Chapter 1: Introduction

Proteins are the ultimate nanotechnology devices. Inside our cells, these molecular machines do all sorts of fantastic things — catalyze chemical reactions, create the electrical signals in neurons, copy DNA, move vesicles around, transmit information, and generally get the job done inside our bodies. Many of these functions have been studied and worked out in great detail, a triumph of modern molecular biology.

The inimitable Richard Feynman, a great source to consult for practical scientific philosophy, wrote that "What I can not create I can not understand." So in that spirit, we believe that if we claim to understand how proteins work, we should be able to make predictions about their behavior, and if we really understand how they work, we should be able to design proteins with new functions. Thus, we employ an engineering approach to protein design, meaning that we start with a physical model of how proteins work, and we use this model to predict how mutations affect a protein's activity, and to design proteins with new functions.

Relationship to the protein folding problem

The protein folding problem asks if you can predict the structure of a protein, given its amino acid sequence. The protein design problem asks if you can find an amino acid sequence that folds into a particular structure. Which problem is harder? Can you solve one without solving the other?

We believe that proteins can be designed computationally without solving the protein folding problem, and that this approach will provide unique insights into how proteins work. From one perspective, design is easier than folding: out of the millions of sequences that fold into your target structure, you only need to find one of them. From another perspective, design is harder than folding: the design algorithm will exploit any flaws in your protein energy calculations if they appear to stabilize the target structure.

Design goal

The overall goal is to develop an algorithm that will take the structure of a scaffold protein, and the structure of a small molecule, and design a set of mutations needed to create a binding site in the scaffold. We only consider mutations at a limited number of "design positions"; the rest of the protein simply serves as a rigid structure for constraining the conformational flexibility of the designed binding site. Thus, we only need to consider a limited range of protein conformations and do not need to solve the full protein folding problem.

