Simulating the Folding of Proteins

Tyler Gibson

Steven Brunetti

Armando Diaz

May 08, 2009

Submitted to Dr. Miguel Bagajewicz

Table of Contents

Abstract

The computational modeling of protein structures is an active field of research that could provide valuable insight on the nature of proteins and allow for a comparatively quick method of determining the structures of unknown proteins at a much lower cost than that of performing laboratory experiments. Ab initio computational methods provide a promising means to predict the structure of a protein based on thermodynamic data and the sequence of amino acids making up the structure. To attempt to reduce computational time while approximating the manufacture of a polypeptide, a linear optimization technique was employed to reduce the free energy of the molecule, and the polypeptide was slowly constructed by optimizing a short chain of amino acids and then adding additional residues to the C-terminal end of the chain and re-optimizing the process. Additionally, a genetic algorithm was employed as an alternative optimization method. Both methods used the ECEPP/3 energy parameters, the GBr6 model of electrostatic solvation energy, and an approximation of the volume of the hydration shell around the polypeptide to determine the conformational energy of the polypeptides. The linear optimization technique was found to be an inferior method in terms of both computational efficiency and the accuracy of results, often giving unrealistic or poorly optimized outputs. On the other hand, the genetic algorithm performed admirably to find the global minimum of conformational free energy. However, it produced structures that exist at lower energy than the NMR measured structures, indicating inaccuracies within the energy parameters used. Further research should focus on determining the optimum energy parameters to enable the successfully optimized results to correspond with actual native structures as found in nature.

Introduction

In 1957, Christian Anfinsen performed his famous experiment demonstrating that proteins fold spontaneously into their native, biologically active states. This experiment suggests that a protein's three-dimensional structure is determined based on its sequence of amino acid residues, and as a result knowledge of the primary structure of a protein should make it possible to determine the tertiary structure of a protein. This also implies that there is a reliable mechanism or combination of mechanisms that guide the folding of any particular protein, as protein folding is a very rapid process that occurs from a time span of a few milliseconds to a few seconds[1]. Computer-based simulations of protein folding are an active field of research today, as the successful prediction of protein structures will allow an unprecedented glimpse into the nature of proteins that have not yet been synthesized or analyzed in the laboratory[2]. By taking an ab initio approach that seeks to model proteins based on their thermodynamic optima, it should be possible to obtain a close approximation to their three-dimensional native structures.

Two approaches of ab initio modeling are used here. In one, the gradient of the forces is analyzed to seek local minima in a method that seeks to approximate the process of the polypeptide chain growing during translation in the ribosome. The other method uses genetic algorithms to seek the global optima from a wider range of possibilities.

Protein Structure

Proteins are composed of long chains of amino acids that have been polymerized into polypeptides. Amino acids are chiral organic molecules that are composed of an amine functional group, a carboxylate functional group, a hydrogen atom, and a variable "R" group all bonded to a central carbon atom in a levorotary conformation as shown in Figure 1. There are twenty essential amino acids that make up every protein, and they are chained together to form a peptide bond through a dehydration reaction that takes place in the ribosome.

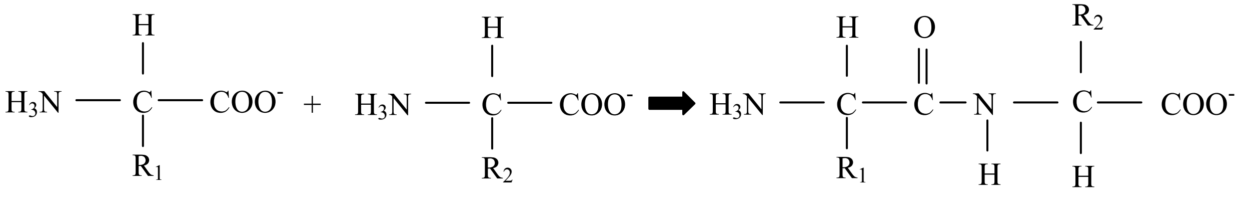

Figure 1: The dehydration reaction between two amino acids forms a dipeptide.

