For: *Distributed & Grid Computing – Science Made Transparent for Everyone. Principles, Applications and Supporting Communities*, Tektum Publishers, Berlin.

The inverse protein folding problem: structure prediction in the genomic era

Thomas Simonson, David Mignon, Marcel Schmidt am Busch, Anne Lopes and Christine Bathelt

Laboratoire de Biochimie (UMR CNRS 7654), Department of Biology, Ecole Polytechnique, 91128, Palaiseau, France.

1 Introduction: structure prediction on a genomic scale

Over the past decade, the genomes of about 1000 organisms have been entirely "sequenced": the exact nucleotide sequence of the DNA that makes up their chromosome(s) has been experimentally determined [1]. These nucleotides (millions to billions, depending on the complexity of the organism) encode all the molecules the cells need to produce, including their full complement of proteins. Proteins are the essential actors of the living cell: biochemical catalysts, motors, pumps, reading and interpreting the genetic message, directing the response to external signals or attacks. Humans, for example have a genome of about 3.4 billion nucleotides, including about 25,000 genes that code for proteins [2, 3]. A protein is a polymer, or chain of amino acids, with a length usually between 100 and 1000 units. The amino acids are drawn from a small, natural "library" of 20 compounds (Fig. 1) [4]. The protein amino acid sequences can be deduced directly from the gene sequence that encodes them. The mapping that connects the nucleotide sequence and the amino acid sequence is known as the genetic code.

The challenge today is to determine the structure and biological function of all the known proteins [5–7]. Indeed, although the amino acid sequences of millions of proteins have been determined, most of their three-dimensional molecular structures are unknown. Yet the knowledge of these structures is essential to identify, understand, and possibly engineer or modify their biological functions. Predicting the three-dimensional structure from the amino acid sequence is the classic, "Protein Folding Problem", one of the most important problems in molecular biology today.

In the cell, the amino acid sequence of a protein uniquely directs it to "fold" into a specific, three-dimensional, molecular structure (Fig. 2). In effect, the amino acid chain has a unique, preferred, three-dimensional arrangement, which corresponds to its lowest possible free energy [4]. It also has the ability to rapidly explore the available

conformational space to find this preferred structure. The preferred structure is known as the "native" structure. The ability to fold rapidly into a unique, native structure is an essential and universal property of natural proteins. In contrast, a random, artificially constructed polymer of amino acids will almost certainly not "fold"; *i.e.*, it will not adopt a unique structure, but will divide its time among a large number of structures of comparable stability. Or worse: the sample will form an aggregate and precipitate in the form of a powder at the bottom of the test tube. Thus, over the course of millions of years of evolution, chance and natural selection have shaped modern protein sequences, building up a large, but very specific repertoire of viable sequences, capable of folding and performing a useful biological function.

Protein structure prediction (with the high precision needed to understand its function) is difficult for two main reasons. First, a protein has many degrees of freedom and an almost unlimited set of possible conformations. Considering amino acids alone or within very small polypeptides, one finds that each one can occupy about ten different conformations. For a small protein of 100 amino acids, then, there are on the order of 10^{100} available conformations—one googol. The second difficulty comes from the weak stability of the folded structure. To denature, or "unfold"' a protein, only about 10 kcal/mol are usually required. For a small protein of 1000 atoms, this represents 0.01 kcal/mol per atom. This energy can be compared to the average kinetic energy of each atom at room temperature: about 1 kcal/mol. Thus, the most stable, native structure is only separated from non-native structures by a very small free energy difference. Fortunately, there is often limited information available about the native structure, which can lead to a useful prediction, despite these diffculties. For example, the amino acid sequence may be similar to the sequence of another protein, whose 3D structure is already known. Proteins with similar sequences are said to be "homologous", and structure prediction in this case reduces to a "homology modelling" problem [8]. This type of prediction is carried out by the Predictor@Home distributed computing project, described elsewhere in this book (and at predictor.scripps.edu).

At the other end of the prediction spectrum, is a brute force, ab initio approach: one simulates the molecular motions of the polypeptide chain in solution and simply waits for it to spontaneously fold up into its native structure. In effect, one tries to reproduce in the computer the biological folding reaction as it occurs in the cell (or at least the test tube). This very ambitious approach has been carried out for several years by the pioneering Folding@Home distributed computing project [9], also described in this book. It provides not only a structure prediction, but a detailed physical picture of the folding reaction.

In this chapter, we describe a third approach. Instead of searching for the optimal conformation, or fold for a given amino acid sequence, we consider the inverse problem. For a given fold, we search for the best amino acid sequences [10, 11]. With the Protein Folding Problem, we needed to search a vast conformational space. With the present,

"Inverse Folding Problem", we need to search a completely different space: the space of amino acid sequences (of a given length). This brings us back to the "genomic space" from which we started out.

We are solving the inverse folding problem for a representative subset of all known protein structures. This subset includes 1500—2000 structures of protein "domains", collected in the "Structural Classification of Proteins" (SCOP) database [12]. A protein "domain" is a structural unit, made of 50–300 amino acids, which is either a small protein, or a part of a larger protein. Domains represent an intermediate level of structural organization, since larger proteins are invariably built up from several distinct domains, and a protein domain can often fold into its specific structure by itself, even if it is removed from the rest of the protein to which it belongs [12, 13]. Historically, most domains correspond to small, ancient proteins that, over time, have become fused with other domains to form the larger, modern, multi-domain proteins [14]. By breaking proteins down into their underlying domains, we make the inverse folding problem tractable.

