage, we compute and store GB solvation radii for all residues, assuming the Native Environment Approximation (NEA). In a standard GB formulation, the GB energy function is not pairwise-additive, since the solvation radius of each atom depends on the position and chemical type of all other atoms in the molecule. To render the GB function pairwise-additive, we assume during the solvation radii calculation that each residue is surrounded by the native sequence and conformation. Thus, for each rotamer, we compute the GB solvation 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 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 in *parameters.str*. Importantly, to alleviate possible clashes of a sidechain in a particular rotamer with the backbone, we do $N_{\min} = 15$ steps of Powell energy minimization (*see* **Note 3**), keeping everything else (everything but sidechain *I*) fixed. If a resulting solvation radius is too large (e.g., due to overlap of the residue with the rest of the molecule), it is reset to a maximum value (999.0 Å). After the minimization, sidechain coordinates and solvation

radii are stored in a local PDB file (*matrix/local/Rota/1025.pdb*; 1025 is the residue number *I*) to be used in **step 3** from Subheading 3.3.

3.3 Interaction Energy Matrix

1. First, we compute the diagonal terms of the interaction energy matrix using the file *matrixI.inp*. This rather fast calculation is usually run sequentially over all nonfrozen positions; it is also possible to run the separate positions in parallel on multiple cores. For each position *I*, we reread the solvation radii and sidechain coordinates (*matrix/local/Rota/1025.pdb*). We loop over the allowed amino acid types (depending on whether position *I* is active, inactive, frozen, or part of the ligand) and the corresponding rotamer states. For each rotamer, we compute the energy due to interactions that sidechain *I* makes with itself and with the backbone. The energy function includes bond, angle, dihedral, improper, van der Waals, Coulomb, GB, and SASA energies. 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 information for each position is printed with the standard format: residue number (1025), amino acid type (ARG), one letter code (R), rotamer index number (5) followed by four energy values: the unfolded state (or unbound ligand) energy (*estimated by* Eq. 3), the bonded terms plus vdW, the electrostatic term, including GB, and the surface area term. A further decomposition of individual energy terms is displayed when the "enriched format" is requested in *parameters.str*.
2. 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.
3. The energy matrix calculation continues with the off-diagonal terms, using *matrixIJ.inp*, which computes 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.pdb*), created in **step 7** from Subheading 3.2. 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., $C_\alpha - C_\alpha$ 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, say. 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 $N_{\min}\ (15-50)$ 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 line 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 1**, an "enriched format" option is possible, which prints a more detailed output.

4. Finally, run the shell script *concat_matrix.sh* to join all the 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*.

1. One may want to use a protein dielectric constant that is different from 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 accordingly (*see* **Note 4**).
2. During the energy matrix construction (*see* Subheading 3.3, **steps 1** and **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 set 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 other peptide positions were not allowed to mutate. 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**).

3. 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 $k_B T$ units) to overcome local energy barriers, and run several long simulations [millions of steps; (*see* **Notes 5**–**7**)]. 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, improving 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 its structure slightly in the presence of the new amino acid type (*see* **Note 6**).
4. All exploration parameters mentioned in **steps 2** and **3** are set up via a simple, user-editable configuration file (*proteus.conf*), which is read as the standard input by the proteus executable.
5. After the exploration step, proteus is run again in post-processing mode, to convert the resulting solutions into a more readable (fasta-like) format. The output file *proteus.rich* reports each solution by the sequence of: (a) amino acid types, (b) residue numbers, and (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.

1. Repeat the above steps for a chosen subset of designed sequences. Keep each protein and ligand sequence invariant, and explore its 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.
2. 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.

3. 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_α Calculations

In some applications, we wish to determine sidechain protonation states through pK_α calculations. For each titratable sidechain, the energy will include a pH-dependent term, $E_{corr}{}^{X}$, where X is the sidechain type.

