Computational protein design: the Proteus software and selected applications

Thomas Simonson*,[1], Thomas Gaillard[1], David Mignon[1], Marcel Schmidt am Busch[1,3], Anne Lopes[1,4], Najette Amara[1], Savvas Polydorides[1,2], Audrey Sedano[1], Karen Druart[1], and Georgios Archontis[2]

[1]Laboratoire de Biochimie (CNRS UMR7654), Department of Biology, Ecole Polytechnique, 91128 Palaiseau, France.

[2]Department of Physics, University of Cyprus, Nicosia, Cyprus.

[3]Institut fuer theoretische Physik, Johannes Kepler Universitaet Linz, Altenberger Strasse 69, 4040 Linz, Austria.

[4]Present address: Institut de Génétique Moléculaire, Université de Paris-Sud, Orsay, France

*Email: thomas.simonson@polytechnique.fr

Abstract

We describe an automated procedure for protein design, implemented in a flexible software package, called Proteus. System setup and calculation of an energy matrix are done with the XPLOR modelling program and its sophisticated command language, supporting several force fields and solvent models. A second program provides algorithms to search sequence space. It allows a decomposition of the system into groups, which can be combined in different ways in the energy function, for both positive and negative design. The whole procedure can be controlled by editing 2–4 scripts. Two applications consider the tyrosyl-tRNA synthetase enzyme and its successful redesign to bind both O-methyl-tyrosine and D-tyrosine. For the latter, we present Monte Carlo simulations where the D-tyrosine concentration is gradually increased, displacing L-tyrosine from the binding pocket and yielding the binding free energy difference, in good agreement with experiment. Complete redesign of the Crk SH3 domain is presented. The top 10000 sequences are all assigned to the correct fold by the SUPER-

FAMILY library of Hidden Markov Models. Finally, we report the acid/base behavior of the SNase protein. Sidechain protonation is treated as a form of mutation; it is then straightforward to perform constant-pH Monte Carlo simulations, which yield good agreement with experiment. Overall, the software can be used for a wide range of application, producing not only native-like sequences but also thermodynamic properties with errors that appear comparable to other current software packages.

1 Introduction

Computational protein design (CPD) continues to develop as an important tool for biotechnology [1–8]. Early applications led to proteins with novel ligand-binding functions [9, 10], novel enzyme activity [11], and proteins that were completely "redesigned": around 2/3 of their sequence was mutated, yet their structure and stability were retained [12]. In the last few years, CPD has allowed the creation of new protein folds [13–15], completely new enzymes [16–18], and the assembly or deassembly of multiprotein complexes [19–23].

CPD methods are mainly characterized by (a) the energy function, (b) the description of the folded protein's conformational space, (c) the treatment of the unfolded state, and (d) the search method used to explore sequences and conformations. While the search method is important for efficiency, the accuracy of the results are mainly determined by the first three ingredients, especially the energy function. Energy functions from molecular simulations are developed from first principles [24–28] and have the capability to predict protein structure, stability, and ligand binding with a high accuracy [29–33]. In a CPD context, however, additional approximations are necessary, so that the energy function is modified, both by the use of an implicit solvent model [34–38] and through additional, empirical, contributions [2, 3, 12, 13, 39–42].

Several software implementations have been reported. The Rosetta suite is currently the most successful and widely used [12, 13, 15, 18, 41, 43], but others exist and have also been successful [6–10, 44–54]. They differ in the characteristics (a–d) listed above, the range of choices offered for each one, the degree of empiricism of the energy function, the applicability to different classes of molecules, the mode of user interaction, the availability and ease of development of source code, and so on.

Here, we describe a software implementation, Proteus 2.0, that significantly extends and improves an earlier one [54–56]. Its three main components are (1) the molecular simulation program XPLOR [57], with its capability to describe biomolecular

interaction energies; (2) a sophisticated set of scripts, written in the XPLOR scripting language [57, 58], that control the calculation of an energy matrix for the system of interest [59]; (3) a C program, "proteus", for exploring the space of sequences and conformations using various search algorithms, including a mean field and a Monte Carlo method. XPLOR can be downloaded by academic users from the Yale University web site, whereas local modifications to XPLOR, our XPLOR scripts, the proteus source code, and documentation are available on request (and will soon be available online). The software is modular and flexible, allowing the use of four different molecular mechanics force fields, four solvent models, including two Generalized Born (GB) variants [60–62], several rotamer libraries, and a wide range of fitness functions, using any combination of protein stability, ligand affinity, and ligand specificity, including positive and negative design. The current version does not allow the most recent methods for flexible backbone design [63–65], but calculations can be performed using an ensemble of predefined backbone conformations.

This and the earlier implementation have been used for several applications with good success. One application consisted in redesigning 95 small proteins from six structural families, then using the designed sequences to perform homologue searching [66, 67]. For this application, the energy matrix calculations were ported to a volunteer distributed computing framework, based on the Berkeley Open Infrastructure for Network Computing [68], and made available through our Proteins@Home project, in which over 20,000 volunteers participated [66, 67]. Where comparison was possible, the quality of the designed sequences was comparable to several other CPD implementations. Over 85% of the designed sequences were assigned to their correct SCOP family by the SUPERFAMILY library of Hidden Markov Models for fold recognition [69]. We also tested their capability to retrieve natural homologues from sequence databases. Using low energy designed sequences, we could retrieve 60–70% of known SH2, SH3, and PDZ domains and around 90% of known Kunitz-type inhibitors and interleukin-8 chemokines.

A second application consisted in redesigning the asparaginyl-tRNA synthetase enzyme to decrease binding of its natural substrate asparagine (Asn) and increase binding of the substrate analogue aspartate (Asp) [70, 71]. The best designed sequences did not display detectable catalytic activity; nevertheless, MD simulations and Poisson-Boltzmann free energy calculations gave good evidence that they did indeed have a strongly reduced Asn binding and increased Asp binding, compared to the native enzyme [71]. A third application (not yet published) led to a successful redesign of the

stereospecificity of the tyrosyl-tRNA synthetase enzyme, with one designed variant having a preference for the substrate analogue D-tyrosine, with respect to the natural substrate L-tyrosine. A fourth application used the software in a somewhat different way, to study acid/base changes ("mutations") in a test set of 12 proteins [31, 72]. Indeed, the CPD problem maps precisely onto the problem of computing acid/base constants (pK_a's) for protein sidechains, as long as one uses an appropriate algorithm (constant-pH Monte Carlo) to sample the Boltzmann ensemble of conformations and protonation states. Calculations with the Amber ff99SB force field and a good GB variant gave good agreement between computed and experimental pK_a's, with rms errors of about 1.2 pK_a units, almost as good as the widely-used PropKa program [73, 74] (0.8 pK_a units for the same test). Finally, sidechain reconstruction tests gave good results earlier [62], and better results very recently with the superior, Amber force field: for nine proteins, we obtained 77% of sidechains with correct χ_1 and χ_2 angles, compared to 81% with the widely-used SCWRL4 program [75] (which uses a much larger rotamer library); these tests will be published elsewhere.

Below, we describe the methods used in our current software and some illustrative applications. In the "Theoretical Methods" section, we describe the energy function, including the solvent and unfolded state models. The energy function combines a molecular mechanics treatment of the protein(s) and any ligands with an implicit model of the solvent. When a Generalized Born (GB) solvent model is used, we introduce a "Native Environment" approximation for each sidechain, described below. It allows the total energy to be expressed as a sum of one- and two-residue terms. This makes possible the usual two-stage CPD procedure, introduced by Mayo and coworkers, where an energy matrix is precomputed, then used during a second, sequence exploration stage [59]. The solvent model also includes terms that depend on the solvent-accessible surface area of each atom. Here, too, approximations are used to reduce the surface areas to a sum over one- and two-residue terms [62, 76]. For the unfolded state, we use a simple model, where the protein is viewed as a collection of independent tripeptides [42, 54]. The energy of these tripeptides includes an empirical correction for each amino acid type [13, 39, 77], optimized so that the abundancies of each type match some target values, such as the natural frequencies in a given protein family.

In the same, theoretical section, we describe the Monte Carlo exploration of sequence space and the relation between the sampled distribution of sequences and conformations and the folding free energy of each sequence. We include the special case where "sequence" changes correspond to changes in the sidechain protonation states.

We also describe the mean field exploration method and a heuristic method.

In the next, Computational Protocols section, we describe the computational steps and their implementation in a typical design calculation. These include the setup of the system for XPLOR, the positioning of sidechain rotamers on the protein backbone, the pre-calculation of GB solvation radii, the calculation of diagonal and off-diagonal energy matrix terms (the most expensive step), the exploration of sequence and conformation space. Many of the details are in the Supplementary Material.

Finally, we describe some illustrative applications. We focus on practical aspects and difficulties, and features that are specific to the present, most recent software implementation. We first consider the design of two protein:ligand complexes. Both involve the tyrosyl-tRNA synthetase enzyme, with two possible substrate analogues as its ligand: O-methyl-tyrosine (me-Tyr) and D-tyrosine (D-Tyr). In particular, we report MC simulations where L-Tyr and D-Tyr are both present, but the concentration of D-Tyr is gradually increased so that it displaces L-Tyr from the binding pocket. The mid-point concentration can then be interpreted as a binding free energy difference. Next, we consider whole protein design, with the c-Crk SH3 domain as an example. Finally, we describe the calculation of acid/base constants for the SNase protein. The calculation of thermodynamic properties in the D-Tyr and SNase applications helps to illustrate the accuracy of our free energy function and structural models. Additional thermodynamic calculations (such as stability changes due to point mutations) will be reported elsewhere. Full details on the computational methods used for the present applications are given in Supplementary Material.

2 Theoretical Methods

2.1 Energy function: general form

We use a molecular mechanics energy function along with an implicit solvent model [62]. The molecular mechanics parameterizations that are currently available within Proteus correspond to the Charmm19 force field [78] and the ff99SB version of the Amber force field [25, 79]. XPLOR also allows a "polar hydrogen" version of OPLS [80] and the Charmm22 force field [24], but these two parameterizations are not yet implemented within the Proteus CPD scripts. Since the energy function includes a contribution from the implicit solvent, it should be viewed as a Potential of Mean Force, or PMF [34, 81].

Four main solvent models are available [62]. The two simplest use a simple dielectric screening function to reduce the Coulombic interactions between protein atoms: either a uniform screening factor ("CDIE" option in XPLOR) or a distance-dependent screening factor ("RDIE" option) [62]. The other two solvent models use a more complex, Generalized Born (GB) screening function. The GB energy contribution has the form:

$$E^{\mathrm{GB}} = \frac{\tau}{2} \sum_{ij} q_i q_j g_{ij}, \tag{1}$$

where the sum is over all pairs of protein charges, $\tau = \frac{1}{\epsilon_w} - \frac{1}{\epsilon_p}$, ϵ_p is the protein dielectric constant, ϵ_w is the solvent dielectric constant (80 at room temperature), and g_{ij} represents the interaction (or Green's function) between a unit charge at the q_i position and the solvent polarization induced by another unit charge at the q_j position. This last quantity is approximated by

$$g_{ij} = \left(r_{ij}^2 + b_i b_j \exp[-r_{ij}^2 / 4 b_i b_j] \right)^{-1/2} \tag{2}$$

where r_{ij} is the distance between atoms i, j and b_i is the "solvation radius" of atom i. This radius approximates the distance from i to the protein surface and is a function of the coordinates of all the protein atoms. Two GB implementations are available in XPLOR and Proteus: the "GB/ACE" version of Schaefer & Karplus [82, 83] and the version of Truhlar and coworkers [84], which we refer to as "GB/HCT". They differ by the method used to compute the b_i; for details, see Moulinier et al [61].