What does this have to do with structure prediction? In fact, we want to solve a "fold recognition" problem (Figure 3). For each domain, we consider several million possible sequences, and identify the most favorable. These provide a "signature" of the 3D domain structure, or fold. Indeed, if we consider now a new protein sequence, for which the 3D structure is unknown, we can compare it to our database of computed sequences. If the new sequence is similar to one or more in our database, we can infer that it will adopt the same 3D structure. In effect, we have identified the fold of the new sequence, and this is the first step towards structure prediction by homology modelling (above) [6].

In addition to structure prediction, another application of these techniques is the construction of new proteins, or "protein design". Among the sequences associated with a given protein domain, we can select those that are likely to perform a desired function, such as binding specifically to another protein, or catalyzing a particular chemical reaction. By selecting sequences that stabilize a given fold and, at the same time, are capable of performing a specific chemical or biological function, we perform molecular evolution in the computer. This technique for protein design is referred to as "Directed Evolution". Directed evolution has been successfully used in recent years to develop new biosensors, new catalysts, and to create completely new protein folds [15–19].

The most CPU-intensive steps for solving these problems are highly parallel calculations that are weakly coupled and ideally-suited for a distributed computing environment. Therefore, a key ressource for solving them is the community of internauts, with their unique ability to donate cycles from their computers to help advance science, technology, and medecine.

In the next sections, we describe our basic methodology for structure prediction and protein design, along with selected results obtained in the last few months. The first

ingredient of our method is a simplified, discrete description of the protein's conformational space when it is in the folded state. The second is a description of the unfolded state. The third is a classic, "molecular mechanics" decription of proteins, which allows us to calculate the energy of any given sequence in any given conformation [20]. The fourth ingredient is a divide-and-conquer technique, where the necessary energy data are precomputed for the fold of interest, taking into account all possible sequences and sidechain arrangements. The overall complexity of this step is only quadratic with respect to the number of amino acids in the protein. The fifth ingredient is a large, object-oriented software package that implements the molecular mechanics energy model and the divide-and-conquer algorithm [21, 22]. The sixth is an algorithm and software for efficiently exploring sequence space, searching for the best sequences [19, 23]. Finally, an essential, seventh ingredient is the Berkely Open Infrastructure for Network Computing, or BOINC, which is described by its developers elsewhere in this book [24]. This is the software platform that allows us to propose our project to the internaut community, and progress towards structure prediction and protein design on a genomic scale.

2 Protein structure: the importance of being discrete

A protein is a polymer of amino acids [4]. The amino acids are drawn from a natural library of 20 compounds. They share a common, backbone moiety, which is used to link them chemically in the polypeptide chain. The different amino acids are distinguished by their specific, sidechain moiety, which ranges in size and complexity from a single hydrogen atom up to groups like methyl-indole or butyl-guanidinium (Fig. 1). The backbone degrees of freedom give rise to the overall fold, while the sidechain degrees of freedom determine the local structure. In general, we are interested in the inverse folding problem: identifying the amino acids that stabilize a given protein fold. Therefore, we can assume the backbone degrees of freedom are fixed and concentrate on those of the sidechains [19, 23].

The complexity of the sidechain conformational space remains formidable. It can be reduced to a manageable level, however, thanks to the "rotamer" concept, introduced by Janin et al [25] and exploited by Jay Ponder and Frederic Richards in their pioneering structure prediction work [10]. The sidechain geometries in proteins can be defined by a few torsional angles, corresponding to rotations of chemical groups around single chemical bonds. These angles are named conventionally χ_1, χ_2, $\cdots$, going from the backbone out along the sidechain (see Fig. 1). To explore the corresponding conformational space, it is very convenient to perform discrete steps along each torsional degree of freedom. This is especially well-suited to proteins, since in practice, some values of the torsion angles are much more probable than others. In fact, in experimentally-determined protein structures, there are distinct preferences for a limited set of torsional values [10]. The corresponding conformations are known as "rotamers". Although

each amino acid type has typically 2–3 torsion angles, these adopt, on average, on the order of just ten preferred rotamers. Thus, using the discrete rotamer description has an enormous, simplifying effect on the protein's conformational space. For reviews of the preferred rotamers in protein structures, and for databases of preferred rotamers, see [10, 26–28].

3 The role of the unfolded state

Protein stability is determined by a competition between the native, folded structure and an ensemble of unfolded structures. Indeed, the unfolded structure is not unique. When the protein deviates from its folded structure, after an unusually violent collision with another molecule, for example, it will wander for a time among a large collection of less compact structures, before finding its way back to the most stable, native structure. In the language of thermodynamics, the existence of many different unfolded structures means that the unfolded state is stabilized by entropy. For a protein to function in the cell, it should spend much or most of its time in the folded state, since this is the 3D structure that is competent to perform the protein's function, be it catalysis of a biochemical reaction, transmission of a signal, or energy transduction. From thermodynamics, the time spent in the folded state increases exponentially as the folding free energy gets larger (more favorable). Therefore, to identify the most favorable sequences, we look for those that produce a large, favorable, free energy difference between the folded and the unfolded state.