1. First, compute the correction energy term $E_{corr}{}^{X}$ at $pH = 7$ (*see* Eq. 7), by evaluating the energy GX^{model} of the model compound in solution with Eq. 3, and replace the values representing the unfolded state energy from the diagonal matrix elements with $-G_X^{model}$.
2. 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).
3. Run a proteus MC simulation, to identify optimum combinations of sequences (protonation states) and structures at the specified pH. Start with one million equilibration steps at high temperature $\left(k_B T = 1\,\text{kcal/mol}\right)$, extract the final state and continue with ten 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.
4. At the end of the MC simulation, compute the probabilities of each protonated state at each active, titratable position (*see* **Note 8**).
5. Run a full pH scan by increasing progressively the pH from 0 to 15 and repeating **steps 1–4**.
6. Fit the fractional occupancy of the protonated state to the modified Hill equation (*see* Eq. 8) for each titratable sidechain using the Perl script *evalpka.pl*; extract the pK_α value with the corresponding Hill coefficient at the mid-point of the sigmoidal curve.

Table 1 (*adapted from* Ref. 42) shows pK_α calculations for nine 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 of four and eight. For sidechains with large pK_α 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_α shifts

Experimental range	Number of sidechains	[a]Null model	[a]MC		[a]PROPKA3
			$\varepsilon_p = 4$	$\varepsilon_p = 8$	
$\lvert\Delta pK_\alpha\rvert < 1$	85	0.5	0.9	1.0	0.6
$1 \le \lvert\Delta pK_\alpha\rvert < 2$	34	1.7	1.3	1.2	1.0
$2 \le \lvert\Delta pK_\alpha\rvert$	11	2.6	1.8	1.8	1.1
All	130	1.1	1.1	1.1	0.8

[a]Rms deviations between computed and experimental pK_α shifts

An application example involves the chemical and structural optimization of a complex between the MHC class II protein HLA-DQ8 and the vinculin epitope [43]. 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_α 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 ($4.0 \le \mathrm{pH} \le 7.0$). 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 ($C_\gamma - C_\gamma$) and their titrating behavior is coupled. The net charge of this group of residues could not be verified by X-ray crystallography [44], and was a matter of discussion in subsequent studies of HLA-DQ8 and MHC class II proteins [45, 46]. We performed pK_α calculations with two dielectric constants, $\varepsilon_p = 4$ 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 $\varepsilon_p = 4$ and Propka predict a neutral histidine and a protonated αGlu31. The pK_α of the other glutamic acid, βGlu86, is overestimated by Proteus, but becomes better at $\varepsilon_p = 8$. Similar pK_α 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 residues that take part in binding (*see* Fig. 3). Structure preparation as performed by preliminary pK_α calculations and sidechain placement is an important byproduct of Proteus.

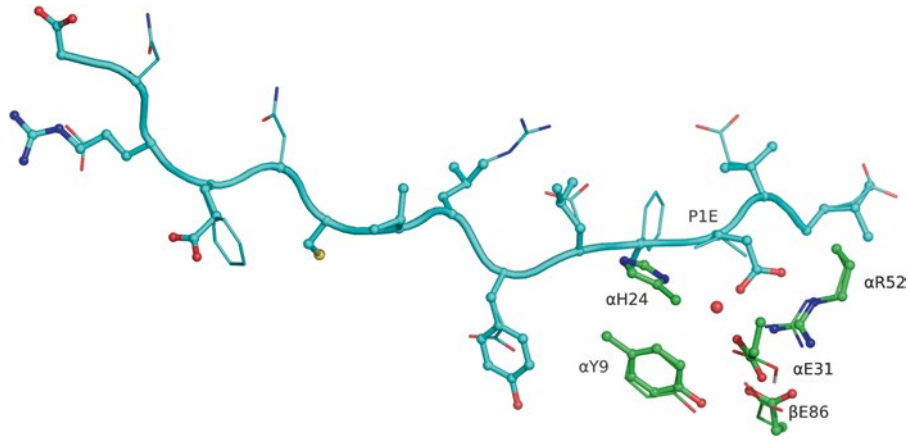

Fig. 3 Superposition of the starting X-ray structure of the insulin complex (*ball*-and-*stick view*) and the optimized conformation of the vinculin complex (*thick lines*)

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. This can be done with the protein sequence fixed or variable. Here, for simplicity, we describe an application where the protein sequence is fixed, and we focus on the relative binding strength of two ligands.