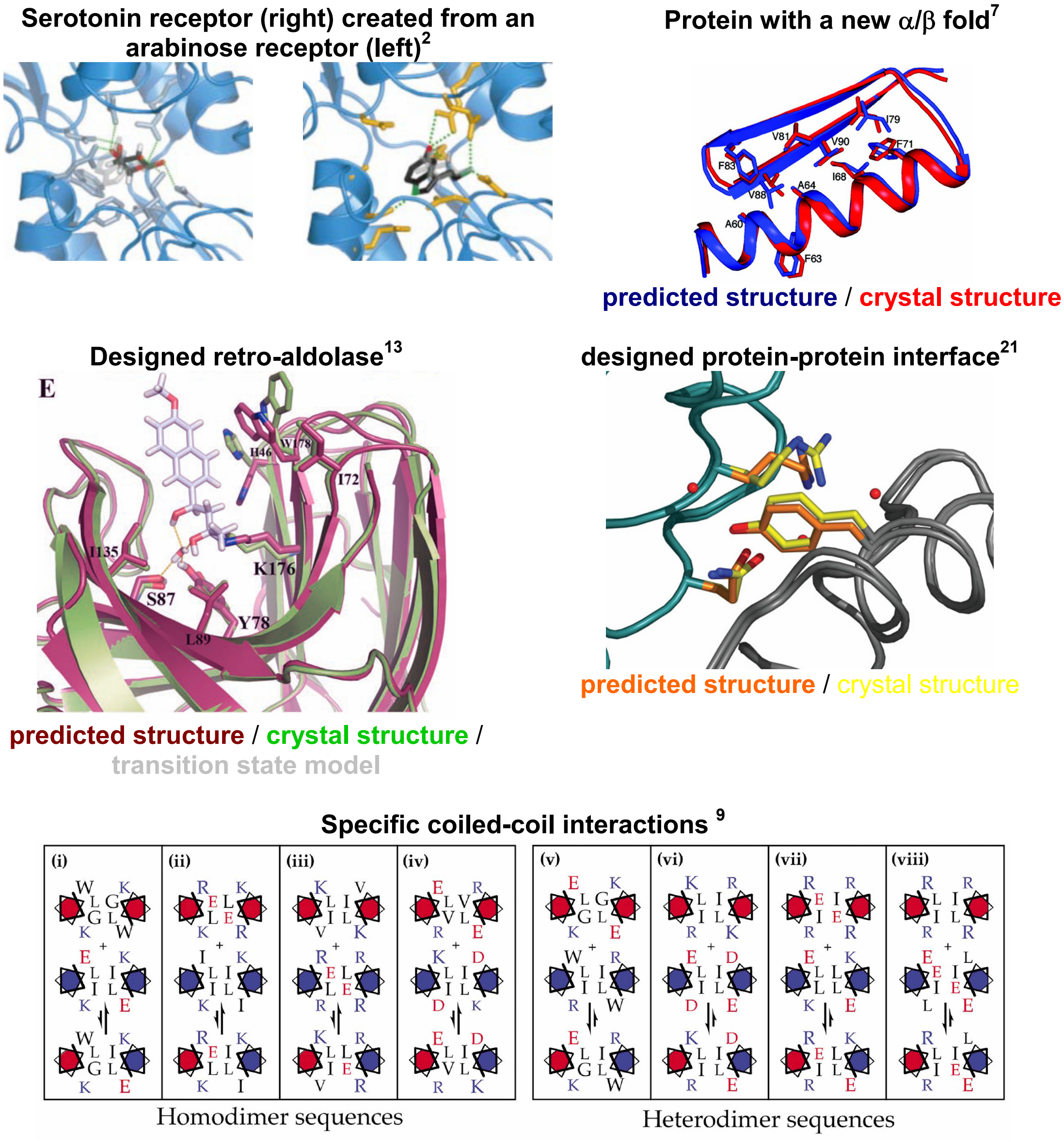
The ideal scaffold protein can host a wide range of different binding sites. It should be stable, so it can accommodate destabilizing mutations. The proposed binding site should also be lined with sidechain contacts, which will be easier to modify by mutation than backbone contacts. Natural scaffold proteins include antibodies, which bind different antigens, and alpha/beta barrel proteins, which host a wide range of enzyme active sites.[1] In this thesis, we use ribose binding protein, based on the pioneering work of Hellinga,[2,3] who used this as a scaffold for computational protein design (although note that ref. [3] was recently retracted[4]).

Innovative aspects of the research

The computational protein design field is small, but has seen some remarkable successes.[2,5-13] In each of these examples, the protein was designed in a computer, then experimentally validated with a crystal structure or activity measurement (Figure 1). However, the generality of these design algorithms is unclear, because each lab typically uses its own custom software, and standard models for protein design have not yet emerged. Furthermore, protein design typically requires multiple iterations of feedback from experimental results before reaching the desired target. We would like to address these problems using better sampling strategies and more accurate energy functions.

Many of the individual components of our design calculation are related to algorithms that have been proposed and validated before in the literature, including the molecular mechanics potential energy function,[14] continuum solvent model,[15,16] side chain rotamer library,[17] ligand docking procedure,[18] probabilistic description of protein conformation and mean field algorithm,[19] multi-state design framework,[9] and genetic algorithm.[20]

The unique aspect of this project is the combination of these existing methods to tackle interesting design problems. This integration has never been achieved before, partly because of the technical difficulty, and partly because others in this field have broken down the design problem into different pieces. Especially notable is our treatment of ligand flexibility, protonation equilibria, conformational entropy, high resolution structural sampling, accurate continuum solvation model, and explicit unbound and unfolded reference states. These factors are often ignored in protein design calculations, despite evidence that they are important in ligand binding.


Serotonin receptor (right) created from an arabinose receptor (left)[2]

Protein with a new α/β fold[7]

predicted structure / crystal structure

Designed retro-aldolase[13]

designed protein-protein interface[21]

predicted structure / crystal structure

predicted structure / crystal structure / transition state model

Specific coiled-coil interactions [9]

Figure 1. Examples of computational protein design.