Modelling the unfolded state is thus an essential ingredient of our structure prediction method. The structures of several thousand proteins in their folded state have been determined by Xray crystallography, as well as nuclear magnetic resonance and other techniques [29]. However, the unfolded state is very hard to characterize, because it is dynamic and poorly ordered. Therefore, following extensive earlier work, we adopt a very simple, empirical model of the unfolded state [19, 23]. We simply assume the unfolded polypeptide chain is largely extended, so that the amino acid sidechains interact primarily with solvent, and only weakly with each other. This general organization is assumed not to depend much on the amino acid sequence (although the competition between folded and unfolded states does). Therefore, backbone interactions in the unfolded state will largely cancel when different amino acid sequences are compared.

The advantage of this model is its simplicity. In practice, the unfolded state energy involves only short-ranged interactions between each amino acid side chain and neighboring backbone groups. It can be computed rapidly and separately from the folded state. The model can be improved empirically, by comparing the computed sequences with experimental sequences. For example, we add (below) empirical corrections that force the model to give the correct, experimental, amino acid composition for the protein class of interest. If the experimental database indicates an average of 5 tryptophans

for every 100 amino acids, we can retrieve this percentage by slightly increasing or decresing the contribution of tryptophans to the unfolded state energy. If tryptophans are too abundant in our computed sequences, we would make tryptophan "more stable" in the unfolded state, for example. Then tryptophans will contribute less favorably to the folded/unfolded competition, and will be found less frequently among the favored sequences. Such an empirical correction is expected to compensate for some of the systematic errors inherent in the simple unfolded structure model. Ultimately, however, the unfolded model can only be validated by detailed comparisons between computed and experimental sequences. Previous work in this field indicates that the quality of the model is reasonable [19, 23, 30, 31].

4 Relating structures and energies: a molecular mechanics description of proteins

The most important computational models in use today for proteins are based on a "molecular mechanics" description. They represent the protein as a collection of spherical particles (the atoms), approximately incompressible, connected together by springs, each one bearing a small electric charge [20, 32]. The charges provide a simple representation of the electropositive or electronegative character of the different chemical groups of atoms. The springs represent the covalent bonds between atoms. They serve to maintain the proper stereochemistry and the proper rigidity of chemical groups: carbons with tetrahedral or planar bonding arrangements, single or double chemical bonds, and so on. Solvent molecules can be described in the same way. To parameterize such a model for a large class of molecules like proteins takes several decades of researcher-years. Once in place, and desite its simplicity, a molecular mechanics model is a powerful tool to study the structure and stability of biomolecules.

The molecular mechanics description is based on simple physical concepts, well-known since Coulomb, Laplace, and Newton. We can thus write a mathematical expression for the potential energy of such a molecular "Lego". The potential energy in turn gives us the forces between atoms, which determine the molecular motions. Thus, in 1977, Martin Karplus and his collaborators produced the first computer simulation of the molecular dynamics of a small protein, over a short period of a few picoseconds, given the limited computing power available at the time [33]. The computer programs that implement such molecular mechanics models today are usually made up of around 100,000 lines of code. They have many applications, in addition to the folding and inverse folding problem, such as the study of protein–ligand interactions or enzyme reaction mechanisms.

A key element of the energy model is the description of the aqueous solvent in which a protein bathes. Indeed, the native structure is normally compact, or globular,

and the protein folding reaction tends to segregate the less polar amino acids in the core of the structure, and the more polar ones at the surface. The less polar amino acid sidechains are made of alkane groups, which do not form very favorable interactions with water: they are said to be "hydrophobic". This segregation reduces the alkane–water interface, and the globular structure is stabilized by a "hydrophobic effect". Thus aqueous solvent plays an active role in driving the protein into its native structure, and structure prediction must take this into account. Yet it is much too expensive, for the applications below, to explicitly model thousands of water molecules, whose detailed behavior is not of interest *per se*. Therefore, most structure prediction methods rely on simplified descriptions of the aqueous solvent. In these descriptions, the solvent appears "implicitly", through its effect on the protein–protein interactions.

A very simple implicit solvent model might assume, for example, that a polar medium like water will reduce the electrostatic interactions within the protein by a constant factor ϵ. This reduction corresponds to a dielectric shielding effect, and the reduction factor ϵ is the microscopic analogue of a dielectric constant. In fact, this simple model works surprisingly well. It is usually supplemented by a second ingredient. To capture the hydrophobic/hydrophilic character of the nonpolar/polar groups, we add to the energy function a term that depends on the solvent exposure of each atom, through it exposed surface area. Atoms in the core will have a zero exposed area; atoms at the surface will be only partly buried by their neighbors and will have a non-zero exposed area. For polar atoms, the area is multiplied by a negative coefficient; for nonpolar atoms, it is multiplied by a positive (or less negative) coefficient. Thus, exposing polar atoms improves (lowers) the energy; exposing alkane atoms increases the energy. The coefficients have the dimensions of a surface tension: energy per surface area. This combined solvent model will be referred to as the "Coulomb/Accessible Surface Area" model, or CASA [34]. For a review of this and more sophisticated implicit solvent models, see [35].