1. 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 (further than 8 Å, say, from the active position) to their native conformation (*see* **Note 5**).
2. Assign a correction term to the mutant ligand (*see* Eq. 4), to reflect a low initial, relative concentration. This term has two parts. The first part is $k_{\text{B}}T\ln\left(L_{X_{\text{mut}}} / L_{X_{\text{nat}}}\right)$. The second part is the energy difference between the two unbound ligands, computed with Eq. 3. The first contribution can be set to $-5\,\text{kcal}/\text{mol}$; 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.
3. Run a short equilibration stage (500,000 steps) at high temperature, followed by a long production stage (ten million steps) at room temperature starting from the final state of equilibration.

4. Count the number of steps with the mutant ligand present and deduce the population fraction with a bound mutant ligand.
5. Repeat **steps 1–4** while gradually increasing the relative concentration term of the mutant ligand from −5 to +5 kcal / mol . As we increase the concentration, $L_{X_{\text{mut}}}$ gradually replaces $L_{X_{\text{nat}}}$ in the binding site.
6. Fit the data to the appropriate titration curve (*adapted from* Eq. 5) and obtain the binding free energy difference from the mid-point, where the populations of the bound mutant and native ligands are equal.

A ligand titration example: This example involves the redesign of the cyclic 13-residue peptide compstatin, which regulates the function of protein C3 of the complement system. We and our collaborators have studied extensively the binding of compstatin and its analogs to C3 by computational and experimental methods [36, 37, 47, 48]. In recent work [38, 39], 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 (*see* 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 [38]. 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* Fig. 4a); this interaction is captured by the Proteus design. Position −1 is predicted to not have a strong propensity for one particular sidechain type; it somewhat disfavors 14 out of

18 types, especially bulky hydrophobic sidechains. This can be explained by the fact that sidechains at position –1 are oriented toward the solvent.

7. 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 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* Fig. 4b). 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 in the same way as the protein, or a nonpeptidic molecule like the heme in hemoglobin. In that case, we need to define the topology of the new molecule and specify the necessary parameters and possibly rotamers. 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 and adds disulfide bonds and terminal group patches, to generate the corresponding molecular structure. The coordinates of any missing hydrogens are assigned, and the structures are saved in the *system.psf* and *system.pdb* files.
3. The energy minimization steps done in **steps 1** and **3** from Subheading 3.3 balance to some extent the suboptimal orientations available to the sidechains due to the discrete rotamer space. The number of minimization steps can be adjusted for

Table 2
Sequence optimization, affinity, and specificity calculations in the compstatin:C3 complex, targeting the N-terminal extension of compstatin

Extension residues							
Position –2				**Position –1**			
	$\Delta\Delta G^a$	$\Delta\Delta G^b$	$\Delta\Delta G^c$		$\Delta\Delta G^a$	$\Delta\Delta G^b$	$\Delta\Delta G^c$
aa type	**(kcal/mol)**			**aa type**	**(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 ten most populated rotamer conformations, taking into account all sidechains within 8 Å from the extension

specific cases. For several systems, extending the minimization to more than 50 steps was shown to increase computational cost without a significant improvement in the results.

4. The protein dielectric constant is an empirical parameter. Its value depends on the type of calculation and the solvation