Models for understanding molecular recognition

Qualitatively, specific molecular recognition occurs between molecules with steric complementarity and chemical complementarity (charge patterning and hydrogen bonding)[22] (see Figure 2). Steric complementarity is important both for producing favorable van der Waals interactions between ligand and receptor, and also for preventing the formation of buried pockets of water. Charge patterning is important for specificity. In water, intermolecular interactions are often strengthened when salt bridges are replaced with uncharged groups. Thus, charge patterning does not always create the most stable complexes. It does, however, confer specificity: a single charge buried in a molecular interface without a salt bridge partner is extremely unfavorable, thus preventing ligands with the wrong charge pattern from binding.

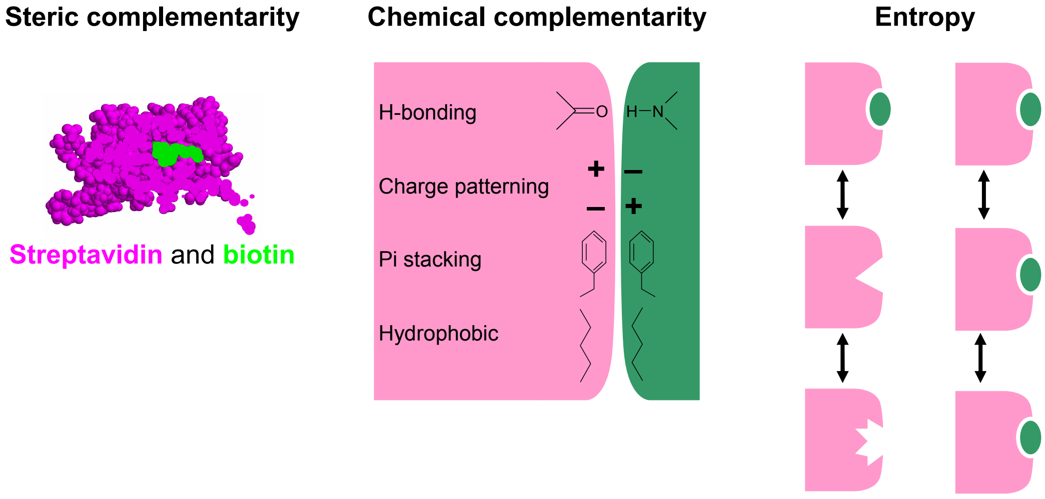


Figure 2. Qualitative model of molecular recognition.

While a qualitative understanding is the most intuitive way to think about molecular recognition, it does have limitations. Each molecular interface contains many favorable and unfavorable interactions, so it is difficult to know whether the interaction is net favorable without adding up the actual strength of each interaction. Furthermore, protein backbone and side chain conformations shift in response to mutations, and these conformational changes are difficult to predict by just eyeballing the structure.

If qualitative understanding is one extreme, the other extreme is a full quantum mechanical treatment of the protein and solvent. From this perspective, the energy of the system is solely based on Coulomb's law interactions between electrons and nuclei, and the evolution of the system is given by the Schrödinger equation. In addition to being totally impractical computationally, this approach also clouds our understanding of the system: all of the energy is electrostatic and is not partitioned into more understandable categories.

Fortunately, over the past thirty years, several groups have developed molecular mechanics potential energy functions (Figure 6), which model proteins as a collection of atoms connected by springs that hold bond lengths and angles near their standard values. A torsion angle energy term penalizes eclipsed conformations. Standard molecular mechanics potentials also include two energy terms for atoms that are not bonded to each other: van der Waals interactions, and Coulomb's law interactions between charges. Some versions also include a hydrogen-bonding term, but often this is handled by the van der Waals and Coulomb's law terms.