An essential property of the energy model just described is *pairwise additivity*: the energy takes the form of a sum of pairwise interactions between atoms or groups. It can be written:

$$U(r_1, r_2, \cdots, r_N) = \sum_i \sum_j U_{ij}(r_j, r_j), \tag{1}$$

where i, j represent individual amino acids, r_i is a vector that specifies the spatial positions of all the atoms of amino acid i, and N is the number of amino acids in the protein (its length). Although the total energy U (left) depends on the the positions of all N amino acids, it can be broken down into just N^2 terms (right), each of which has just "pairwise complexity", depending on just two amino acids i, j. This property makes possible the divide-and-conquer method described in the next section.

5 A divide-and-conquer method for protein design

The divide-and-conquer approach described here was introduced by Mayo and coworkers [15]. It relies on two simplifications in the protein description: the pairwise energy function, explained in section 4, and the simplified, discrete description of the protein conformational space, explained in section 2.

With the protein backbone fixed and the rotamer approximation for the sidechains, we have just a few degrees of freedom for each amino acid. To find the optimal amino acid sequence and structure, we must consider just 20 amino acid types and $\sim$10 possible rotamers at each position. However, the size of the problem grows exponentially with the length of the protein chain, leading to a combinatorial explosion. For a small protein of 100 amino acids, for example, we have around $100^{10} = 10^{20}$ structures for a single amino acid sequence. Considering all possible sequences (and a typical rotamer set), there are over 10^{400} structures: googols of googols!

Fortunately, we are using a pairwise energy function (Eq. 1), and we can treat each amino acid pair separately. For a given amino acid i, and using a typical rotamer library, there are just n = 200 possible combinations of amino acid types and rotamers. For a pair ij, there are then n^2 = 200 $\times$ 200 combinations. In a protein of N = 100 amino acids, there are N^2 such pairs. If we arrange the amino acids along the lines and columns of a two-dimensional table, we will need $N \times n$ lines and columns to tabulate all the energies, corresponding to all pairs and all combinations of amino acid types and rotamers. This table (schematized in Figure 4) can be viewed as an energy matrix. It has $(Nn)^2$ = 400,000,000 elements in our example, which can be precalculated and stored. The complexity of this precalculation is only quadratic with respect to the protein length N, and the storage space needed will be only a few gigabytes. It is this precalculation that is the limiting step in our structure prediction and protein design methods. This is the computational step that is distributed to internaut volunteers. Once the energy matrix has been computed, the exploration of sequence and structure space can be done quickly and efficiently.

6 Object-oriented software for structural biology

The molecular mechanics model used here is a classic model, implemented in several large software packages that are standards in the fields of computational chemistry and structural biology. We have ported two of these packages to a distributed computing platform: X-PLOR [21] and CNS [22]. Both of these are software descendents of another standard package: CHARMM [36], which was initiated earlier, but continues to be actively developed today. X-PLOR was developed by Axel Brünger and his coworkers at Yale University in the early 1990's. CNS was developed as its successor, by Axel Brünger and an international group of collaborators. Our group has contributed to the

development of both packages, and one of us is a co-author of CNS [22, 37].

X-PLOR is predominantly written in Fortran, with only those parts closely tied to the operating system written in C. Nevertheless, building on and reinforcing the design policies of its parent, CHARMM, X-PLOR has a remarkably object-oriented character. Originally written at a time when efficient C and C++ compilers were not available (at least for supercomputers), X-PLOR was built with a highly-modular structure, so that system-dependent routines were limited to just one or two source modules, representing a few pages of code, out of almost 200,000 lines total. Thus, it could be easily ported to a few dozen different architectures, and used by several hundred research groups. Dynamic memory allocation, in the earliest versions of the program, was done with Fortran routines! Another key feature (also inherited from CHARMM) is that X-PLOR implements its own, rather high level command language. Again, in the days before Python or TCL, the X-PLOR developers had no choice but to build there own, flexible, high-level command interface.

The same philosophy was taken further with the CNS package. The command language is richer and a large library of macros and routines written in the CNS language are distributed with the program, representing 1/3 of the 150,000 lines of total source code. CNS is the current standard for modelling in the structural biology community, with about 1,000 research teams actively using it. About 2–3 macromolecular structures are published in the scientific literature each day that were determined with the help of the CNS program.

The calculation of the energy matrix is a double loop over the two amino acids ij in Eq. (1). To facilitate the construction of alternate amino acid types and rotamers, we introduce two "giant" amino acids, which each have 19 amino acid sidechains, corresponding to all possible types except glycine (which has no sidechain: Fig. 1). The loop structure is the following:

```
1 For all amino acids i {
2  For all amino acid types {
3    For all corresponding rotamers {
4     Move a giant residue into position i
5     Force its torsion angles into the desired
         rotamer using restrained energy minimization
6     For all amino acids j < i {
7      For all amino acid types {
8       For all corresponding rotamers {
9        Move a giant residue into position j
10       Force its torsion angles into the desired
            rotamer using restrained energy minimization
11       Minimize the structure slightly, taking into
            account the interactions of i and j with each
```

```
            other and with the protein backbone
12       Compute the interaction energy U(ij)
13       Write to file
14 }}}}}}
```

Because of the rich command language of both X-PLOR and CNS, the calculation could be implemented in both X-PLOR and CNS scripts of about 1,000 lines each. The scripts are available from our web site: biology.polytechnique.fr/biocomputing.