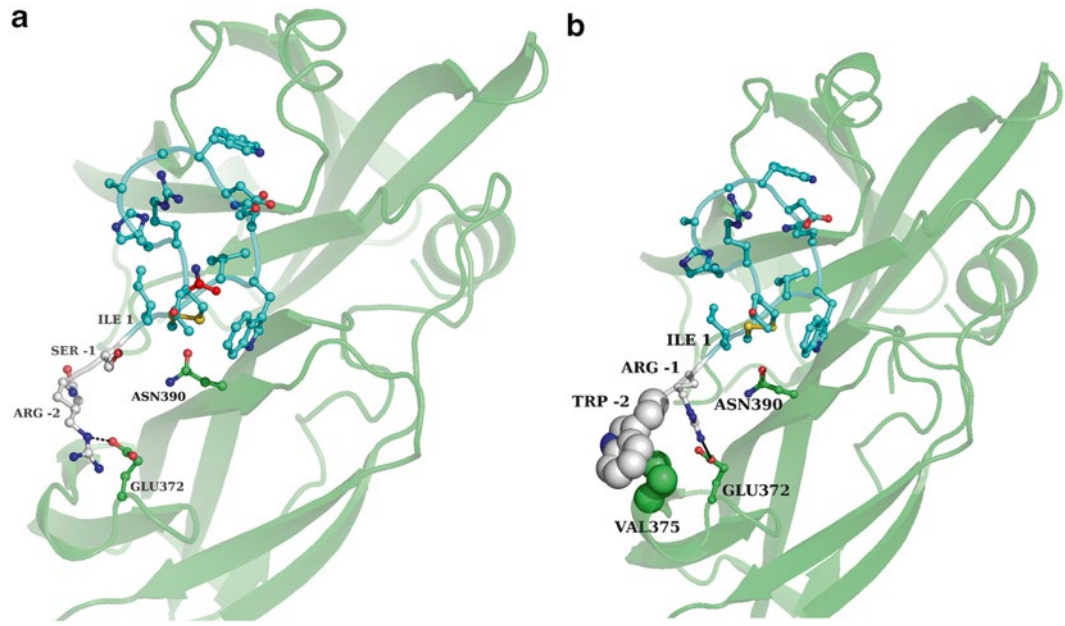

Fig. 4 3D structure of the cyclic 13-residue peptide compstatin analog W4A9 (*cyan*) and a two-residue extension to the N-terminal end (*white*) in complex with the protein C3 (*green*). (a) Starting structure used by Proteus, (**b**) minimized structure of a predicted mutant

model used. For CPD applications with a GBSA implicit solvent model, we found that low dielectric values of 4–8 give reasonable results. pK_α calculations on a large data set of titrating sites showed good accuracy for $\varepsilon_p = 8$ [42]. For whole protein designs, a higher value such as $\varepsilon_p = 16$ may give better results [49, 50].

5. To obtain adequate sampling, we restrict the sequence/conformation space depending on the application. For the compstatin redesign, we focused 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 other sidechain within 8 Å from any atom of the extension changes only its conformation. The remaining residues are held fixed, together with the backbone, in the X-ray conformation. With these "local" space restrictions, the exploration converged within ten million steps. The quality of the sampling can be assessed by repeating the calculation with different random number seed values, or by performing both backward and forward pH or ligand concentration scans (*see* Eqs. 4 and 7). The convergence of the method can also 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 [51]. 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. With MC exploration, it is possible to run multiple simulations in parallel, with different temperatures, such that the simulations periodically exchange their temperatures. This method is known as Replica Exchange, or REMC. It is activated in the *proteus.conf* file by indicating the number of simulations (or "walkers"), their temperatures, and the interval between temperature swaps. Each walker then generates its own output files. On a multi-core machine, the simulations will run in parallel if the OpenMP library is present.
8. 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

GA, SP, and EM acknowledge financial support through a grant offered by the University of Cyprus.

References

1. Kortemme T, Baker D (2004) Computational design of protein–protein interactions. Curr Opin Chem Biol 8(1):91–97
2. Floudas C, Fung H, McAllister SR, Monnigmann M, Rajgaria R (2006) Advances in protein structure prediction and de novo protein design: a review. Chem Eng Sci 61:966–988
3. Boas EF, Harbury PB (2007) Potential energy functions for protein design. Curr Opin Struct Biol 17(2):199–204
4. Lippow SM, Tidor B (2007) Progress in computational protein design. Curr Opin Biotechnol 18:305–311
5. Das R, Baker D (2008) Macromolecular modeling with Rosetta. Biochemistry 77(1): 363–382
6. Karanicolas J, Kuhlman B (2009) Computational design of affinity and specificity at protein-protein interfaces. Curr Opin Struct Biol 13:26–34
7. Damborsky J, Brezovsky J (2009) Computational tools for designing and engineering biocatalysts. Curr Opin Struct Biol 19: 458–463
8. Mandell DJ, Kortemme T (2009) Backbone flexibility in computational protein design. Curr Opin Biotechnol 20:420–428
9. Suarez M, Jaramillo A (2009) Challenges in the computational design of proteins. J R Soc Interface 6:477–491
10. Saven JG (2010) Computational protein design: advances in the design and redesign of biomolecular nanostructures. Curr Opin Colloid Interface Sci 15:13–17
11. Pantazes RJ, Greenwood MJ, Maranas CD (2011) Recent advances in computational protein design. Curr Opin Struct Biol 21: 467–472
12. Der BS, Kuhlman B (2013) Strategies to control the binding mode of de novo designed protein interactions. Curr Opin Struct Biol 23(4):639–646
13. Moal IH, Moretti R, Baker D, Fernandez-Recio J (2013) Scoring functions for protein-protein interactions. Curr Opin Struct Biol 23(6)
14. Zanghellini A (2014) de novo computational enzyme design. Curr Opin Biotechnol 29: 132–138