The R groups differentiate the amino acid residues from one another, and as a result each residue in the polypeptide can be identified based on its side chain. A protein's primary structure is considered to be the linear list of amino acid residues. Also, it includes any disulfide bonds that exist in between cysteine residues, which contain sulfhydryl functional groups that can become linked. The primary structure of a protein is often described by the one-letter abbreviation for each amino acid, starting from the N-terminus, where the amino acid residue contains its original charged amine group instead of a peptide bond. Less often they are described by their three-letter abbreviations, so a tripeptide consisting of cysteine, aspartate, and tryptophan could be abbreviated as either Cys-Asp-Trp or CDW.

The secondary structure of an amino acid describes the more common shapes that the nitrogen and carbon "backbone" of the polypeptide chains tend to fold into, such as helices or pleated sheets. These structures typically arise as a result of hydrogen bonding between atoms along the backbone, the most common being the right-handed α-helix and the pleated β-sheet shown in Figure 2, which are compact and thermodynamically favorable conformations.

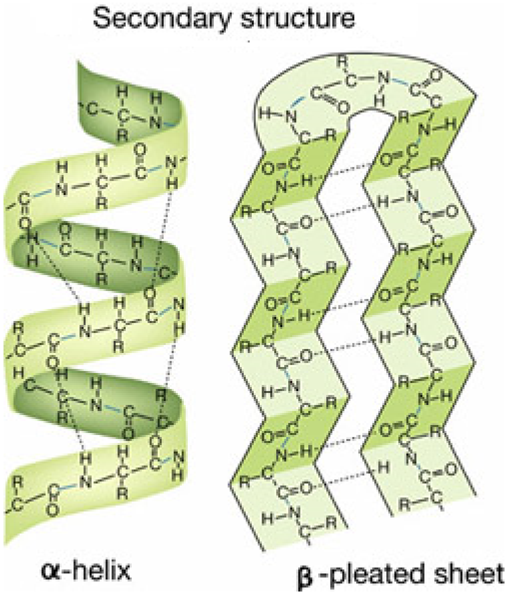

Figure 2: The more common secondary structures in a polypeptide. Image is ©2007 ABCTE.

A polypeptide's tertiary structure is the net three-dimensional structure of a single polypeptide chain that includes the folding of its secondary structures and the locations of the side chains off the backbone. This information can be experimentally determined using X-ray diffraction or nuclear magnetic resonance, and is often stored in protein databases that are accessible on the internet. Hydrophobic and hydrophilic interactions are thought to play a significant role in the development of the tertiary structure of a protein, as proteins tend to consist of hydrophobic cores with the more hydrophilic residues on the outer layers of the protein, where they are more accessible to solution. Finally, quaternary structure is the composite structure formed by multiple polypeptide chains that have linked together into a larger unit, each with its own unique tertiary structure.

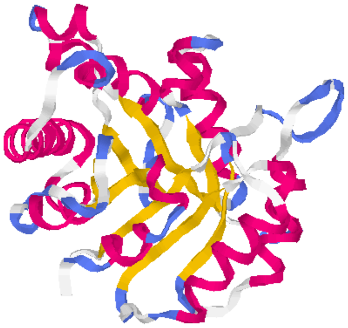

Figure 3: A conventional graphic showing the tertiary structure of triose phosphate isomerase. Alpha helixes are conventionally represented by coils, while beta sheets are shown as flat arrows.

Protein Thermodynamics

Anfinsen's 1957 experiment showed that RNAse A could be reversibly denatured and re-natured by removing the denaturant, implying that proteins exist in conformations that optimize their free energy. This theory has given rise to the concept of the "energy well," where in a plot of energy against the conformation, the native state exists at a global minimum like seen in Figure 4[1]. However, there are several local optima where a protein could theoretically exist in a semi stable conformation. The native state is often a very specific conformation for optimum stability. It is likely that proteins, when they misfold, have fallen into a different minimum. This presents an obstacle to attempts to compute the native conformation of a protein by a search for the global optimum, as it is difficult to determine which conformation is the native state of a polypeptide.