All four screening functions above can be combined with a solvent accessible surface energy term:

$$E_{surf} = \alpha \sum_i \sigma_i A_i \tag{3}$$

The sum is over the protein atoms i, A_i is the solvent accessible surface area of atom i, σ_i is an atomic solvation coefficient (measured in kcal/mol/Å^2) that depends on the atom type, and α is an overall scaling factor for the surface term.

2.2 GB energy term: approximation as a sum over atom pairs

Because the solvation radii b_i depend on all the protein coordinates [60, 61, 82, 84], the GB energy contribution E^{GB} cannot be expressed as a sum over atom or residue pairs, violating the two-stage, Mayo CPD protocol [59]. To avoid this difficulty, various approximations have been proposed [38, 45, 85–87]. Proteus uses the following simple method [71]. Consider two different residues, I and J, with given sidechain types and

rotamers. Let i and j be atoms, belonging to I and J, respectively. When computing the I, J interaction energy, a contribution $\frac{\tau}{2} q_i q_j g_{ij}$ arises from the i, j pair. To compute b_i in g_{ij} (Eq. 2), we assume the whole protein except for I is in a fixed, reference conformation with a fixed, reference sequence; similarly for b_j. The reference sequence and conformation are normally taken from the experimental native structure (but other choices are possible). We refer to this as the "Native Environment", or NE approximation. Notice that for each i, j pair, the NE approximation assumes three different sequences when computing b_i, b_j, and g_{ij}. The net effect is that each residue pair I, J experiences an effective, native-like, dielectric environment. For sidechain:backbone interactions, the same method applies, taking J to represent the backbone (assumed to have a fixed conformation). In practice, all the b_i are precomputed at the same time as the diagonal I, I terms of the energy matrix (see below). The quality of the NE approximation is quite good, as illustrated by the successful applications below and described in detail elsewhere (manuscript in preparation).

2.3 Accessible Surface Area energy term: approximation as a sum over atom pairs

Surface areas are computed using the Lee and Richards algorithm [88], implemented in the XPLOR program [57]. For reasons of efficiency, following Street & Mayo [76], we assume that A_i can be obtained by summing the contact areas A_{ij} between atom i and its neighbors j, and subtracting the contact, or solvent-inaccessible area $C_i = \sum_j A_{ij}$ from the total area of atom i. This approximation has the enormous advantage that the surface energy takes the form of a sum over pairs of amino acids. However, it leads to a systematic error, since the contact areas can overlap: a portion of atom i can be in contact with two atoms j and j' at a time. Street and Mayo showed, and we confirmed [62] that the systematic error can be largely corrected by applying a scaling factor of less than one to contact areas A_{ij} that involve at least one buried atom (i or j); for details, see [62]. In earlier work, we did extensive testing and comparison of several different sets of surface parameters, based on sidechain reconstruction, protein solvation energies, and mutations of over 1000 sidechains (including buried sidechains) [55, 62]. Details on our current implementation and its accuracy will be published elsewhere.

2.4 Unfolded energy: approximation as a sum over residues

In the unfolded state model, following earlier studies [42, 59, 89], the amino acid sidechains do not interact with each other, but only with nearby backbone and with solvent. One way to implement this idea is to consider that each amino acid X is part of a tripeptide with a sequence Ala-X-Ala and a given backbone geometry. In practice, we and others have found that this simple model should be supplemented by an empirical energy correction, e_X, optimized so that the overall amino acid compositions are reasonable when whole proteins are designed [13, 39, 54, 77]. Notice that if a correction is to be added, we can simply view the whole contribution E_X of each amino acid as an empirical quantity that depends only on the amino acid type of X, without referring to tripeptides.

To optimize the E_X values, we typically consider a test set of proteins and determine the experimental amino acid frequencies $f_X^{\rm exp}$ for a set of experimental homologues. We then proceed iteratively, with the E_X initially set to a plausible starting guess. At each iteration, several thousand sequences are computed for each test protein. The corresponding amino acid frequencies, $f_X^{\rm calc}$, averaged over all sequences, proteins, and amino acid positions, are compared to the experimental frequencies $f_X^{\rm exp}$. The energy E_X is then modified according to the Boltzmann-like relation:

$$E_X^{\rm new} = E_X^{\rm old} + 0.5 \ln \frac{f_X^{\rm exp}}{f_X^{\rm calc}}. \tag{4}$$

With this scheme, if a given type X is too abundant in the designed sequences, Eq. (4) leads to an increased stability of the unfolded state when X is present, so that X will be less abundant in the next round. After about ten rounds, the frequencies converge to the target values and the procedure can be stopped. Illustrative values are shown in Fig. 1 and Table 1, and compared to the values obtained with the tripeptide method. The E_X values shown were optimized for the SH3 protein family, using CPD runs with the Amber ff99SB force field, the GB/HCT solvent model, and a protein dielectric constant of 16. The tripeptide values were computed with the same energy function, and averaged over all positions in two proteins (SNase and lysozyme) and over 800 conformations (rotamer sets) for each protein. The dispersion of the tripeptide energies is comparable to the size of the dots in Fig. 1 and is not visible. The two sets are very similar, which shows that only small empirical tuning of E_X is needed to reproduce the experimental amino acid frequencies.

2.5 Sequence exploration: heuristic and mean field optimization methods

Proteus allows three sequence exploration methods. The first is a mean field approach, which has been described elsewhere [62, 90–92]. This method calculates iteratively the Boltzmann probability $P(i,k)$ of each rotamer k of each residue i, which is related to the mean energy $E(i,k)$ of sidechain i:

$$E(i,k) \propto -k_B T \ln P(i,k). \tag{5}$$

$E(i,k)$ is the Boltzmann average of the interaction energy between sidechain i and its environment; k_B is the Boltzmann constant and T is the temperature. Since the protein backbone is fixed, we can write

$$E(i,k) = E_{BB}(i,k) + \sum_{j \neq i} \sum_{l} E(ik,jl) P(j,l) \tag{6}$$

where E_{BB} is the interaction energy with the backbone, the first sum is over protein sidechains j, the second sum is over the rotamers l of sidechain j, and $E(ik,jl)$ is the interaction energy between sidechains i and j when they occupy rotamers k and l. We assume the optimal sidechain positions correspond to the most probable rotamers.

The second exploration method is a heuristic procedure developed by Wernisch et al. [54, 89]. A "heuristic cycle" proceeds as follows: an initial amino acid sequence and set of sidechain rotamers are chosen randomly. These are improved in a stepwise way. At a given amino acid position i, the best amino acid type and rotamer are selected, with the rest of the sequence held fixed. The same is done for the following position $i+1$, and so on, performing multiple passes over the amino acid sequence until the energy no longer improves (or a set, large number of passes is reached), and the cycle ends. The method can be viewed as a steepest descent minimization, starting from a random sequence, and leading to a nearby, local, (folding) energy minimum. For design calculations, we typically perform ∼500.000 heuristic cycles for each protein, thus sampling a large number of local minima on the energy surface.

2.6 Sequence exploration: statistical mechanical framework and Monte Carlo method

The third exploration method in Proteus is a Monte Carlo one. This method has a considerable advantage, since it leads to a distribution of sidechain types and rotamers

that is rigorously defined and controlled by the user. Thus, the Metropolis Monte Carlo algorithm leads to a Boltzmann distribution, which is thermodynamically correct for ordinary molecular systems [93, 94]. However, the CPD context is unusual, since it involves a protein whose sidechain types can fluctuate, a situation that might appear unphysical. In fact, this situation can be modelled as a large collection of variants of the given protein P, say P_1, P_2, $\cdots$. In principle, this collection should include all possible mutants of P, and each mutant should be present in large numbers. A Monte Carlo move that mutates one variant P_i (with a specific set of rotamers), into another, P_j, can then be viewed as a conformational change, where P_j changes from its unfolded to a folded state, while P_i changes from the folded to the unfolded state. Thus, the CPD situation maps onto an ordinary molecular system; the partition function has the usual form [95], and the Boltzmann distribution of sidechain types and conformations has a simple physical interpretation. If we assume P_i and P_j differ by a single mutation, the ratio of their Boltzmann probabilities has the form:

$$\frac{P_j}{P_i} = \exp\left[-\beta\left((E_j - E_i) - (E_{X_j} - E_{X_i})\right)\right]. \tag{7}$$

Here, $\beta = 1/k_B T$, E_i and E_j are the energies of the folded P_i and P_j (with the chosen sets of rotamers), and the energies E_{X_i}, E_{X_j} represent the contribution of the mutated sidechain to the unfolded state energy before and after the mutation. The exponent on the right is the energy change that enters into the Metropolis Monte Carlo test.

When the "mutations" take the form of sidechain protonation/deprotonation, the problem is the classic one of constant-pH Monte Carlo [31, 96–100]. If we compare two states that differ by the addition of a proton to a specific titratable sidechain j, with the protein in a given conformational state, say J, the ratio of Boltzmann probabilities has the form [31]:

$$\frac{P_J(\mathrm{prot})}{P_J(\mathrm{deprot})} = \exp\left[-\beta(\Delta E_J - \Delta E^j_{\mathrm{model}}) + 2.303(\mathrm{pK}^j_{\mathrm{a,model}} - \mathrm{pH})\right]. \tag{8}$$

Here, the energy changes correspond to protonation of, respectively, the folded protein (ΔE_J) and a small model compound ($\Delta E^j_{\mathrm{model}}$) that is chemically similar to the titrating sidechain j and whose experimental $\mathrm{pK}^j_{\mathrm{a,model}}$ is known. The energy change associated with the model compound is $\Delta E^j_{\mathrm{model}} + 2.303kT(\mathrm{pK}^j_{\mathrm{a,model}} - \mathrm{pH})$. This quantity plays the same role as the unfolded energy change $E_{X_j} - E_{X_i}$ in the case of the sidechain mutation.

3 Computational protocols: general features

3.1 The energy matrix: system setup and XPLOR scripts

With a pairwise energy function and a finite conformational space, the residue:residue interaction energies can be precomputed and stored in an energy matrix [101]. Here, the calculation is done with XPLOR [57]. XPLOR has its own scripting language, like its ancestor, CHARMM [102] and its descendants, CNS [58, 103] and NIH-XPLOR [104]. The system setup, the calculation of the diagonal and off-diagonal blocks of the matrix are done using a sophisticated set of XPLOR scripts (about 4000 lines). Only three of them are normally edited by the user to specify the details of the design, including the choice of force field, solvent model, rotamer library, and the position and nature of the allowed mutations. A few shell scripts automate the whole procedure.

The system to be designed is first set up with XPLOR in the usual way for a molecular mechanics study [57]. The chosen force field is implemented through molecular topology and force field parameter files. Both the Charmm19 polar hydrogen and the Amber ff99SB all-hydrogen protein force fields are fully supported for CPD. The OPLS polar hydrogen and Charmm22 all-hydrogen force fields are supported by XPLOR but not fully implemented in our CPD procedure. Two GB variants are supported: GB/ACE (compatible with Charmm19) [82, 83] and GB/HCT (compatible with ff99SB) [61, 62, 84, 105].