7 Exploring sequence space

For a given protein fold, defined here as the backbone structure, we want to identify the amino acid sequences that maximally stabilize the fold. Therefore, we need to solve an optimization problem in the space of sequences. To explore this space, one strategy is to perform random amino acid mutations, and to accept or reject them based on the resulting protein stability. This is very similar to a natural evolutionary process, limited to point mutations (changes of a few amino acids), in which more complicated events like chromosomal recombination or splicing errors are neglected. After each round of point mutations, the mutant protein's structure must be predicted. This corresponds to an easy homology modelling problem, since the backbone structure (the fold) is unchanged. Finally, from the predicted structure, the stability change must be estimated. This requires a model also for the unfolded state, as discussed above (Section 3). Indeed, a mutation may have a small effect on the folded state energy, but a large effect on the unfolded state. If the unfolded state is too strongly stabilized, the round of mutations will be rejected.

A more efficient search protocol, developed by Wernisch et al [23], intermixes sequence and structure changes more intimately, in a heuristic way. A "heuristic cycle" consists in the following steps. First, the amino acid type and rotamer at each position (or a large subset of positions) are randomized. Then, an iterative, steepest descent minimization is performed. The best amino acid type and rotamer are identified at the first position (given the current values at all the other positions); the best combination at the second position is identified, and so on. Each position is considered in turn, and multiple passes through the sequence are performed in this way, until no more improvement is obtained. This concludes the heuristic cycle; the final sequence represents a local optimum in sequence space. We emphasize again that the "best" sequence or rotamer means the one that maximizes the protein's stability; *i.e.*, the energy difference between the folded and unfolded states. After several 100,000 heuristic cycles, a representative set of good sequences is deemed to be obtained. The calculations are implemented in a C++ program, Proteus, initially adapted from the program Optimiser of Wernisch et al [23]. Proteus is available on our web site.

8 Protein design on the BOINC distributed computing platform

Our project is based on the Berkeley Open Infrastructure for Network Computing, BOINC (boinc.berkeley.edu) [24]. The client–server organization of BOINC is described elsewhere in this book. Several other projects based on BOINC are also described. Here, we limit ourselves to a brief overview of features specific to the Proteins@Home project.

Proteins@Home currently targets machines running the Windows XP operating system on Intel, or Intel-compatible processors. X-PLOR was compiled for Windows using the Intel Visual Fortran and C compilers. Only small changes to the X-PLOR code were needed, which are available from the authors on request. A particularity of the client setup is that X-PLOR runs as a subprocess of the main client program. The latter handles the graphics display of the screenserver windows and launches X-PLOR. The main client program is written in C++. It was adapted from the client program of Seti@Home, which is freely and generously available from the SETI developers (see setiathome.berkeley.edu). Graphics display is done with the OpenGL library [38].

On the server side, the workunits are prepared using a Perl script. A workunit corresponds to a pair of interacting amino acids within a particular protein. A generic X-PLOR (or CNS) script is edited for the specific protein at hand; for example, the 3D coordinates of the particular protein are embedded directly within the script. The script implements four of the six nested loops shown in Section 6: those corresponding to lines 2, 3, 7, and 8. It calculates all the energy matrix elements for a particular pair ij of amino acids. For a rotamer library with a total of 200 rotamers, 40,000 matrix elements are computed. For each one, about 50 steps of restrained energy minimization are performed, giving a total of two million energy evaluations per workunit. This requires about 1–2 hours on a typical volunteer machine. User memory requirements are quite small, and well-adapted to a wide variety of volunteer machines.

Because the workunits require a long calculation (1–2 hours), mechanisms for restarting after an interruption are essential. With the nested loop structure in the calculation, it is easy to write information to a file at the beginning of each of the two outermost iterations (lines 2, 3 in the script, Section 6). When the calculation starts, it looks for this file and starts the two outer loops in the appropriate place (choosing the appropriate amino acid type and rotamer for amino acid i). The duration of the two inner loops is about one minute. Thus, the efficiency of the procedure is high. A significant number of wasted cycles can only occur if the screensaver is launched, then interrupted every minute or less. In practice, few machine cycles are wasted on a typical volunteer machine.

9 Selected results: performance of the energy function

To illustrate the quality of the computational model, we describe briefly some recent testing and parameter optimization. We apply our model to two generic problems. First, we predict sidechain positions for 29 proteins, given their backbone structures and amino acid sequence. This is the so-called sidechain reconstruction problem. Second, we predict the stability changes associated with a large set of point mutations in twelve different proteins.