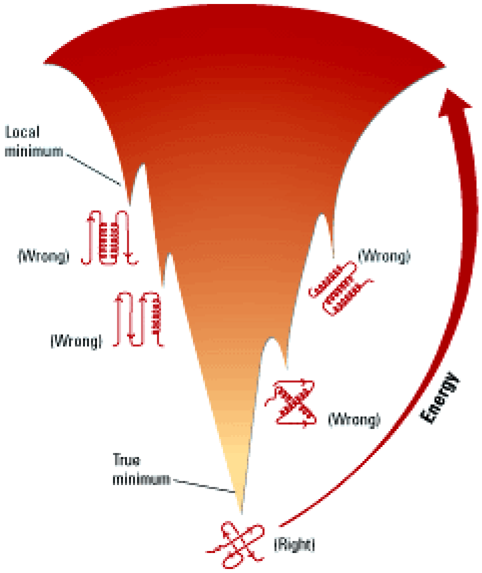

Figure 4: The "energy well" concept illustrated.

Predicting the Conformation of a Protein

There are multiple computational methods currently being used to determine the structure of a protein based on its primary structure. One common technique, homology, involves designing programs to recognize patterns in unfamiliar amino acids, allowing it to apply folding conformations from proteins that have already been analyzed[3,4]. This method requires that the unknown proteins be similar to those already measured. Molecular dynamics have also been used to simulate protein folding and to attempt to gain a sense of the mechanisms by which the folding occurs; however, these simulations are currently limited by current computational power. One of the more impressive results provided from this method obtained a simulation that spanned the length of one microsecond[5]. Other promising forms of computer simulation attempt to create an *ab initio* prediction that attempts to determine the protein structure from no more information than the primary structure, often using thermodynamics to determine the energy of the conformation and a computational search method to seek the optimum[6,7]. Because this method is compatible with current technological limitations, it offers a promising method to predict the three-dimensional structure of a protein.

Finding the thermodynamic optimum for a protein configuration based on its torsion angles is a computationally intensive task, as the torsion angles have a wide degree of variability. One method used herein will attempt to model the construction of a polypeptide chain as it emerges from the ribosome, performing local optimizations on smaller segments of the polypeptide chain and building the entire chain in a similar fashion. In addition, genetic algorithms have been shown to have great potential for these predictions as well, and provide a good basis of comparison for both methods[8].

Genetic Algorithms

In 1971, Ingo Rechenberg introduced the concept of evolutional computing as an optimization method in his Ph.D. thesis *Evolutionsstrategie*[9]. This led directly to the invention of some of the first genetic algorithms by John Holland which became popularized in his 1975 book *Adaptation in Natural and Artificial Systems*[10]. Since then, the algorithms have seen application in numerous problem domains from engineering to time tabling and scheduling, usually as an approach to finding the global optimum of a problem set. They are particularly useful in a complex fitness landscape where a genetic algorithm is less vulnerable to converging at local optima than a gradient search.

The techniques used to create a genetic algorithm are inspired by evolutionary biology and include selection, inheritance, mutation, and recombination (or crossover). In nature, a population is subject to natural selection. Differences in fitness between individuals within the population lead to differences in the survival probability for each individual. Those individuals that reach maturity pass on their genes—and the traits that gave higher fitness values—to the next generation through reproduction and recombination. In recombination, the genes inherited by the offspring are decided by a shuffling of the parental chromosomes. Occasionally, genes are changed through mutation creating more variation in the gene pool. The mutation may be deleterious or beneficial to the overall fitness of the individual which is reflected in a change of its survival probability. Changes in the chromosomes of successive generations are all guided by natural selection as beneficial traits are preserved while the deleterious ones are weeded out in what is known as evolution.

In order for this process of evolution to be mimicked in the protein folding program, a "chromosome" and "gene" must be defined in the context of the genetic algorithm. The model for the individual is its chromosome, whose genes code for particular traits. A chromosome in this case is the tertiary structure of the amino acid sequence that results from its unique set of dihedral angles and a gene is a particular dihedral angle in the set. In addition, the 'fitness' of a chromosome is derived from the total potential energy calculated as a result of the protein being in the particular conformation, where higher fitness corresponds with lower potential energy. Applying selection, crossovers, and mutations over many iterations is intended to move the population towards a global optimum. This optimum should represent the lowest energy conformation of the protein or the 'native conformation'.[11]

Methodology

Information on the conformation of proteins is often stored in terms of the internal molecular coordinates, requiring the program to convert the coordinates into a Cartesian plane. From these coordinates the energy of the system can be found based on the electrostatic forces, Lennard-Jones constants, and hydrogen bonding interactions as well as the energy of solvation which is found implicitly. An optimization method can then be used to minimize the energy of the system.