15. Khoury GA, Smadbeck J, Kieslich CA, Floudas CA (2014) Protein folding and de novo protein design for biotechnological applications. Trends Biotechnol 32(2):9099–9109
16. Schmidt am Busch M, Lopes A, Mignon D, Simonson T (2008) Computational protein design: software implementation, parameter optimization, and performance of a simple model. J Comput Chem 29:1092–1102
17. Polydorides S, Amara N, Simonson T, Archontis G (2011) Computational protein design with a generalized Born solvent model: application to asparaginyl-tRNA synthetase. Proteins 79:3448–3468
18. Simonson T, Gaillard T, Mignon D, Schmidt am Busch M, Lopes A, Amara N, Polydorides S, Sedano A, Druart K, Archontis G (2013) Computational protein design: the Proteus software and selected applications. J Comput Chem 34:2472–2484
19. Brünger AT (1992) X-plor version 3.1, A System for X-ray crystallography and NMR. Yale University Press, New Haven
20. Srinivasan J, Cheatham T, Cieplak P, Kollman P, Case DA (1998) Continuum solvent studies of the stability of DNA, RNA, and phosphoramidate-DNA helices. J Am Chem Soc 120:9401–9409
21. Simonson T (2013) Protein-ligand recognition: simple models for electrostatic effects. Curr Pharm Des 19:4241–4256
22. Brooks B, Bruccoleri R, Olafson B, States D, Swaminathan S, Karplus M (1983) Charmm: a program for macromolecular energy, minimization, and molecular dynamics calculations. J Comput Chem 4:187–217
23. Cornell W, Cieplak P, Bayly C, Gould I, Merz K, Ferguson D, Spellmeyer D, Fox T, Caldwell J, Kollman P (1995) A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. J Am Chem Soc 117:5179–5197
24. Pokala N, Handel TM (2005) Energy functions for protein design: adjustment with protein–protein complex affinities, models for the unfolded state, and negative design of solubility and specificity. J Mol Biol 347:203–227
25. Dahiyat BI, Mayo SL (1997) De novo protein design: fully automated sequence selection. Science 278:82–87
26. Wernisch L, Hery S, Wodak S (2000) Automatic protein design with all atom force fields by exact and heuristic optimization. J Mol Biol 301:713–736
27. Pace CN, Grimsley GR, Scholtz JM (2009) Protein ionizable groups: pKa values and their contribution to protein stability and solubility. J Biol Chem 284:13285–13289
28. Aleksandrov A, Thompson D, Simonson T (2010) Alchemical free energy simulations for biological complexes: powerful but temperamental. J Mol Recognit 23:117–127
29. Tuffery P, Etchebest C, Hazout S, Lavery R (1991) A new approach to the rapid determination of protein side chain conformations. J Biomol Struct Dyn 8(6)
30. Gaillard T, Simonson T (2014) Pairwise decomposition of an mmgbsa energy function for computational protein design. J Comput Chem 35:1371–1387
31. Koehl P, Delarue M (1994) Application of a self-consistent mean field theory to predict protein sidechain conformations and estimate their conformational entropy. J Mol Biol 239:249–275
32. Zou BJ, Saven JG (2005) Statistical theory for protein ensembles with designed energy landscapes. J Chem Phys 123:154908
33. Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E (1953) Equation of state calculations by fast computing machines. J Chem Phys 21:1087–1092
34. Frenkel D, Smit B (1996) Understanding molecular simulation. Academic, New York
35. Qu H, Ricklin D, Lambris JD (2009) Recent developments in low molecular weight complement inhibitors. Mol Immunol 47(2): 185–195
36. Tamamis P, Pierou P, Mytidou C, Floudas CA, Morikis D, Archontis G (2011) Design of a modified mouse protein with ligand binding properties of its human analog by molecular dynamics simulations: the case of c3 inhibition by compstatin. Proteins 79(11):3166–3179
37. Tamamis P, Lopez de Victoria A, Gorham RD, Bellows ML, Pierou P, Floudas CA, Morikis D, Archontis G (2012) Molecular dynamics in drug design: new generations of compstatin analogs. Chem Biol Drug Des 79(5):703–718
38. Gorham RD, Forest DL, Tamamis P, Lopez de Victoria A, Kraszni M, Kieslich CA, Banna CD, Bellows ML, Larive CK, Floudas CA, Archontis G, Johnson LV, Morikis D (2013) Novel compstatin family peptides inhibit complement activation by drusen-like deposits in human retinal pigmented epithelial cell cultures. Exp Eye Res 116:9096–9108
39. Gorham RD, Forest DL, Khoury GA, Smadbeck J, Beecher CN, Healy ED, Tamamis P, Archontis G, Larive CK, Floudas CA, Radeke MJ, Johnson LV, Morikis D (2015) New compstatin peptides containing n-terminal extensions and non-natural amino acids exhibit potent complement inhibition and improved solubility characteristics. J Med Chem 58(2): 814–826