For both problems, we optimized both the surface coefficients and the dielectric constant in the energy function (Section 4). A typical predicted structure is shown in Figure 5, and compared to the known, experimental, Xray structure. Most of the sidechains in the predicted structure have been correctly placed, and overlap nicely with the experimental sidechains. On average, for our test set of 29 proteins (about 3000 sidechains), 80% of the amino acids have their sidechains in the correct χ_1 rotamer, similar to previous work with similar models. See [34] for a detailed description of our data.

For the stability changes, we considered 140 mutations in 12 proteins for which experimental stability measurements are available. After optimizing the surface coefficients and the dielectric constant (five adjustable parameters), we could reproduce the experimental stability changes to within about 2 kcal/mol. This is comparable to other studies with models of this complexity [39]. However, this error level is still a bit high, given the exponential relation between errors in the energy and the time spent in the folded state (see above). Fortunately, the effects of the errors are alleviated by two factors. First, when a round of mutations is performed (at the beginning of each heuristic cycle, for example; see above), the energy errors for different mutated amino acids are random quantities that have a tendency to partially cancel each other. Second, an empirical correction is added to the energy function, such that our computed sequences tend to reproduce the correct, experimental abundancies of the different amino acids in natural proteins. This empirical correction is expected to reduce the error for the stability changes, so that the final mean error is somewhere between 0 and 2 kcal/mol. The quality of the corresponding sequences is illustrated below.

10 Generating sequences for an oncogenic protein, the Src homology 3 domain

The Src Homology 3, or SH3 domain is one of the best-characterized members in the growing family of protein interaction modules. SH3 domains and their binding partners are abundant in species as different as yeast and *Homo sapiens*. Because of their involvement in protein–protein interactions, mutated forms of SH3 domains appear in

several forms of cancer. Baker and coworkers carried out the complete redesign of the C-Src SH3 domain in 2003 [40]. Their predicted sequence failed to adopt the SH3 fold. Rather, it stayed unfolded, as revealed by biophysical experiments. In 2002, Wodak and coworkers [30] reported the complete redesign of 11 SH3 domains. Although the overall sequence identity between the computed and natural sequences was around 23%, the experimental sequences were poorly reproduced for surface regions of the proteins. Here, we describe our results on four completely redesigned SH3 domains: the SH3 domain of the Grb2 protein, that of the Vav protein, of the c-Crk protein and the cytoskeletal protein spectrin. They include from 59 to 73 amino acids. Using X-PLOR, we computed the matrix elements for all pairs of amino acids. Protein sequences were then computed using the Proteus program. Proteus applies a heuristic procedure to search for the optimal amino acid and rotamer combination.

As explained above (Section 3), our unfolded state model is very simple. To obtain native-like sequences in the redesigned proteins, an empirical correction is added to the energy function, such that the overall amino acid composition of the computed sequences matches the experimental composition for this class of proteins. We obtained the experimental amino acid composition from the ExPASy Proteomics Server, which is available free of charge via the Internet. Only protein sequences that share a minimum of 35% sequence identity with the target structures were considered. Computed sequences were obtained with certain amino acid types frozen in their native type: alanines, cysteines, prolines and glycines were not mutated, because these amino acids have a particular, strong effect on the local backbone structure, which is assumed fixed. The empirical correction was then computed iteratively, by doing rounds of sequence generation (with Proteus) and updating the correction, until convergence (after about 200,000 cycles for each protein).

Inspection of the chosen 3D structures shows (see Figure 6) that a small beta strand [4]at the N-terminal part is followed by an extensive loop, which includes between 14 and 18 amino acids. The loop is the followed by additional four beta-strands, giving an ensemble of five antiparallel strands. Strands two and three, and strands three and four are connected by small turns, which comprise around five amino acids.

Figure 6 compares experimental and computed sequences obtained for the Grb2 SH3 domain. The four upper sequences are experimental sequences from four different species: Rattus norvegicus, Pongo pygmaeus, Mouse, and Zebrafish. The fifth is the proto oncogene C-crk (P38) (Adapter molecule crk). Below, 25 computed sequences are listed. These 25 proteins exhibit the highest similarity to the native sequences. Blosum matrix scores are used to score the sequences [41]. High scores indicate closely related sequences. Randomly chosen SH3 sequences, when compared to Grb2, have an average score of around 150. The depicted sequences score from 200 to 215. General speaking, Blosum distinguishes high favorable mutations (changes within one colored group), moderately favorable mutations (red to blue, orange to yellow), neutral muta-

tions (orange to pink), and unfavorable mutations (blue or red to orange or yellow). Comparison of experimental and computed sequences reveals the conservation of many essential features of the native sequences, consistent with the high Blosum score of the calculated sequences.

It is of interest to analyze the behavior in the core and at the surface, as mentioned. The two phenylalanines in positions nine and nineteen (marked as F in see Figure 6) are core residues, and their two aromatic sidechains interact though π–π stacking. This motif is reproduced in the computed sequences. The arginines in positions 21 and 22 are solvent-exposed. Here, we predict the mutation of the first arginine ("R") to asparagine ("N") and the conservation of the second arginine (giving a red column in Figure 6). The YV motif in positions 2–3 is conserved in the experimental sequences and occurs with a high abundancy in the computed sequences. From position 8 to 22 we predict sequences that are consistent with the experimental ones. The LF motif in position 6 and 7 is replaced by polar (Q,N) or ionic (K,K) amino acids, or by mixture of ionic amino acids cand threonine (T). Inspection of the 3D structure shows that this LF motif is at the begining part of a surface loop. Solvent exposed polar or ionized amino acids are energetically favorable in this type of position. For positions 24 and 25, we predict LI, compared to IL in the native c-Crk sequence. The WW motif at positions 35 and 36 of the native sequence is conserved throughout the SH3 family. The computed sequences reproduce this motif in many cases.