Determining Atom Positions in Cartesian Coordinates

The internal molecular coordinates are conventionally listed as the bond length r, which measures the distance between two atoms i and j, the bond angle θ made by three atoms $i, j,$ and k within the same plane. The dihedral or torsion angle ω is defined, for any four atoms $i, j, k,$ and l, as the angle between the plane formed by atoms $i, j,$ and k and the plane formed by the atoms $j, k,$ and l[12]. The coordinates of one atom can be found by a matrix multiplication of the internal coordinates with the coordinates of the previous atom[13]. By placing the first atom in the chain at the origin, this operation can be adapted into Equation 1 provided by Lavor.

$$\begin{bmatrix} x_{n_1} \\ x_{n_2} \\ x_{n_3} \\ 1 \end{bmatrix} = B_1 B_2 ... B_n \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (n = 1, ..., N) \tag{1}$$

The matrices B_n for n = 1 to N, the number of atoms along the chain of atoms, are defined in Equation 2, also from Lavor[14].

$$B_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad B_2 = \begin{bmatrix} -1 & 0 & 0 & -r_{12} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$B_3 = \begin{bmatrix} -\cos\theta_{13} & -\sin\theta_{13} & 0 & -r_{23}\cos\theta_{13} \\ \sin\theta_{13} & -\cos\theta_{13} & 0 & r_{23}\sin\theta_{13} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

$$B_i = \begin{bmatrix} -\cos\theta_{(i-2)i} & -\sin\theta_{(i-2)i} & 0 & -r_{(i-1)i}\cos\theta_{(i-2)i} \\ \sin\theta_{(i-2)i}\cos\omega_{(i-3)i} & -\cos\theta_{(i-2)i}\cos\omega_{(i-3)i} & -\sin\omega_{(i-3)i} & r_{(i-1)i}\sin\theta_{(i-2)i}\cos\omega_{(i-3)i} \\ \sin\theta_{(i-2)i}\sin\omega_{(i-3)i} & -\cos\theta_{(i-2)i}\sin\omega_{(i-3)i} & \cos\omega_{(i-3)i} & r_{(i-1)i}\sin\theta_{(i-2)i}\sin\omega_{(i-3)i} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

For this simulation, the negatively-charged oxygen atom at the C-terminus of the protein was considered the origin. This allows these equations to be applied to the planar bond in the carboxylate functional group. When any atoms branch off of the backbone, they are treated as a separate chain of matricies. The program continues until the N-terminus is reached.

For this simulation, the bond lengths and bond angles are held constant, as well as the dihedral angles of atoms relative to each other. This constricts the output of the simulation to more realistic geometries.

Protein Energy Function

The energy of a protein is calculated as a composite of two major components, each of which can be divided into even smaller terms. Overall, the minimum energy can be found as the minimum of the sum of the protein's internal energy and the solvation energy of the protein.

Internal Energy

The internal energy of the molecule was calculated by using the third generation of the Empirical Conformational Energy Program for Peptides, or ECEPP/3, which uses several values that have been determined and verified experimentally to provide electrostatic forces, Lennard-Jones forces, and hydrogen bonded forces[15]. The energy terms are calculated as shown in Equation 3, which is obtained from Klepeis et al[16].

$$E = \underset{\text{(Electrostatic)}}{\sum_{(ij)\in ES} \frac{q_i q_j}{r_{ij}}} + \underset{\text{(Nonbonded)}}{\sum_{(ij)\in NB} F_{ij} \frac{A_{ij}}{r_{ij}^{12}} - \frac{C_{ij}}{r_{ij}^{6}}} + \underset{\text{(Hydrogen bonded)}}{\sum_{(ij)\in HX} \frac{A'_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^{10}}} \qquad (3)$$

The electrostatic potentials[17], as well as the constants A, C, A', and B[18,19], have been measured and quantified for each type of atom in the twenty naturally occurring amino acids, allowing the program to call from these values and determine the energy value for any given atom pair based on the coordinates in the Cartesian plane found earlier. The interatomic radii can be found by applying the Pythagorean Theorem in three-dimensional space.