As usual in molecular mechanics, the system is divided into "residues", which usually include backbone and sidechain moieties. The backbone and possibly some of the sidechains are held fixed, or "frozen". The other sidechains can be "active" (allowed to mutate) or "inactive" (they do not mutate). Ligands can be present and can be frozen, inactive, or active. For each non-frozen ligand, sets of rotamers must be provided by the user. To allow mutations, the XPLOR setup includes a step where (a) all possible sidechains are grafted (or "patched") onto each active backbone position. Later, at any given step of the Monte Carlo exploration, only one or two at a time will interact with the rest of the system, the others behaving as dummies. A similar approach is used for an "active" ligand: all possible ligand types are added to the system. At this stage, the list of allowed sidechain types can be manually edited to impose restrictions at particular sites. For example, one residue might be allowed to explore only nonpolar types, while another has only acid/base activity (switching between different protonation states). Such restrictions can also be imposed readily at later stages (see below).

With the molecular topology in place, the setup includes several further steps. First, (b) for each protein residue, a discrete set of possible conformations is drawn from a rotamer library; the corresponding conformations are constructed within XPLOR and the sidechain coordinates saved. The result of (b) is a collection of PDB files, one for each residue (active or inactive) and each sidechain type and rotamer at that position (around 200 files for a single active sidechain). For each position, type, and rotamer, (c) the Born solvation radii are computed with XPLOR, with all the other positions occupying their native type and conformation. These radii are written to a single file. For each position, type, and rotamer, (d) the intra-sidechain and sidechain:backbone interaction energies are computed and stored in a file. These energies correspond to the diagonal of the energy matrix. The energy is actually computed after a short energy minimization (usually around $N_{\min} = 15$ steps), with only the current sidechain allowed to move (see Supplementary Material). This minimization is designed to alleviate the rotamer approximation. Finally, (e) rotamers that have high backbone:sidechain energies are eliminated (by a shell script); for each residue and sidechain type, a file is produced containing the list of retained rotamers.

Steps (b-d) are performed with a single XPLOR script. Compared to the original XPLOR 3.854 distribution (http://www.csb.yale.edu), there are very few modifications to the XPLOR code itself. The main one is the GB implementation [61]. String arrays have also been added to the script language and XPLOR string variables can now be assigned directly from the shell command line. Amber force field files have been created (including extensions for some minor sidechain protonation states).

With the setup in place, calculation of the remaining, off-diagonal matrix blocks is done automatically, with a shell script either submitting individual pairs of positions to a PBS batch queue system or running them directly from the shell. Each pair is computed by a single XPLOR script, which loops over allowed residue types and rotamers. As for the diagonal matrix terms, the energy is computed after a short minimization ($N_{\min}$ steps), with only the current pair allowed to move. Notice that there is no further, on-the-fly minimization during the subsequent Monte Carlo exploration. In applications, it may be necessary to adjust $N_{\min}$ empirically. The individual energy terms are written to a file, with a verbosity level that is set by the user. With the highest verbosity, the files contain enough information to modify the energy function substantially *a posteriori* (with a perl script): the solute dielectric constant, the atomic surface coefficients, and the unfolded energies E_X can all be modified without recomputing the matrix, so that many different parameterizations can be tested efficiently.

Modifying the E_X can also allow different experimental conditions to be modelled, including different pH values or ligand concentrations (see below).

3.2 Sequence exploration: the proteus program

The next step is to explore sequence and conformation space. This is done with a second program, called proteus, written in C. The program is controlled by a command file with an XML format and a simple syntax. Individual commands (with sensible defaults) are used to control the exploration method (mean field, heuristic, or Monte Carlo), the number of exploration steps, the temperature, details of the Monte Carlo move scheme, the starting sequence and conformation, and locations of input files (the energy matrix) and output files (sequences, energies). Possible Monte Carlo moves are rotamer and/or type changes for one or two residues at a time. Backbone moves are not currently supported, although multiple backbone conformations can be present.

Individual commands can also be used to apply restrictions to the system in a flexible and powerful way. In the matrix calculation, above, individual residues were assigned a list of possible types and rotamers, defining an exploration space $\mathcal{S}_M$. Here, the system can be restricted to a subset of $\mathcal{S}_M$. The simplest example is to make an active amino acid keep its native type. Another example is to make it occupy just one or a few rotamers. More complex examples are given below.

Another command allows one to define groups of residues. These, in turn, can be used to impose additional restrictions within $\mathcal{S}_M$. For example, one group of residues can be made to behave as a copy of another, occupying the same amino acid sequence. The two copies might correspond to two particular backbone conformations that we want to select for or against. They could also correspond to a protein with or without a bound ligand; the holo and apo copies would then have the same sequence but explore different rotamer sets. In the first example (two backbones), each group would be present as a distinct physical object in the XPLOR setup and the energy matrix. In the second example, however (apo *vs.* holo), only one physical object is needed in the energy matrix; the second, virtual copy is created within proteus.

Finally, a single proteus command is used to define one or more energy functions, which will drive the exploration. By default, the total energy is used (which includes the contribution of the unfolded state, through the E_X). However, more complex functions can be constructed that treat individual groups differently. For example, arbitrary weights can be applied to individual groups or group interactions. Weights

can be zero, so that the corresponding interactions are ignored. They can be positive or negative, so that the interactions are selected for or against, allowing both positive and negative design. A threshold can also be applied to individual group interactions; if the energy increases beyond the threshold, its value is replaced by the threshold. This allows a particular energy to help drive the exploration only when its value is in a particular range; for example the folding free energy might contribute fully only when it drops below a threshold. In the context of Monte Carlo exploration, we would typically combine the individual group or group pair combinations into a weighted sum. All these mechanisms (groups, space restrictions, energy weights and thresholds) can be used easily to build up complex models involving multiple backbone conformations, multiple ligands, and complex optimization criteria, involving both positive and negative design.

4 Selected applications

4.1 The complex between tyrosyl-tRNA synthetase and O-methyl-tyrosyl adenylate

Tyrosyl-tRNA synthetase (TyrRS) attaches tyrosine (Tyr) to the appropriate ("cognate") $tRNA^{Tyr}$, establishing the link between the amino acid type and the nucleotide triplet that forms the anticodon within the tRNA. This and other aminoacyl-tRNA synthetases have been extensively engineered to modify their preferred amino acid substrate, replacing the wildtype substrate by a variety of analogues [106, 107]. The engineering has been done by experimental directed evolution, and has allowed the genetic code to be expanded to include non-natural amino acids, such as O-methyl-tyrosine (me-Tyr). We have performed a similar engineering using CPD. We started from a mutant TyrRS, which differs from the wildtype, archaeal, *Methanococcus jannaschii* enzyme at four positions. The mutations were selected experimentally to enhance me-Tyr binding and activity [106]. The crystal structure of the mutant TyrRS, without the amino acid ligand, is known (PDB code 1U7X) [108]; it has a C_α deviation of 0.7 Å with respect to the wildtype apo-enzyme [108]. Three of the mutations are close to the ligand; one (Glu107Thr) is far away (12.5 Å from the substrate sidechain). The backbone structural changes are mainly localized near the Leu162Pro mutation, which shortens a helix by two amino acids.

In our redesign, we kept the backbone fixed in the mutant, 1U7X conformation. We considered the enzyme bound to O-methyl-tyrosyl adenylate (me-TyrAMP), a stable

intermediate formed from me-Tyr and ATP (in the absence of tRNA). The ligand's backbone was fixed in the conformation seen in a close orthologue (from yeast: PDB code 2DLC) [109]. We kept the 1U7X sidechain types at two of the mutated positions: Thr107 (which is distant) and Pro162 (which affects the backbone). The other two positions that are mutated in 1U7X, Tyr32Gln and Asp158Ala, were treated as active, and allowed to mutate freely (but not to Gly, Cys, or Pro). Other sidechains within 16 Å of the me-Tyr sidechain were inactive, so that they can explore their rotamer space but not mutate. More distant sidechains were held fixed, or "frozen". The me-Tyr sidechain had the same rotamers as a Tyr sidechain, including four possible orientations of the methyl, relative to the phenyl ring: two in-plane and two perpendicular. Calculations were done with the Charmm19 force field, the CASA solvent model with a uniform dielectric constant of 8, and a surface area energy term, with the following atomic surface coefficients (in kcal/mol/Å^2): alkane atoms = -0.005; polar atoms = -0.08; aromatic atoms = -0.04; ionic atoms = -0.10; hydrogens = 0. The dielectric value is lower than the value used earlier (16) for several protein:ligand binding tests, including Tyr binding to TyrRS mutants [55, 70], but gives good results, possibly because the redesign of TyrRS to bind a methylated ligand is not very sensitive to the electrostatic treatment. We used a rotamer library of Tuffery et al from either 1995 or 2003 [110], and the heuristic exploration method [89]. Additional details, including the unfolded energy values E_X (needed for the two active positions) are given in Supplementary Material.

Results are summarized in Table 2. Six sequences were obtained, shown with their mean ligand binding (free) energies. Among the 289 theoretically possible sequences, the experimental 1U7X sequence (Gln32, Ala158) is correctly predicted, and has the second highest binding affinity. The Gln32 sidechain found in 1U7X is actually retained in four out of six mutants, even though it was free to mutate into 16 other types. The experimental Ala158 mutation is correctly predicted for the 2nd highest-scoring sequence. With this mutant, the Gln32 sidechain occupies a rotamer similar to the crystal structure. The me-Tyr sidechain phenyl has an orientation similar to the wildtype holo-enzyme; its methyl points towards the Gln32 sidechain. The other five high scoring sequences are QL, AL, QQ, AA, and QF; all six mutants are within 1.3 kcal/mol of each other in terms of ligand binding affinity. Performing short MD simulations (2 ns each) and estimating the binding free energies with a Poisson-Boltzmann model (PB) [33, 71, 111], the experimental, QA mutant is ranked fourth, with a binding free energy within 2.2 kcal/mol of the top mutant. The mutants QL and QQ have

poorer affinities (by about 3 kcal/mol) according to the PB model. It may be that the other three high-scoring mutants, AL, AA, and QF also have an experimental activity for me-Tyr binding and possibly catalysis, even though they were not selected by the experimental directed evolution [106].