40. Hawkins GD, Cramer C, Truhlar D (1997) Parameterized model for aqueous free energies of solvation using geometry-dependent atomic surface tensions with implicit electrostatics. J Phys Chem B 101:7147–7157
41. Schaefer M, Karplus M (1996) A comprehensive analytical treatment of continuum electrostatics. J Phys Chem 100:1578–1599
42. Polydorides S, Simonson T (2013) Monte Carlo simulations of proteins at constant pH with generalized born solvent. J Phys Chem B 34:2742–2756
43. van Heemst J, Jansen DTSL, Polydorides S, Moustakas AK, Bax M, Feitsma AL, Bontrop-Elferink DG, Baarse M, van der Woude D, Wolbink G-J, Rispens T, Koning F, de Vries RRP, Papadopoulos GK, Archontis G, Huizinga TW, Toes RE (2015) Crossreactivity to vinculin and microbes provides a molecular basis for HLA-based protection against rheumatoid arthritis. Nat Commun 6:1–11
44. Lee K, Wucherpfennig K, Wiley D (2001) Structure of a human insulin peptide-HLA-DQ8 complex and susceptibility to type 1 diabetes. Nat Immunol 2(6):501–507
45. Yaneva R, Springer S, Zacharias M (2009) Flexibility of the MHC class II peptide binding cleft in the bound, partially filled, and empty states: a molecular dynamics simulation study. Biopolymers 91(1):14–27
46. Henderson KN, Tye-Din JA, Reid HH, Chen Z, Borg NA, Beissbarth T, Tatham A, Mannering SI, Purcell AW, Dudek NL, van Heel DA, McCluskey J, Rossjohn J, Anderson RP (2007) A structural and immunological basis for the role of human leukocyte antigen DQ8 in celiac disease. Immunity 27(1)
47. Bellows M, Fung H, Taylor M, Floudas C, Lopez de Victoria A, Morikis D (2010) New compstatin variants through two de novo protein design frameworks. Biophys J 98(10): 2337–2346
48. Tamamis P, Morikis D, Floudas CA, Archontis G (2010) Species specificity of the complement inhibitor compstatin investigated by all-atom molecular dynamics simulations. Proteins 78(12):2655–2667
49. Schmidt am Busch M, Mignon D, Simonson T (2009) Computational protein design as a tool for fold recognition. Proteins 77: 139–158
50. Schmidt am Busch M, Sedano A, Simonson T (2010) Computational protein design: validation and possible relevance as a tool for homology searching and fold recognition. PLoS One 5(5):10410
51. Mignon D, Simonson T (2015) Sequence exploration in computational protein design with stochastic, heuristic and exact methods (in press)