Electrostatic Solvation Energy

Solvation energy is a fundamental issue in biomolecular modeling[20]. There are two generally accepted methods of modeling the solvation energy. The first method, explicit solvation, functions by simulating the presence of a large amount of water molecules surrounding the protein of interest, which can be a very computationally intensive process that contains an excessive number of degrees of freedom. Intrinsic solvation modeling, on the other hand, treats the water as a mathematical continuum and simplifies the interaction between the solvent and the protein solute into much simpler equations[21].

Implicit solvation energy is composed of two parts: a nonpolar term and an electrostatic term as indicated in Equation 4. The nonpolar term, ΔG_{np}, consists of the energetic cost of displacing the water atoms and providing a phase boundary, while the electrostatic term, ΔG_{elec}, accounts for the interactions between charged atoms on the solute and the solvent[20].

$$\Delta G_{solvation} = \Delta G_{np} + \Delta G_{elec} \tag{4}$$

In order to calculate the electrostatic term of the solvation energy, Tjong et al. developed a generalized Born model of the Poisson-Boltzmann equation (Eq. 5), which is a very accurate but computationally intensive method of determining this solvation energy.

$$\nabla \cdot \varepsilon(r) \nabla \phi(r) = -4\pi\rho(r) - 4\pi \sum_i e z_i c_{io} \exp\left[\frac{-e z_i \phi(r)}{k_B T}\right] \tag{5}$$

where ε is the dielectric constant, ϕ is the electrostatic potential, and ρ is the solute charge density. All of these properties are function of the position vector r. The second term on the right-hand arises from the Boltzmann distribution of mobile ions in the solvent (k_BT is the thermal energy) [20]. Ionic species *i* have valency z_i and bulk concentration c_{io}. Equation four can be linearized, yielding the following result:

$$\nabla \cdot \varepsilon(r) \nabla \phi(r) - \frac{8\pi e^2 I \phi(r)}{k_B T} = -4\pi\rho(r) \tag{6}$$

As an alternative to the computationally intensive Poisson-Boltzmann equation, Tjong et al. have found that a generalized Born formalism, specifically the GBr^6 model, can closely approximate the accurate results of the linearized Poisson-Boltzmann equation at a significant decrease in computational cost. The Born formalism is inspired by the Born formula (Equation 7) for the solvation energy of ions.

$$\Delta G_{elec} = \left(\frac{1}{\varepsilon_i} - \frac{1}{\varepsilon_s}\right)\frac{Q^2}{2R} \tag{7}$$

In Equation 7, R is the radius of the spherical solute, ε_i is the dielectric constant of the solute, Q is the charge of the solute, and ε_s is the dielectric constant of the solvent.

The Born formula can be modified slightly for use in spherical proteins as shown in Equation 8, where f_{ij} is a function of the distance r_{ij} between atoms, and q_i and q_j are the charges of two atoms of interest *i* and *j*.

$$\Delta G_{GB}^{o} = \left(\frac{1}{\varepsilon_i} - \frac{1}{\varepsilon_s}\right)\frac{q_i q_j}{2 f_{ij}} \tag{8}$$

The term f_{ij} can be calculated as a function of the distance between atoms *i* and *j* and the Born radii of said atoms as shown in Equation 9. The Born radii (B_i and B_j) refer to the degree by which atoms are buried inside the protein molecule; atoms buried deeper within the molecule contribute less to the electrostatic energy. This reflects the physical ability of atoms on the exposed surface of proteins to participate in more profound interactions with the surrounding solvent.

$$f_{ij} = \left[r_{ij}^2 + B_i B_j \exp\left(\frac{-r_{ij}^2}{4 B_i B_j} \right) \right]^{1/2} \tag{9}$$