These molecular mechanics potentials are typically used in molecular dynamics simulations, which trace protein motions over time. At each time step, the computer uses the molecular mechanics potential to calculate a force on each atom, then uses the force to update its velocity and position. Unfortunately, most of the computer time is spent simulating water molecules, even though the protein is usually the molecule of interest. Water does significantly affect the behavior of the protein, but fortunately many of these effects can be understood in a smeared-out continuum model. First, water solvates charges: it exerts an attractive force pulling both positive and negative charges towards the protein's surface. Second, water screens charges: interactions between charges in a vacuum are weakened by a factor of 80 when they are placed in water. Third, water's hydrogen bond network is disrupted at the protein surface: this is why oil does not dissolve in water. The first two factors can be treated by modeling water as a continuum dielectric, and the third factor can be treated with a surface tension term.

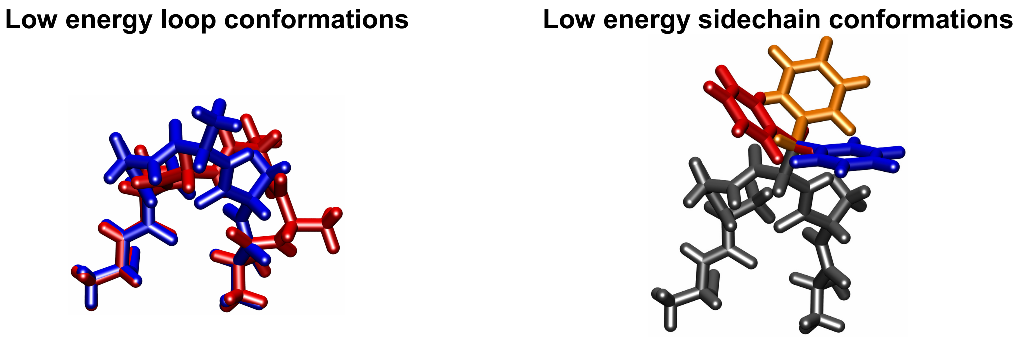
Finding a protein's low energy conformations

Even when water is treated in a continuum model, molecular dynamics is extremely slow, because the simulation typically proceeds in femtosecond time steps. Simulating a 1 second unbinding event would take 1 million years of time on a typical desktop computer.

Most of the time in a molecular dynamics simulation, the protein is simply jiggling around some equilibrium conformation. Every once in a while, the protein crosses an energetic barrier and settles into another conformation. To speed up the calculation, can we just skip these barriers and directly identify the low energy conformations? Our strategy for doing this is to break the protein-ligand system into three parts: protein backbone conformation, protein side chain conformation, and ligand position / conformation. Then, we can find the low energy configurations of each part, and mix and match these low energy conformations to explore the whole system's conformational space.

To find the low energy conformations of the protein backbone at design positions, we use the following rather elaborate scheme (all of the simpler strategies we tried missed some conformations). First, we take snapshots from a high temperature molecular dynamics simulations. Second, we sample the backbone ϕ and ψ angles on a 30° grid. Third, we search the entire Protein Data Bank for loops whose endpoints match the fixed portions of our protein scaffold. Finally, we feed all of these conformations into a genetic algorithm search, which randomly perturbs and splices together structures to generate new structures.

To find the low energy protein side chain conformations, we clustered side chain conformations observed in the Protein Data Bank into a rotamer library, placed rotamers at each design position, and applied an energy cutoff to eliminate unfavorable rotamers (Figure 3).



Low energy loop conformations

Low energy sidechain conformations

Figure 3. Low energy protein conformations.

To find the low energy docked ligand positions / conformations, we move the ligand over a translational and rotational grid, and eliminate ligand orientations that clash with the scaffold, or do not make sufficient contact with side chains at design positions.