4.2 The complex between tyrosyl-tRNA synthetase and D-tyrosine

Another goal we have pursued is to change the TyrRS stereospecificity, switching from an L-tyrosine to a D-tyrosine preference. A TyrRS with inverted specificity could potentially be used to help incorporate D-tyrosine into proteins *in vivo*. Selected results are shown here, as another illustration of Proteus capabilities. More details will be published elsewhere. Experimental work showed recently that a single amino acid substitution in the *Escherichia coli* enzyme (Asp81Arg), suggested by the computational design, does indeed lead to an inverted specificity, with a distinct preference of the mutant enzyme for D-Tyr (P. Plateau and S. Ye-Lehmann, personal communication). In Figure 2, we show the results of MC simulations of both the wiltype and mutant enzyme, where the amino acid ligand is allowed to freely adopt either the L- or D-stereoisomer. In effect, the ligand is treated as active, and allowed to mutate between two types. The simulations are performed with the Amber ff99SB force field, the GB/HCT solvent model (with a protein dielectric constant of 8), a surface area term (as above), and the Tuffery rotamer library. The concentration of L-Tyr is held fixed, while the D-Tyr concentration is gradually increased. This is achieved by adding a term $\delta E = kT \log[\text{D-Tyr}]$ to the energy of the unbound D-Tyr (E_X in Eq. 7), a method that is precisely analogous to pH variation in constant-pH MC simulations [31, 100, 112]. For the matrix elements involving the non-natural ligand, D-Tyr, we explored the possibility of using a slightly larger number $N_{\min}$ of minimization steps than for the natural ligand and the intra-protein terms. Good results were obtained using $N_{\min} = 25$ for matrix elements involving D-Tyr and 15 for the rest of the matrix. We also tested different protein dielectric constants, obtaining comparable results with 2, 4, 6, and 8.

Figure 2 shows the titration curves for both the wildtype and mutant TyrRS as [D-Tyr] is increased, using a protein dielectric of two. The fraction of bound D-Tyr increases from zero to one, with midpoints of 1.8 and 1.4 kcal/mol, respectively. Each midpoint value represents the binding free energy difference between L- and D-Tyr, with

the positive sign indicating a preference for L-Tyr. The computed wildtype preference is a bit larger than experiment (about 1.3 kcal/mol; P. Plateau and S. Ye-Lehmann, personal communication). The smaller value for the mutant indicates a reduced preference for L-Tyr. However, the mutant value is larger than the experimental one, which is known to be negative, though its precise value could not be measured. Nevertheless, with reasonable choices for the two adjustable parameters, ϵ_P and the D-Tyr $N_{\min}$, the qualitative behavior and error magnitudes are reasonable. In addition, the results are rather robust, since with $N_{\min} = 15$ for D-Tyr, the wildtype/mutant difference is similar, and the different dielectric constants only change the free energies by about 1 kcal/mol or less (not shown).

4.3 The Crk SH3 domain and its peptide ligand

SH3 domains are small, all-beta domains of about 60 residues that help control protein:protein binding [113, 114]. We have already used several SH3 domains to help parameterize and test our software [54, 66]. Here, we present some recent results for the complete redesign of the Crk SH3 domain with an improved force field and solvent model, alone or in complex with a deca-peptide ligand (sequence: PPPALPPKKR). We start from the crystal structure of the protein:peptide complex (PDB code 1CKA) [115]. The last, Arg residue of the deca-peptide is missing from the PDB structure and omitted from our model. The system was modelled with the Amber ff99SB all-atom force field and an implicit solvent model that combines the GB/HCT generalized Born variant with a solvent accessible surface area contribution [55, 62, 84]. A large protein dielectric constant of 16 was used, similar to earlier whole protein designs [55, 66, 67]. We used the Tuffery rotamers [110] and unfolded energies E_X that were optimized to reproduce the amino acid abundancies in the SH3 family (Table 1). The entire protein sequence (57 residues) was allowed to vary, except for four Pro and four Gly residues. The peptide, when present, was inactive (fixed sequence, variable rotamers). Sequence exploration was done both with the Monte Carlo method. Full computational details are given in Supplementary Material. In particular, Fig. 1 in Supplementary Material gives a flowchart for the calculations, with a description of the input and output files. In terms of CPU and memory use, the Monte Carlo runs required about 0.6 gigabytes of memory and took a few hours on a single core of a 3 GHz Intel Xeon processor. The energy matrix calculations take much longer. For a single pair of active residues there are around 40,000 possible type/rotamer combinations and the corresponding block of

matrix elements requires about 8 hours on a single core of a similar processor. The overall matrix for 1CKA involves about 1200 such pairs and can require several days, depending on the size of the cluster used. Notice that the matrix can then be edited automatically to change the dielectric constant, surface energy coefficients, or reference energies without further computation.

Results are summarized in Figure 3. Designed and experimental sequences are plotted as logos. The designed sequences correspond to the top 10000 folding energies obtained with Monte Carlo exploration for the protein:peptide complex. Results for the apo-protein are mostly similar (not shown). The experimental sequences correspond to the SH3 seed alignment in the Pfam database (61 proteins) [116]. The native Crk sequence and the top 25 designed sequences (holo-protein) are also shown as an alignment. The designed amino acid types are in good agreement with the Pfam types. 11 positions that make up the hydrophobic core are very well reproduced (red dots between the two sequence logos). Only five positions deviate strongly from the Pfam types; they are highlighted between the two logos (Fig. 3) as crosses. All five are highly exposed at the protein surface, except for position 150, which is an Asp in the native protein and forms a salt bridge with Lys8 in the peptide ligand. In the designed proteins, Lys8 switches to a different, more exposed rotamer, while the native Asp150 is mutated into (mostly) Ala. The peptide Lys8 rotamer allows it to form a salt bridge with nearby Glu149, present in some of the sequences. The functionally important Trp170 is always preserved in the designed sequences, but the neighboring (and more exposed) Trp171 is mutated to Val in most of the top 10000 designed sequences.

The top 10000 Monte Carlo sequences were submitted to the Superfamily library of Hidden Markov Models [69], which detect similarities to proteins and protein families in the SCOP database [117]. All 10000 sequences were correctly assigned, not only to the SH3 family, but also to the correct Crk native structure. The E-values for the SH3 family assignments were around 10^{-10}, compared to 10^{-20} for the native sequence. The E-values for assignment to the Crk structure were around 10^{-3}, compared to $3\ 10^{-5}$ for the native sequence. Fig. 4 illustrates the stability of two of the designed sequences during 20 nanosecond MD simulations in explicit solvent (see details in Supplementary Material). Results are also shown for the wildtype Crk protein and for a sequence designed earlier with the simpler, CASA solvent model [66]. The older design was produced experimentally and found to be only partly structured (I. Guijarro and P. Plateau, personal communication). The newer designs have a significantly improved stability in the MD simulations and are expected to be structured; experimental tests

are underway.

4.4 The acid/base behavior of Staphylococcal nuclease

Sidechain acid/base reactions can be treated as "mutations" and treated with practically the same formalism as protein design, with only minor changes to the software. The main difference is the interpretation of the "unfolded" reference state. Instead of an unfolded protein, the reference state now corresponds to a collection of model compounds in solution. Each model compound is the analogue of a titrating sidechain type, as usual in pK_a calculations for proteins [31, 112, 118, 119]. The reference energies E_X are the energies of each model compound in its optimal rotamer in solution. Here, as an illustration, we report the titration behavior of the protein Staphylococcal nuclease, or SNase. We consider a hyperstable SNase mutant, known as Δ+PHS, for which 17 Asp and Glu sidechain pK_a's are known experimentally [120]. Calculations were done using the experimental backbone structure (PDB code 3BDC), the Amber ff99SB force field, the GB/HCT solvent, a surface area term (same atomic coefficients as for TyrRS above), and a protein dielectric constant of 4. Results with a dielectric of 8 are slightly poorer. Monte Carlo simulations were done at pH values between 0 and 15, every 0.5 pH units, with 26 million MC steps at each pH value. The last 20 million steps at each pH value were used for averaging. The full pH scan (0.8 billion MC steps) takes about 12 hours on a recent laptop computer using a single core.

The titration curves for the first four Asp and Glu sidechains are shown in Fig. 5. The sigmoidal behavior is typical of experimental titration curves. The calculated Hill coefficients (slopes) are between 0.70 and 0.85, within the typical experimental range [100]. For all 17 Asp and Glu sidechains, the mean rms deviation between the computed and experimental pK_a values is 1.45 pH units, compared to 1.2 with the simple Null model and 1.3 with the popular PROPKA program [74, 121]. While this is slightly larger than the errors reported in some recent studies (around 1 pKa unit; reviewed in [112]), it is similar to other studies of SNase, which is considered a challenging benchmark [112, 120, 122, 123]. For example, a study using Rosetta gave an rmsd of over 2 [122]. The largest experimental pK_a shift is for Asp21, and is well reproduced, with a predicted pK_a of 6.5, *vs.* 6.5 from experiment. The largest errors, 2.1–2.6 units, are for Asp19, Asp77, and Glu129. Finally, we note that for a larger test set of proteins and titratable groups, the typical errors are smaller than for SNase, and close to 1.1 pKa unit [31]. More details will be published elsewhere (Polydorides and Simonson, in

preparation).

5 Concluding discussion

The design software described here combines several well-established ingredients: a molecular mechanics energy function, implicit solvent, a fixed backbone, and sidechain rotamers [2, 51, 77]. Sequences are selected based on their folding free energy, using a tripeptide model of the unfolded state. Some approximations are made that allow the energy to be written as a sum over pairs of residues or groups, and allow the energy matrix to be precomputed. This makes the calculations very efficient, allowing billions of MC steps per day on a desktop computer. To alleviate the rotamer approximation, a slight energy minimization is performed before computing each matrix element, inducing small departures from the library rotamers. A drawback is that the energies depend on the number of minimization steps, introducing some uncertainty. An alternative used by some programs is to completely reparameterize the energy function, to adapt it to the space of dihedral internal coordinates, a method that has its own drawbacks. Monte Carlo moves for the protein backbone are not currently supported, but calculations can be done with an ensemble of backbones [66].

The Proteus software is made up of the XPLOR molecular modelling program (modified locally), a sophisticated collection of scripts written in the XPLOR command language (totalling about 4000 lines), a C program, proteus, for sequence/conformation exploration, and a set of perl and shell scripts. Source code is freely available to academics or anyone with an XPLOR source code license. Energy matrix calculations are mainly controlled through three XPLOR scripts and two shell scripts. Sequence/conformation exploration is controlled by one main proteus script (with an XML format). System setup is done with XPLOR, in a way that is highly automated and similar to other modelling programs, like Charmm or Amber.

The software is designed to be flexible, allowing several molecular mechanics force fields and solvent models to be used, and any rotamer library, including backbone-dependent ones. The system can be decomposed into groups, which can be present with multiple copies and contribute to the energy in various ways. For example, protein sequences can be selected based on stability, binding affinity for one ligand, and specificity relative to another ligand [71]. The software has some unusual features, illustrated in the applications above. Thus, acid/base activity is fully supported, so that sidechain titration curves can be obtained easily, and protonation states can vary

when nearby positions mutate. In addition, multiple ligands can be present and interconvert via Monte Carlo "mutations", as shown for TyrRS. Here too, titration curves are easily obtained, giving estimates of the binding free energy differences between the ligand species.

The applications above included classic sequence exploration for the small protein Crk and a TyrRS:ligand complex, with the latter leading to an experimentally-active sequence. We also described calculations of thermodynamic properties, including TyrRS:L-Tyr/D-Tyr binding free energy differences and SNase acid/base constants. These applications illustrate the need to adjust certain parameters empirically for different applications, especially the protein dielectric constant with GB, but also the atomic surface coefficients and the unfolded energies. This is not expensive, thanks to the matrix editing capability with the more verbose matrix formats. For the acid/base calculations, accuracy is comparable to several other recent software tools [31, 112]. The ligand binding calculations are sensitive to the number $N_{\min}$ of minimization steps performed for each matrix element, and their accuracy was only qualitative for the TyrRS application. More work is needed for this type of application to evaluate and improve the Proteus performance. Its potential for large-scale, competitive ligand binding simulations will be reported elsewhere. We believe that by doing sequence exploration and thermodynamic calculations with the same software and energy function, we are more likely to identify physically meaningful parameterizations of the model, and successfully design a wide range of new proteins.