Overall, the results show that our method has the potential to become a powerful tool to predict amino acid sequences that share the most important, native-like features, and hence will provide a reliable tool for fold recognition, structure prediction and protein design.

References

[1] SERVICE, R. F. Gene sequencing: The race for the $1000 genome. *Science 311* (2006), 1544–1546.

[2] E. S. LANDER ET AL. Initial sequencing and analysis of the human genome. *Nature 409* (2001), 860–921.

[3] C. VENTER ET AL. The sequence of the human genome. *Science 291* (2001), 1304–1351.

[4] BRANDEN, C., AND TOOZE, J. *Introduction to Protein Structure*. Garland Publishing, NY, 1999.

[5] SCHUELER-FURMAN, O., WANG, C., BRADLEY, P., MISURA, K., AND BAKER, D. Progress in modeling of protein structures and interactions. *Science 310* (2005), 638–642.

[6] MANNHOLD, R., KUBINYI, H., TIMMERMAN, H., AND LENGAUER, EDS., T. *Bioinformatics: From Genomes to Drugs.* Wiley, New York, 2002.

[7] BAKER, D., AND SALI, A. Protein structure prediction and structural genomics. *Science 294* (2001), 93–96.

[8] MARTI-RENOM, M., STUART, A., FISER, A., SANCHEZ, R., MELO, F., AND SALI, A. Comparative protein structure modeling of genes and genomes. *Annu. Rev. Biophys. Biomol. Struct. 29* (2000), 291–325.

[9] SHIRTS, M., AND PANDE, V. Screen savers of the world, Unite! *Science 290* (2002), 1903–1904.

[10] PONDER, J., AND RICHARDS, F. M. Tertiary templates for proteins: Use of packing criteria in the enumeration of allowed sequences for different structural classes. *J. Mol. Biol. 193* (1988), 775–791.

[11] EISENBERG, D. A problem for the theory of biological structure. *Nature 295* (1982), 99–100.

[12] ANDREEVA, A., HOWORTH, D., BRENNER, S., HUBBARD, J., C, C. C., AND MURZIN, A. SCOP database in 2004: refi nements integrate structure and sequence family data. *Nucl. Acids Res. 32* (2004), D226–229.

[13] PEARL, F., TODD, A., SILLITOE, I., DIBLEY, M., REDFERN, O., LEWIS, T., BENNETT, C., MARSDEN, R., GRANT, A., LEE, D., AKPOR, A., MAIBAUM, M., HARRISON, A., DALLMAN, T., REEVES, G., DIBOUN, I., ADDOU, S., LISE, S., JOHNSTON, C., SILLERO, A., THORNTON, J., AND ORENGO, C. The CATH domain structure database and related resources Gene3D and DHS provide comprehensive domain family information for genome analysis. *Nucl. Acids Res. 33* (2005), D247–251.

[14] ORENGO, C., AND THORNTON, J. Protein families and their evolution-a structural perspective. *Ann. Rev. Biochem. 74* (2005), 867–900.

[15] DAHIYAT, B., AND MAYO, S. De novo protein design: fully automated sequence selection. *Science 278* (1997), 82–87.

[16] LAZAR, G., MARSALL, S., PLECS, J., MAYO, S., AND DESJARLAIS, J. Designing proteins for therapeutic applications. *Curr. Opin. Struct. Biol. 13* (2003), 513–518.

[17] 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.

[18] DWYER, M., LOOGER, L., AND HELLINGA, H. Computational design of a biologically active enzyme. *Science 304* (2004), 1967–1971.

[19] BUTTERFOSS, G., AND KUHLMAN, B. Computer-based design of novel protein structures. *Ann. Rev. Biophys. Biomolec. Struct. 35* (2006), 49–65.

[20] MCCAMMON, J., AND HARVEY, S. *Dynamics of proteins and nucleic acids*. Cambridge University Press, Cambridge, 1987.

[21] BRÜNGER, A. T. X-PLOR *version 3.1, A System for X-ray crystallography and NMR*. Yale University Press, New Haven, 1992.

[22] 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.

[23] WERNISCH, L., HÉRY, S., AND WODAK, S. Automatic protein design with all atom force fi elds by exact and heuristic optimization. *J. Mol. Biol. 301* (2000), 713–736.

[24] ANDERSON, D. P., AND FEDAK, G. The computational and storage potential of volunteer computing. In *IEEE/ACM International Symposium on Cluster Computing and the Grid* (2006), IEEE Computer Society Press, USA.

[25] JANIN, J., WODAK, S., LEVITT, M., AND MAIGRET, B. Conformation of amino acid sidechains in proteins. *J. Mol. Biol. 125* (1978), 357–386.

[26] 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.