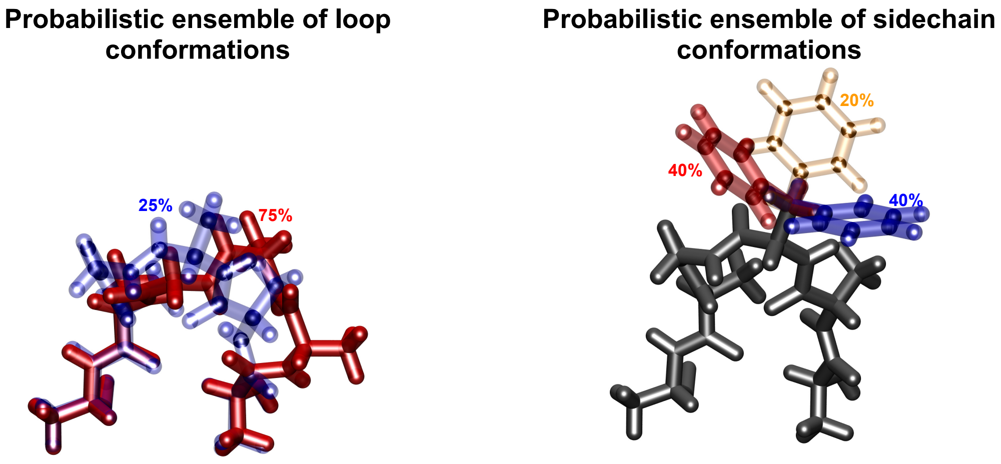
One advantage of identifying all of the low-energy configurations of the protein-ligand system at the beginning of the calculation is that all of the energy terms can be precomputed (both intrinsic energies and interaction energy matrices). These are calculated using a standard molecular mechanics potential with continuum water, as described in the previous section. After making this initial investment of computing time, the energy of a specific configuration of the protein-ligand system can be rapidly calculated simply by adding up the appropriate terms from the precomputed energy matrices.

Probabilistic description of protein conformation

We represent the protein/ligand system as a probabilistic ensemble of the low energy backbone, side chain, and ligand conformations/positions (Figure 4). The probabilities are set by a mean field calculation,[19] which iteratively updates the probability of each side chain and ligand conformation based on its intrinsic energy plus its probability weighted interaction energies.

The probabilistic ensemble allows us to model conformational changes and thermal fluctuations. Mutating design positions to different amino acids may shift the protein conformation, and this shift will be described as a change in the probabilities of various conformations. A single rigid structure would be described by setting the probability of a single conformation to 1 and everything else to 0. However, in order to accurately calculate ligand binding affinity, it is important to model the protein's thermal fluctuations by allowing probabilities besides 0 and 1. The probabilistic model is more realistic, because proteins spend very little time in their global minimum potential energy conformation. Furthermore, if the protein can adopt 1000 conformations and only 1 of them binds ligand, then this decreases its binding affinity by 1000-fold. We can model these sorts of entropic effects using a probabilistic model.

Once the probabilities from the mean field calculation have converged, we can calculate the free energy of the system by adding the probability weighted average energy, and the energy due to the conformational entropy.


Probabilistic ensemble of loop conformations

Probabilistic ensemble of sidechain conformations

Figure 4. Probabilistic model of protein structure.

Evolving binding sites

An important point is that, experimentally, we can only control the amino acid sequence. The structure of the protein is determined by the energetics. Thus, we separate our sequence and structural optimization.

Using the procedure described above, we can calculate the free energy of several different states, given the amino acid sequence at design positions: the protein and ligand free in solution, the protein bound to various different ligands, and the unfolded protein. From these energies, we can calculate the stability of the protein, and its affinity and specificity for various ligands. Based on these energies, we can assign a score to each sequence. The score can also include structural criteria, such as the predicted geometry of catalytic residues.

We use a genetic algorithm[20] to evolve sequences that optimize our chosen scoring function. The genetic algorithm starts with a population of random sequences, and then alternates between rounds of selection and recombination/mutation (Figure 24 on p. 85).