Acknowledgements

This work was supported by the Agence Nationale pour la Recherche (High Performance Computing program; ProtiCAD project). Some of the calculations were done using the French national supercomputer center CINES. We thank Alexey Aleksandrov, Seydou Traoré, Nicolas Panel and Jialin Liu for discussions.

References

[1] Baker, D. Prediction and design of macromolecular structures and interactions. *Phil. Trans. R. Soc. Lond. 361* (2006), 459–463.

[2] Butterfoss, G. L., and Kuhlman, B. Computer-based design of novel protein structures. *Ann. Rev. Biophys. Biomolec. Struct. 35* (2006), 49–65.

[3] Guérois, R., and Lopez de la Paz, M., Eds. *Protein Design: Methods And Applications*. Humana Press, 2007.

[4] Lippow, S. M., and Tidor, B. Progress in computational protein design. *Curr. Opin. Biotech. 18* (2007), 305–311.

[5] Pleiss, J. Protein design in synthetic biology. *Curr. Opin. Biotech. 22* (2011), 611–617.

[6] Pantazes, R. J., Greenwood, M. J., and Maranas, C. D. Recent advances in computational protein design. *Curr. Opin. Struct. Biol. 21* (2011), 467–472.

[7] Saven, J. G. Computational protein design: engineering molecular diversity, nonnatural enzymes, nonbiological cofactor complexes, and membrane proteins. *Curr. Opin. Chem. Biol. 15* (2011), 452–457.

[8] Samish, I., MacDermaid, C. M., Perez-Aguilar, J. M., and Saven, J. G. Theoretical and computational protein design. *Ann. Rev. Phys. Chem. 62* (2011), 129–149.

[9] Looger, L. L., Dwyer, M. A., Smith, J. J., and Hellinga, H. W. Computational design of receptor and sensor proteins with novel functions. *Nature 423* (2003), 185–190.

[10] Havranek, J. J., and Harbury, P. B. Automated design of specifity in molecular recognition. *Nat. Struct. Mol. Biol. 10* (2003), 45–52.

[11] Bolon, D., and Mayo, S. L. Enzyme-like proteins by computational design. *Proc. Natl. Acad. Sci. USA 98* (2001), 14274–14279.

[12] Dantas, G., Kuhlman, B., Callender, D., Wong, M., and Baker, D. A large test of computational protein design: Folding and stability of nine completely redesigned globular proteins. *J. Mol. Biol. 332* (2003), 449–460.

[13] Kuhlman, B., Dantas, G., Ireton, G., Varani, G., Stoddard, B., and Baker, D. Design of a novel globular protein fold with atomic-level accuracy. *Science 302* (2003), 1364–1368.

[14] Liang, H., Chen, H., Fan, K., Wei, P., Guo, X., Jin, C., Zeng, C., Tang, C., and Lai, L. H. De novo design of a $\beta\alpha\beta$ motif. *Ang. Chemie Int. Ed. 48* (2009), 3301–3303.

[15] Koga, N., Tatsumi-Koga, R., Liu, G., Xiao, R., Acton, T. B., Montelione, G. T., and Baker, D. Principles for designing ideal protein structures. *Nature 491* (2012), 222–224.

[16] Röthlisberger, D., Khersonsky, O., Wollacott, A. M., Jiang, L., DeChancie, J., Betker, J., Gallaher, J. L., Althoff, E. A., Zanghellini, A., Dym, O., Albeck, S., Houk, K. N., Tawfik, D. S., and Baker, D. Kemp elimination catalysts by computational enzyme design. *Nature 453* (2008), 190–195.

[17] Jiang, L., Althoff, E. A., Clemente, F. R., Doyle, L., Röthlisberger, D., Zanghellini, A., Gallaher, J. L., Betker, J. L., Tanaka, F., Barbas III, C. F., Hilvert, D., Houk, K. N., Stoddard, B. L., and Baker, D. De novo computational design of retro-aldo enzymes. *Science 319* (2008), 1387–1391.

[18] Richter, F., Leaver-Kay, A., Khare, S. D., Bjelic, S., and Baker, D. De novo enzyme design using Rosetta3. *PLoS One 6* (2011), e19230.

[19] Saven, J. G. Computational protein design: Advances in the design and redesign of biomolecular nanostructures. *Curr. Opin. Colloid Interf. Sci. 15* (2010), 13–17.

[20] Fortenberry, C., Bowman, E. A., Proffitt, W., Dorr, B., Combs, S., Harp, J., Mizoue, L., and Meiler, J. Exploring symmetry as an avenue to the computational design of large protein domains. *J. Am. Chem. Soc. 133* (2011), 18026–18029.

[21] Grigoryan, G., Kim, Y. H., Acharya, R., Axelrod, K., Jain, R. M., Willis, L., Dmdic, M., Kikkawa, J. M., and DeGrado, W. F. Computational design of virus-like protein assemblies on carbon nanotube surfaces. *Science 332* (2011), 1071–1076.

[22] King, N. P., Scheffer, W., Sawaya, M. R., Vollmar, B. S., Sumida, J. P., Andre, I., Gonen, T., Yeates, T. O., and Baker, D. Computational design of self-assembling protein nanomaterials with atomic level accuracy. *Science 336* (2012), 1171–1174.

[23] Lanci, C. J., MacDermaid, C. M., Kang, S. G., Acharya, R., North, B., Yang, X., Qiu, X. J., DeGrado, W. F., and Saven, J. G. Computational design of a protein crystal. *Proc. Natl. Acad. Sci. USA 109* (2012), 7304–7309.

[24] Mackerell, A. D., Bashford, D., Bellott, M., Dunbrack, R. L., Evanseck, J., Field, M. J., Fischer, S., Gao, J., Guo, H., Ha, S., Joseph, D., Kuchnir, L., Kuczera, K., Lau, F. T. K., Mattos, C., Michnick, S., Ngo, T., Nguyen,

D. T., Prodhom, B., Reiher, W. E., Roux, B., Smith, J., Stote, R., Straub, J., Watanabe, M., Wiorkiewicz-Kuczera, J., Yin, D., and Karplus, M. An all-atom empirical potential for molecular modelling and dynamics study of proteins. *J. Phys. Chem. B 102* (1998), 3586–3616.

[25] Case, D. A., Pearlman, D., Caldwell, J., III, T. C., Ross, W., Simmerling, C., Darden, T., Merz, K., Stanton, R., Cheng, A., Vincent, J., Crowley, M., Tsui, V., Radmer, R., Duan, Y., Pitera, J., Massova, I., Seibel, G., Singh, U., Weiner, P., and Kollman, P. *AMBER 6.* (University of California, San Francisco), 1999.

[26] Ponder, J., and Case, D. A. Force fields for protein simulations. *Adv. Prot. Chem. 66* (2003), 27.

[27] Jorgensen, W. L., and Tirado-Rives, J. Potential energy functions for atomic-level simulations of water and organic and biomolecular systems. *Proc. Natl. Acad. Sci. USA 102* (2005), 6665–6670.

[28] Brooks, B., Brooks III, C. L., Mackerell Jr., A. D., Nilsson, L., Petrella, R. J., Roux, B., Won, Y., Archontis, G., Bartels, C., Boresch, S., Caflisch, A., Caves, L., Cui, Q., Dinner, A. R., Feig, M., Fischer, S., Gao, J., Hodoscek, M., Im, W., Kuczera, K., Lazaridis, T., Ma, J., Ovchinnikov, V., Paci, E., Pastor, R. W., Post, C. B., Pu, J. Z., Schaefer, M., Tidor, B., Venable, R. M., Woodcock, H. L., Wu, X., Yang, W., York, D. M., and Karplus, M. CHARMM: The biomolecular simulation program. *J. Comp. Chem. 30* (2009), 1545–1614.

[29] Shaw, D. E., Maragakis, P., Lindorff-Larsen, K., Piana, S., Dror, R. O., Eastwood, M. P., Bank, J. A., Jumper, J. M., Salmon, J. K., Shan, Y., and Wriggers, W. Atomic-level characterization of the structural dynamics of proteins. *Science 330* (2010), 341–346.

[30] Deng, Y., and Roux, B. Computations of standard binding free energies with molecular dynamics simulations. *J. Phys. Chem. B 113* (2009), 2234–2246.

[31] Aleksandrov, A., Thompson, D., and Simonson, T. Alchemical free energy simulations for biological complexes: powerful but temperamental... *J. Molec. Recog. 23* (2010), 117–127.

[32] Wereszczynski, J., and McCammon, J. A. Statistical mechanics and molecular dynamics in evaluating thermodynamic properties of biomolecular recognition. *Quart. Rev. Biophys. 45* (2012), 1–25.

[33] Simonson, T., Archontis, G., and Karplus, M. Free energy simulations come of age: the protein–ligand recognition problem. *Acc. Chem. Res. 35* (2002), 430–437.

[34] Roux, B., and Simonson, T. Implicit solvent models. *Biophys. Chem. 78* (1999), 1–20.

[35] Simonson, T. Electrostatics and dynamics of proteins. *Rep. Prog. Phys. 66* (2003), 737–787.

[36] Baker, N. A. Poisson-Boltzmann methods for biomolecular electrostatics. *Methods Enzym. 383* (2004), 94.

[37] Feig, M., and Brooks III, C. L. Recent advances in the development and application of implicit solvent models in biomolecule simulations. *Curr. Opin. Struct. Biol. 14* (2004), 217–224.

[38] Vizcarra, C. L., Zhang, N. G., Marshall, S. A., Wingreen, N. S., Zeng, C., and Mayo, S. L. An improved pairwise decomposable finite-difference Poisson-Boltzmann method for computational protein design. *J. Comp. Chem. 29* (2008), 1153–1162.

[39] Liang, S., and Grishin, N. V. Effective scoring function for protein sequence design. *Proteins 54* (2004), 271–281.

[40] Guérois, R., Nielsen, J. E., and Serrano, L. Predicting changes in the stability of proteins and protein complexes: a study of more than 1000 mutations. *J. Mol. Biol. 320* (2002), 369–387.

[41] Kortemme, T., Morozov, A., and Baker, D. An orientation-dependent hydrogen bonding potential improves prediction of specificity and structure for proteins and proteinprotein complexes. *J. Mol. Biol. 326* (2003), 1239–1259.

[42] Pokola, N., and Handel, T. M. Energy functions for protein design: Adjustement with protein-protein complex affinities, models for the unfolded state, and negative design of solubility and specificity. *J. Mol. Biol. 347* (2005), 203–227.

[43] Smadbeck, J., Peterson, M. B., Khoury, G. A., Taylor, M. S., and Floudas, C. A. Protein WISDOM: A Workbench for In Silico de novo Design Of bioMolecules. *J. Visual. Exp. in press* (2013), 0000.

[44] Hellinga, H., and Richards, F. Optimal sequence selection in proteins of known structure by simulated evolution. *Proc. Natl. Acad. Sci. USA 91* (1994), 5803–5807.