[27] DUNBRACK, R., AND KARPLUS, M. Backbone-dependent rotamer library for proteins. application to sidechain prediction. *J. Mol. Biol. 230* (1993), 543–574.

[28] DUNBRACK, R., AND COHEN, F. Bayesian statistical analysis of protein sidechain rotamer preferences. *Prot. Sci. 6* (1997), 1661–1681.

[29] BERMAN, H., WESTBROOK, J., FENG, Z., GILLILAND, G., BHAT, T., WEISSIG, H., SHINDYALOV, I., AND BOURNE, P. The Protein Data Bank. *Nucl. Acids Res. 28* (2000), 235–242.

[30] JARAMILLO, A., WODAK, S., WERNISCH, L., AND HÉRY, S. Folding free energy function selects native-like protein sequences in the core but not on the surface. *Proc. Natl. Acad. Sci. USA 99* (2003), 13554.

[31] SAUNDERS, C., AND BAKER, D. Recapitulation of protein family divergence using flexible backbone protein design. *J. Mol. Biol. 346* (2005), 631–644.

[32] BROOKS, C., KARPLUS, M., AND PETTITT, M. Proteins: a theoretical perspective of dynamics, structure and thermodynamics. *Adv. Chem. Phys. 71* (1987), 1–259.

[33] MCCAMMON, J., GELIN, B., AND KARPLUS, M. Dynamics of folded proteins. *Nature 267* (1977), 585.

[34] LOPES, A., ALEKSANDROV, A., BATHELT, C., ARCHONTIS, G., AND SIMONSON, T. Computational sidechain placement and protein mutagenesis with implicit solvent models. *Proteins xxx* (2007), 000.

[35] ROUX, B., AND SIMONSON, T. Implicit solvent models. *Biophys. Chem. 78* (1999), 1–20.

[36] 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.

[37] 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.

[38] WRIGHT, R. S., AND LIPCHAK, B. *OpenGL SuperBible*. SAMS, New York, 2006.

[39] GUÉROIS, R., NIELSEN, J., 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.

[40] 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.

[41] HENIKOFF, S., AND HENIKOFF, J. Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. USA 89* (1992), 10915–10919.

Figure Captions

1. **Protein building blocks. (A)** The polypeptide chain, with a closeup showing the chemical form of the "backbone", to which the 'side chains' R_i, R_{i+1}, ..., are attached. The (C=O) and (N-H) groups are linked by the "peptide" bond, which has a partial double bond character, making the (C=O)-(N-H) "peptide group" stiff and approximately planar. The torsion angles ϕ and ψ, around single bonds, are soft. **(B)** The side chains R_i, R_{i+1}, ..., can be any of the twenty common aminoacid side chains, shown here labelled by their conventionial three-letter abbreviations (see also text). The horizontal axis corresponds roughly to the polarity of the sidechain; the vertical axis corresponds to size.

2. **(A) Protein folding.** The small Trpcage protein in an unfolded conformation (left) and its stable, folded conformation (right). The protein backbone is shown in a simplified, green, tube/ribbon representation. A single, trpytophan sidechain is also shown, which forms the core of the folded structure. **(B) Folded proteins** Space-filling views of cytochrome c (an electron carrier in the respiratory chain) and hemoglobin (an oxygen carrier in the blood), along with a water molecule, approximately to scale.

3. **(A) The inverse folding problem.** A large number of sequences (left) are tested for their ability to stabilize a given backbone fold (center). Favorable sequences are retained (right). In effect, the sequences are "filtered" through the 3D structure. **(B) Fold recognition.** Given the amino acid sequence corresponding to a new gene, we try to match it to a large but finite number of known protein structures. The matching can be performed by comparing the new sequence to the sequence families generated (A) for all the known protein domain structure.

4. **The energy matrix** (right) corresponding to a particular backbone fold (left). In this toy example, the "protein" is a tripeptide (three amino acids); at each position, two different amino acid types are allowed (A or B), with two rotamers each. The backbone is red; amino acid positions are numbered. The interactions highlighted on the structure by black and red arrows, respectively, correspond to the matrix elements highlighted by a black and red dot. The grey square on the matrix corresponds to a Proteins@Home workunit (one pair of amino acid positions).

5. **Sidechain reconstruction** for the protein Staphylococcal nuclease. The protein backbone is shown in a simplified "ribbon" representation. Sidechains are shown as sticks, with the experimental positions (light blue) and computed positions (grey) superimposed.

6. **Computed sequences. Top:** An "alignment" of experimental and computed sequences of the protein Grb2 (see text). The top four lines correspond to the experimental sequences of Grb2 from Rattus norvegicus, Pongo pygmaeus, Mouse, and Zebrafish. Amino acids are indicated by their conventional one-letter abbreviation [4]. The following lines correspond to representative computed sequences. To highlight the similarities (and differences) betweenthe sequences, amino acids are colored according to the chemical properties of their sidechains (aliphatic, aromatic, polar, ionized, weakly-polar). **Bottom:** 3D Grb2 structures colored according to the experimental (left) and computed (right) sequences. The two color schemes are seen to agree qualitatively, indicating that the computed sequences reproduce the experimental pattern of amino acid types.