[45] Wisz, M. S., and Hellinga, H. An empirical model for electrostatic interactions in proteins incorporating multiple geometry-dependent dielectric constants. *Proteins 51* (2003), 360–377.

[46] Koehl, P., and Levitt, M. De novo protein design. I. In search of stability and specificity. *J. Mol. Biol. 293* (1999), 1161–1181.

[47] Zou, J., and Saven, J. G. Using self-consistent fields to bias Monte Carlo methods with applications to designing and sampling protein sequences. *J. Chem. Phys. 118* (2003), 3843–3854.

[48] Ventura, S., and Serrano, L. Designing proteins inside out. *Proteins 56* (2004), 1–10.

[49] Chowdry, A. B., Reynolds, K. A., Hanes, M. S., Voorhies, M., Pokala, N., and Handel, T. M. An object-oriented library for computational protein design. *J. Comp. Chem. 28* (2007), 2378–2388.

[50] Lippow, S. M., Wittrup, K. D., and Tidor, B. Computational design of antibody affinity improvement beyond in vitro maturation. *Nature Biotech. 25* (2007), 1171–1176.

[51] Jaramillo, A., Wernisch, L., Héry, S., and Wodak, S. Folding free energy function selects native-like protein sequences in the core but not on the surface. *Proc. Natl. Acad. Sci. USA 99* (2002), 13554–13559.

[52] Suarez, M., and Jaramillo, A. Challenges in the computational design of proteins. *J. Royal Soc. Interface 6* (2009), S477–S491.

[53] Gainza, P., Roberts, K. E., Georgiev, I., Lilien, R. H., Keedy, D. A., Chen, C., Reza, F., Anderson, A. C., Richardson, D. C., Richardson, J. S., , and Donald., B. R. OSPREY: Protein design with ensembles, flexibility, and provable algorithms. *Methods Enzym. 523* (2013), 87–107.

[54] Schmidt am Busch, M., Lopes, A., Mignon, D., and Simonson, T. Computational protein design: software implementation, parameter optimization, and performance of a simple model. *J. Comp. Chem. 29* (2008), 1092–1102.

[55] Schmidt am Busch, M., Lopes, A., Amara, N., Bathelt, C., and Simonson, T. Testing the Coulomb/Accessible Surface Area solvent model for protein stability, ligand binding, and protein design. *BMC Bioinformatics 9* (2008), 148–163.

[56] Schmidt am Busch, M., Lopes, A., Mignon, D., Gaillard, T., and Simonson, T. The inverse protein folding problem: Protein design and structure prediction in the genomic era. In *Quantum Simulations of Materials and Biological Systems* (2012), J. Zeng, R. Q. Zhang, and H. Treutlein, Eds., Springer Science, Dordrecht, pp. 121–140.

[57] Brünger, A. T. X-plor *version 3.1, A System for X-ray crystallography and NMR.* Yale University Press, New Haven, 1992.

[58] Brünger, A. T., Adams, P. D., DeLano, W. L., Gros, P., Grosse-Kunstleve, R. W., Jiang, J., Pannu, N. S., Read, R. J., Rice, L. M., and Simonson, T. The structure determination language of the Crystallography and NMR System. In *International Tables for Crystallography, Volume F*, M. Rossmann and E. Arnold, Eds. Dordrecht: Kluwer Academic Publishers, the Netherlands, 2001, pp. 710–720.

[59] Dahiyat, B. I., and Mayo, S. L. De novo protein design: fully automated sequence selection. *Science 278* (1997), 82–87.

[60] Bashford, D., and Case, D. Generalized Born models of macromolecular solvation effects. *Ann. Rev. Phys. Chem. 51* (2000), 129–152.

[61] Moulinier, L., Case, D. A., and Simonson, T. Xray structure refinement of proteins with the generalized Born solvent model. *Acta Cryst. D 59* (2003), 2094–2103.

[62] Lopes, A., Aleksandrov, A., Bathelt, C., Archontis, G., and Simonson, T. Computational sidechain placement and protein mutagenesis with implicit solvent models. *Proteins 67* (2007), 853–867.

[63] Smith, C. A., and Kortemme, T. Backrub-like backbone simulation recapitulates natural protein conformational variability and improves mutant sidechain prediction. *J. Mol. Biol. 380* (2008), 742–756.

[64] Mandell, and Kortemme, T. Backbone flexibility in computational protein design. *Curr. Opin. Biotech. 20* (2009), 420–428.

[65] Huang, P. S., Ban, Y. E., Richter, F., Andre, I., Vernon, R., Schief, W. R., and Baker, D. RosettaRemodel: a generalized framework for flexible backbone protein design. *PLoS One 6* (2011), e24109.

[66] Schmidt am Busch, M., Mignon, D., and Simonson, T. Computational protein design as a tool for fold recognition. *Proteins 77* (2009), 139–158.

[67] Schmidt am Busch, M., Sedano, A., and Simonson, T. Computational protein design: validation and possible relevance as a tool for homology searching and fold recognition. *PLoS One 5(5)* (2010), e10410.

[68] Anderson, D. P. BOINC: A system for public-resource computing and storage. In *5th IEEE/ACM International Workshop on Grid Computing* (2004), IEEE Computer Society Press, USA.

[69] Wilson, D., Madera, M., Vogel, C., Chothia, C., and Gough, J. The SUPERFAMILY database in 2007: families and functions. *Nucl. Acids Res. 35* (2007), D308–D313.

[70] Lopes, A., Schmidt am Busch, M., and Simonson, T. Computational design of protein-ligand binding: Modifying the specificity of asparaginyl-tRNA synthetase. *J. Comp. Chem. 31* (2010), 1273–1286.

[71] Polydorides, S., Amara, N., Simonson, T., and Archontis, G. Computational protein design with a generalized Born solvent model: application to asparaginyl-tRNA synthetase. *Proteins 79* (2011), 3448–3468.

[72] Polydorides, S., and Simonson, T. Monte carlo simulations of proteins at constant ph with generalized born solvent. *J. Phys. Chem. B in press* (2013), 0000.

[73] Li, H., Robertson, A. D., and Jensen, J. H. Very fast empirical prediction and interpretation of protein pK_a values. *Proteins 61* (2005), 704–721.

[74] Olsson, M. H. M., Sondergaard, C. R., Rostowski, M., and Jensen, J. H. PROPKA3: consistent treatment of internal and surface residues in empirical pK_a predictions. *J. Chem. Theory Comp. 7* (2011), 525–537.

[75] Krivov, G. G., Shapalov, M. V., and Dunbrack, R. L. Improved prediction of protein side-chain conformations with scwrl4. *Proteins 77* (2009), 778–795.

[76] Street, A. G., and Mayo, S. Pairwise calculation of protein solvent-accessible surface areas. *Folding and Design 3* (1998), 253–258.

[77] Baker, D. A surprising simplicity to protein folding. *Nature 405* (2000), 39–42.

[78] Brooks, B., Bruccoleri, R., Olafson, B., States, D., Swaminathan, S., and Karplus, M. Charmm: a program for macromolecular energy, minimization, and molecular dynamics calculations. *J. Comp. Chem. 4* (1983), 187–217.

[79] Cornell, W., Cieplak, P., Bayly, C., Gould, I., Merz, K., Ferguson, D., Spellmeyer, D., Fox, T., Caldwell, J., and Kollman, P. A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. *J. Am. Chem. Soc. 117* (1995), 5179–5197.

[80] Jorgensen, W., and Tirado-Rives, J. The OPLS potential function for proteins, energy minimization for crystals of cyclic peptides and crambin . *J. Am. Chem. Soc. 110* (1988), 1657–1666.

[81] Simonson, T. Electrostatic free energy calculations for macromolecules: a hybrid molecular dynamics/continuum electrostatics approach. *J. Phys. Chem. B 104* (2000), 6509–6513.

[82] Schaefer, M., and Karplus, M. A comprehensive analytical treatment of continuum electrostatics. *J. Phys. Chem. 100* (1996), 1578–1599.

[83] Calimet, N., Schaefer, M., and Simonson, T. Protein molecular dynamics with the Generalized Born/ACE solvent model. *Proteins 45* (2001), 144–158.

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

[85] Pokala, N., and Handel, T. Energy functions for protein design I: Efficient and accurate continuum electrostatics and solvation. *Prot. Sci. 13* (2004), 925–936.

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

[87] Barth, P., Alber, T., and Harbury, P. B. Accurate, conformation-dependent predictions of solvent effects on protein ionization constants. *Proc. Natl. Acad. Sci. USA 104* (2007), 4898–4903.

[88] Lee, B., and Richards, F. The interpretation of protein structures: estimation of static accessibility. *J. Mol. Biol. 55* (1971), 379–400.

[89] Wernisch, L., Héry, S., and Wodak, S. Automatic protein design with all atom force fields by exact and heuristic optimization. *J. Mol. Biol. 301* (2000), 713–736.

[90] Koehl, P., and Delarue, M. Application of a self-consistent mean field theory to predict protein sidechain conformations and estimate their conformational entropy. *J. Mol. Biol. 239* (1994), 249–275.

[91] Saven, J. G., and Wolynes, P. G. Statistical mechanics of the combinatorial synthesis and analysis of folding macromolecules. *J. Phys. Chem. B 101* (1997), 8375–8389.

[92] Zou, B. J., and Saven, J. G. Statistical theory for protein ensembles with designed energy landscapes. *J. Chem. Phys. 123* (2005), 154908.

[93] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. Equation of state calculations by fast computing machines. *J. Chem. Phys. 21* (1953), 1087–1092.

[94] Frenkel, D., and Smit, B. *Understanding molecular simulation.* Academic Press, New York, 1996.

[95] Hill, T. *Introduction to Statistical Thermodynamics.* Addison-Wesley, Reading, Massachussetts, 1962.

[96] Baptista, A. M., Martel, P. J., and Petersen, S. B. Simulation of protein conformational freedom as a function of pH: constant-pH molecular dynamics using implicit titration. *Proteins 27* (1997), 523–544.

[97] Börjesson, U., and Hünenberger, P. H. Explicit-solvent molecular dynamics simulation at constant pH: Methodology and application to small amines. *J. Chem. Phys. 114* (2001), 9706–9719.

[98] Lee, M. S., Salsbury Jr., F. R., and Brooks III, C. L. Constant pH molecular dynamics using continuous titration coordinates. *Proteins 56* (2004), 738–752.

[99] Mongan, J., Case, D. A., and McCammon, J. A. Constant pH molecular dynamics in generalized Born implicit solvent. *J. Comp. Chem. 25* (2004), 2038–2048.

[100] Georgescu, E. R., Alexov, E., and Gunner, M. Combining conformational flexibility and continuum electrostatics for calculating pK_a's in proteins. *Biophys. J. 83* (2002), 1731–1748.

[101] Dahiyat, B. I., and Mayo, S. L. Protein design automation. *Prot. Sci. 5* (1996), 895–903.

[102] Brooks, B., Brooks III, C. L., Mackerell Jr., A. D., Nilsson, L., Petrella, R. J., Roux, B., Won, Y., Archontis, G., Bartels, C., Boresch, S., Caflisch, A., Caves, L., Cui, Q., Dinner, A. R., Feig, M., Fischer, S., Gao, J., Hodoscek, M., Im, W., Kuczera, K., Lazaridis, T., Ma, J., Ovchinnikov, V., Paci, E., Pastor, R. W., Post, C. B., Pu, J. Z., Schaefer, M., Tidor, B., Venable, R. M., Woodcock, H. L., Wu, X., Yang, W., York, D. M., , and Karplus, M. CHARMM: The biomolecular simulation program. *J. Comp. Chem. 30* (2009), 1545–1614.

[103] Brünger, A., Adams, P., Clore, G., Delano, W., Gros, P., Grosse-Kunstleve, R., Jiang, J., Kuszewski, J., Nilges, M., Pannu, N., Read, R., Rice, L., Simonson, T., and Warren, G. Crystallography and NMR System: a new software suite for macromolecular structure determination. *Acta Cryst. D54* (1998), 905–921.

[104] Schweiters, C., Kuszewski, J., Tjandra, N., and Clore, G. The Xplor-NIH molecular structure determination package. *J. Biomol. NMR 160* (2003), 65–73.

[105] Onufriev, A., Case, D. A., and Ullmann, M. A novel view of pH titration in biomolecules. *Biochemistry 40* (2001), 3413–3419.

[106] Wang, L., Brock, A., Herberich, B., and Schultz, P. G. Expanding the genetic code of *escherichia coli*. *Science 292* (2001), 498–500.

[107] Young, T. S., and Schultz, P. G. Beyond the canonical twenty amino acids: expanding the genetic lexicon. *J. Biol. Chem. 285* (2010), 11039–11044.

[108] Zhang, Y., Wang, L., Schultz, P. G., and Wilson, I. A. Crystal structures of apo wild-type *m. jannaschi* tyrosyl-tRNA synthetase (TyrRS) and an engineered TyrRS specific for O-methyl-L-tyrosine. *Prot. Sci. 14* (2005), 1340–1349.

[109] Tsunoda, M., Kusakabe, Y., Tanaka, N., Ohno, S., Nakamura, M., Senda, T., Moriguchi, T., Asai, N., Sekine, M., Yokogawa, T., Nishikawa, K., and Nakamura, K. T. Structural basis for recognition of cognate tRNA by tyrosyl-tRNA synthetase from three kingdoms. *Nucl. Acids Res. 35* (2007), 4289–4300.

[110] Tuffery, P., Etchebest, C., Hazout, S., and Lavery, R. A new approach to the rapid determination of protein side chain conformations. *J. Biomol. Struct. Dyn. 8* (1991), 1267.

[111] Archontis, G., Simonson, T., and Karplus, M. Binding free energies and free energy components from molecular dynamics and Poisson-Boltzmann calculations. Application to amino acid recognition by aspartyl-tRNA synthetase. *J. Mol. Biol. 306* (2001), 307–327.

[112] Alexov, E., Mehler, E. L., Baker, N., Baptista, A. M., Huang, Y., Milletti, F., Nielsen, J. E., Farrell, D., Carstensen, T., Olsson, M. H. M., Shen, J. K., Warwicker, J., Williams, S., and Word, J. M. Progress in the prediction of pK_a values in proteins. *Proteins 79* (2011), 3260–3275.

[113] Pawson, T. Protein modules and signalling networks. *Nature 373* (1995), 573–580.

[114] Broutin, I., and Ducruix, A. Structural domains and signaling networks. *M'edecine Sci. 16* (2000), 611–616.

[115] Wu, X., Knudsen, B., Feller, S. M., Zheng, J., Sali, A., Cowburn, D., Hanafusa, H., and Kuriyan, J. Structural basis for the specific interaction of lysine-containing proline-rich peptides with the N-terminal SH3 domain of c-Crk. *Structure 3* (1995), 215.

[116] Finn, R. D., Mistry, J., Schuster-Bckler, B., Griffiths-Jones, S., Hollich, V., Lassmann, T., Moxon, S., Marshall, M., Khanna, A., Durbin, R., Eddy, S. R., Sonnhammer, E. L. L., and Bateman, A. Pfam: clans, web tools and services. *Nucl. Acids Res. 34* (2006), D247–251.

[117] Andreeva, A., Howorth, D., Brenner, S. E., Hubbard, J. J., Chothia, C., and Murzin, A. G. SCOP database in 2004: refinements integrate structure and sequence family data. *Nucl. Acids Res. 32* (2004), D226–229.

[118] Bashford, D., and Karplus, M. The pK_a's of ionizable groups in proteins: atomic detail from a continuum electrostatic model. *Biochemistry 29* (1990), 10219–10225.

[119] Simonson, T., Carlsson, J., and Case, D. A. Proton binding to proteins: pK_a calculations with explicit and implicit solvent models. *J. Am. Chem. Soc. 126* (2004), 4167–4180.

[120] Castaneda, C. A., Fitch, C. A., Majumdar, A., Khangulov, V., Schlessman, J. L., and Garcia-Moreno, B. Molecular determinants of the pKa values of Asp and Glu residues in Staphylococcal nuclease. *Proteins 77* (2009), 570–588.

[121] Bas, D. C., Rogers, D. M., and Jensen, J. H. Very fast prediction and rationalization of pK_a values for protein-ligand complexes. *Proteins 73* (2008), 765–783.

[122] Kilambi, K., and Gray, J. J. Rapid calculation of protein pK_a values using Rosetta. *Biophys. J. 103* (2012), 587–595.

[123] Gunner, M., Zhu, X., and Klein, M. C. MCCE analysis of the pK_a's of introduced buried acids and bases in Staphylococcal nuclease. *Proteins 79* (2011), 3306–3319.

Table 1: Unfolded state energies E_X and tripeptide energies E_3

AA type	number[a]	E_3	E_X
Ala	27	6.21 (0.02)[b]	7.27
Arg	18	-17.79 (0.16)	-16.27
Asn	17	1.66 (0.08)	2.25
Asp	16	-5.94 (0.13)	-4.00
Cys	2	6.44 (0.22)	4.06
Gln	10	1.42 (0.23)	2.02
Glu	19	-4.93 (0.12)	-4.75
HID[c]	1	17.26 (0.02)	15.99
HIE[c]	4	18.59 (0.23)	15.77
HIP[c]	1	17.26 (0.02)	18.06
Ile	15	9.35 (0.07)	10.18
Leu	27	7.10 (0.06)	7.95
Lys	35	-2.26 (0.09)	-1.44
Met	9	5.90 (0.16)	2.93
Phe	8	4.09 (0.16)	4.42
Ser	9	4.26 (0.09)	3.64
Thr	19	4.53 (0.07)	4.71
Trp	4	1.37 (0.10)	1.22
Tyr	13	-2.93 (0.07)	-0.21
Val	18	6.91 (0.06)	7.33

The energies are also shown in Fig. 1. They are computed with the ff99SB force field, the GB/HCT solvent model, a protein dielectric constant of 16, and a surface energy term. The atomic surface energy coefficients are as follows (in kcal/mol/Å^2): alkane atoms = -0.005; polar atoms = -0.08; aromatic atoms = -0.04; ionic atoms = -0.10; hydrogens = 0. [a]Positions in SNase and lysozyme with the corresponding amino acid type. [b]Tripeptide energies are averaged over several positions in two proteins, and over 800 conformations (rotamer sets) for each protein (standard deviation in parentheses). [c]Two singly-protonated and a doubly-protonated isoform of His.

Table 2: Variants of *M. jann.* TyrRS designed for O-methyl-Tyr binding

Rotamer library	Amino acids 32, 158	(kcal/mol) Affinity[b]	[c]PBFE affinity
1995	AL	-96.0	3.1
1995	AA	-95.5	3.8
2003	QF	-94.3	5.1
1995	**QA**[a]	-96.6	**5.3**
1995	QL	-96.8	8.5
2003	QQ	-95.6	9.4

[a]This sequence was shown experimentally to be active [106]. [b]Affinity is averaged over the top 1000 conformations obtained from a conformation exploration for each sequence. [c]From PB binding free energy calculations using 400 snapshots from a 2 ns MD trajectory for each TyrRS variant (with explicit solvent).

1. **Unfolded state energies.** The E_X values were optimized for the SH3 protein family, using CPD runs with the Amber ff99SB force field, a GB solvent model, and a protein dielectric constant of 16. The tripeptide energies are computed with the same energy function, and averaged over several positions in two proteins (SNase and lysozyme) and over 800 conformations (rotamer sets) for each protein. The dispersion of the tripeptide energies is comparable to the size of the dots and is not visible.

2. **Titrating TyrRS with D-Tyr.** Each dot represents an MC simulation with a specific D-Tyr concentration. Lines are sigmoidal fits. Black: wildtype *E. coli* protein; grey: Asp81Arg mutant.

3. **Natural and designed SH3 sequences.** The "natural" sequence logo is for the Pfam SH3 alignment; the designed logo is for the top 10000 MC sequences (Crk numbering). Red dots highlight hydrophobic core positions, blue dots are ligand-binding positions, and crosses are positions that are poorly predicted by the design. Sidechains in the stereo structure, above, are colored the same way; the peptide ligand is yellow. In the alignment, below, "Pfam" is the consensus sequence from the Pfam SH3 alignment; "consensus" is the consensus over the top 25 designed sequences.

4. **MD simulations of Crk variants.** The rms deviation (Å) is for the backbone of 11 core residues, relative to the starting, X-ray structure. Mutant 0 is from an earlier design [66]. The 3D structures, above are the starting, X-ray structure (yellow) and MD snapshots taken every 2 ns between 10 and 20 ns; the darker colors are the later snapshots.

5. **SNase acid/base titration.** Titration curves are shown for four selected sidechains, as indicated. Each pH value was simulated for 20 million MC steps. Dots are populations from the MC; lines are sigmoidal fits. Experimental values are marked as x or o (D19, which has the largest error: computed = 4.7; experimental = 2.2).

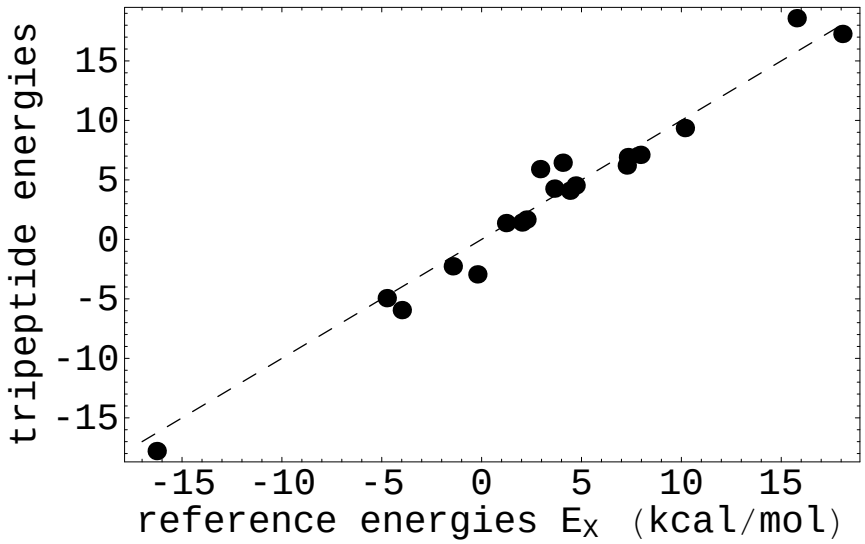

Figure 1: **Unfolded state energies.** The E_X values were optimized for the SH3 protein family, using CPD runs with the Amber ff99SB force field, a GB solvent model, and a protein dielectric constant of 16. The tripeptide energies are computed with the same energy function, and averaged over several positions in two proteins (SNase and lysozyme) and over 800 conformations (rotamer sets) for each protein. The dispersion of the tripeptide energies is comparable to the size of the dots and is not visible.

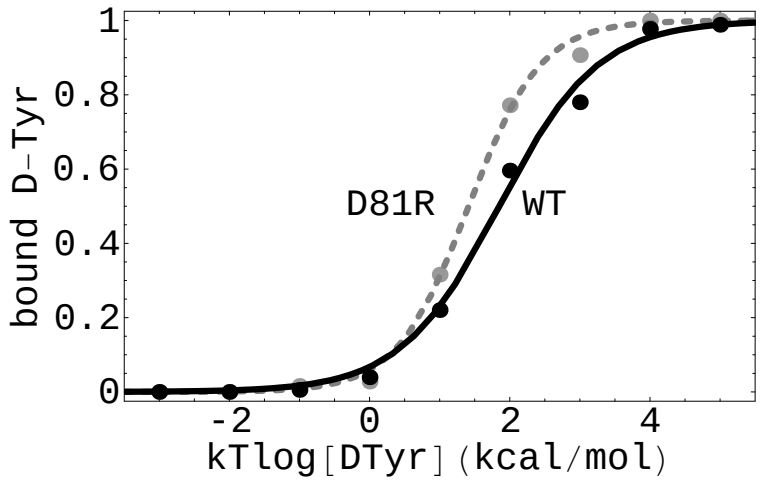

Figure 2: **Titrating TyrRS with D-Tyr.** Each dot represents an MC simulation with a specific D-Tyr concentration. Lines are sigmoidal fits. Black: wildtype *E. coli* protein; grey: Asp81Arg mutant.

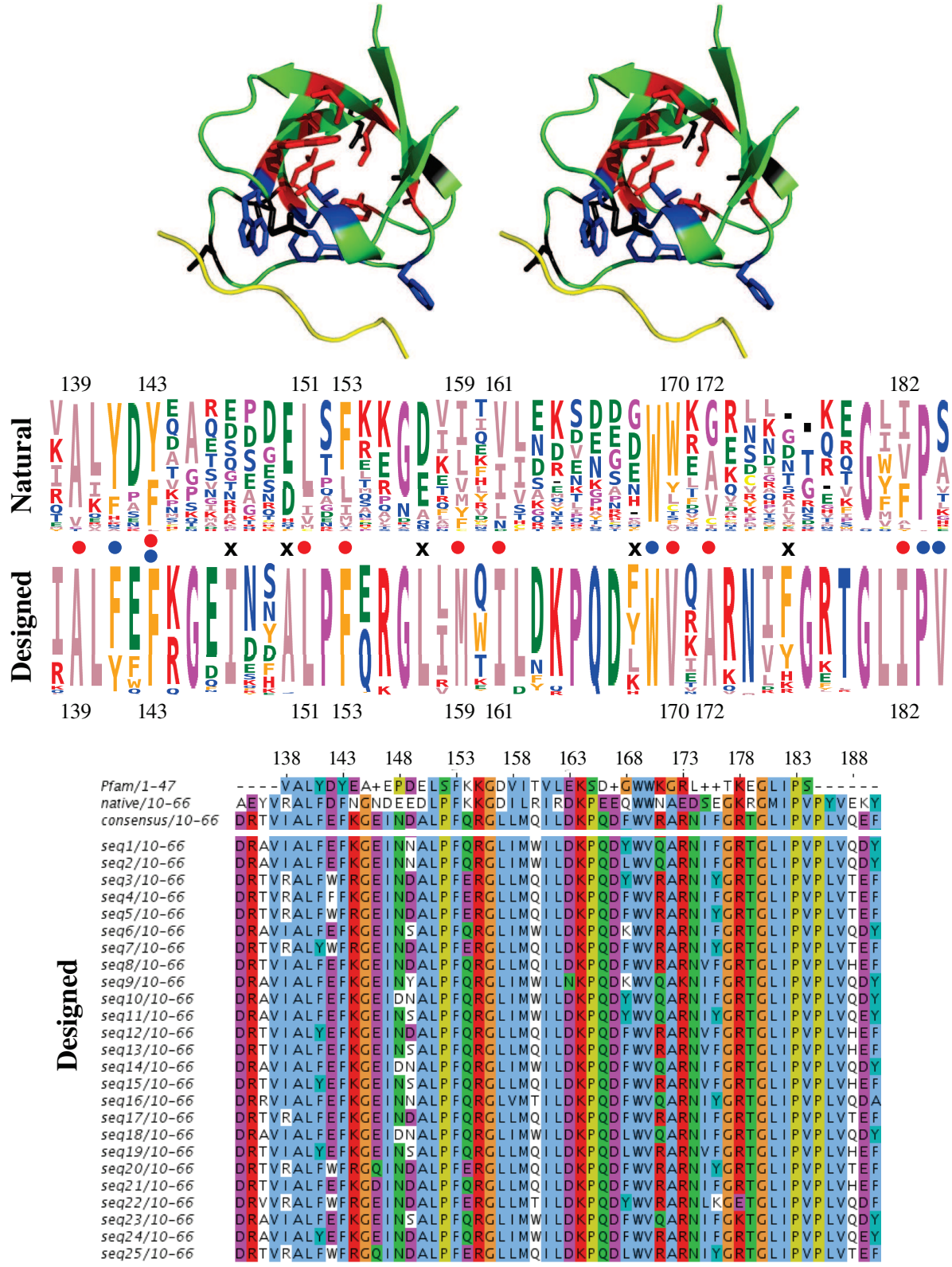

Figure 3: **Natural and designed SH3 sequences.** The "natural" sequence logo is for the Pfam SH3 alignment; the designed logo is for the top 10000 MC sequences (Crk numbering). Red dots highlight hydrophobic core positions, blue dots are ligand-binding positions, and crosses are positions that are poorly predicted by the design. Sidechains in the stereo structure, above, are colored the same way; the peptide ligand is yellow. In the alignment, below, "Pfam" is the consensus sequence from the Pfam SH3 alignment; "consensus" is the consensus over the top 25 designed sequences.

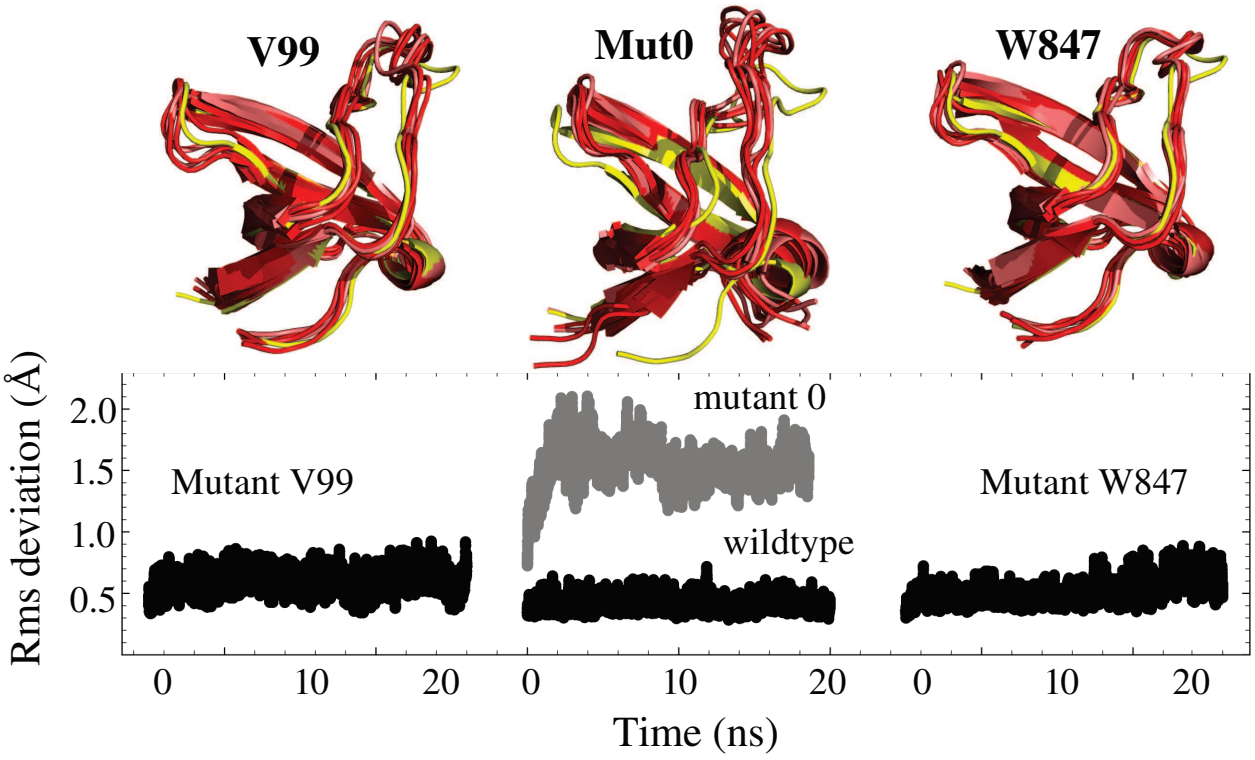

Figure 4: **MD simulations of Crk variants.** The rms deviation (Å) is for the backbone of 11 core residues, relative to the starting, X-ray structure. Mutant 0 is from an earlier design [66]. The 3D structures, above are the starting, X-ray structure (yellow) and MD snapshots taken every 2 ns between 10 and 20 ns; the darker colors are the later snapshots.

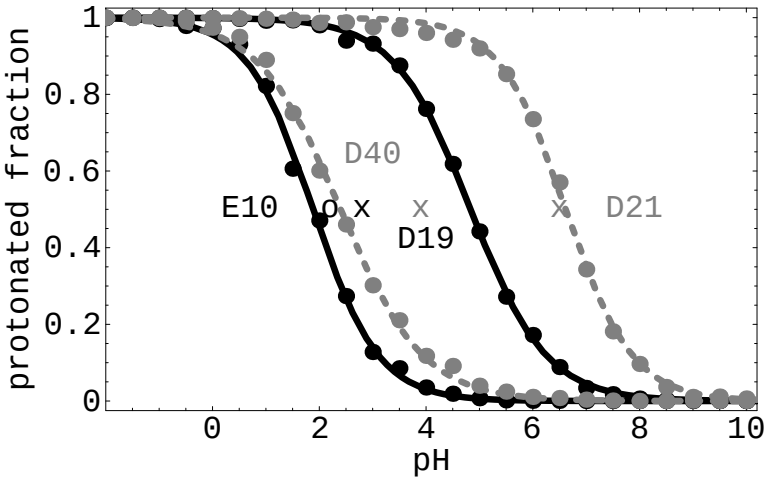

Figure 5: **SNase acid/base titration.** Titration curves are shown for four selected sidechains, as indicated. Each pH value was simulated for 20 million MC steps. Dots are populations from the MC; lines are sigmoidal fits. Experimental values are marked as x or o (D19, which has the largest error: computed = 4.7; experimental = 2.2).