The GBr^6 model used in this study differs from other generalized Born methods in that the Born radius (B_i) is approximated with an r^6 expression as shown in equation 10, instead of using a Coulomb field approximation, which usually gives as much as 100% error in the solvation energy[22,23].

$$\frac{1}{B_i^3} = \frac{3}{4\pi} \int_{solvent} \frac{d^3 r}{(r - r_i)^6} \tag{10}$$

In the above expression, r_i is the location of the ith atom, and the limits of integration cover the infinite solvent dielectric. Equation 10 gives rise to the name of the model since the Born radii is approximated with an expression raised to the sixth power.

The GB r^6 also differs from other generalized Born methods because they do not have additional dependence on solute and solvent dielectric constants beyond the factor $\frac{1}{\varepsilon_i} - \frac{1}{\varepsilon_s}$ already present in the Born formula,[23] while this model incorporates a scaling term f from the Poisson-Boltzmann model as shown in equation 11.

$$\Delta G_{GB} = \Delta G_{GB}^{0} f(\epsilon_i/\epsilon_s) \tag{11}$$

The scaling term f is found by equation 12, where the parameters *A* and *B* are defined by Tjong et al. based on the net charge of the solute and the number of atoms in the solute.

$$f\left(\frac{\epsilon_i}{\epsilon_s}\right) = \frac{A + 2B\epsilon_i/\epsilon_s}{1 + 2\epsilon_i/\epsilon_s} \tag{12}$$

Finally, salt effects have not been considered in previous GB models, so the GB r^6 model fully and accurately accounts for salt effects by modifying equation 8 as shown in equation 13,

$$\Delta G^0_{GB} = -\sum_{i,j}\left[\frac{1}{\epsilon_i} - \frac{exp(-\alpha\kappa f_{ij})}{\epsilon_s}\right] q_i q_j/2f_{ij}] \qquad (13)$$

in which a scaling factor α is introduced and $\kappa^2 = 8\pi e^2 I/\epsilon_s k_B T$ where I indicates the ionic strength. Equations 8 – 13 combine to provide a close estimate to the electrostatic term of the solvation energy as evidenced by the closeness of results to the Poisson-Boltzmann equation[20].

Nonpolar Solvation Energy

The nonpolar energy of solvation, ΔG_{np}, is found as a composite of the energies of each individual atom based on the exposure of these atoms to water as shown in equation 14, in which VHS_i indicates the volume of the hydration shell for atom *i*[24].

$$\Delta G_{np} = \sum_i \delta_i (VHS_i) \qquad (14)$$

The concept of a hydration shell can be better illustrated using Figure 5, where the grey area indicates the volume of the hydration shell itself and the white area shows the volume occupied by the amide. The van der Waals radii of all atoms near the atom in question (in this case, the amide hydrogen) are used to approximate the volume occupied by the solute, and the overlap of this volume with the sphere of radius R_H^h, the hydration radius of hydrogen, must be subtracted to obtain the proper volume of the hydration shell.

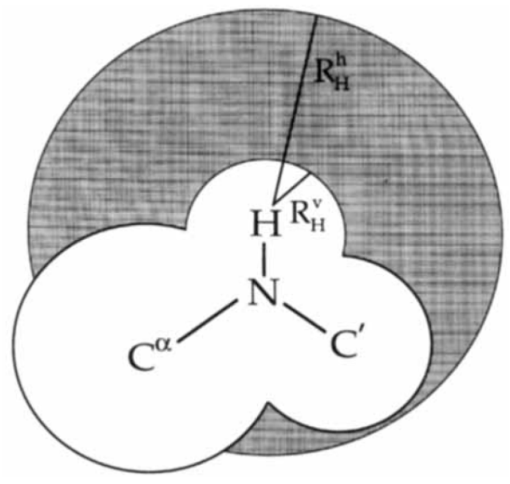

Figure 5: The exposed volume of the hydration shell about the H atom is shaded in. Not shown are the hydration shells for the nitrogen or carbon atoms.

Augsperger et al.[24] developed a method to approximate the volume of the hydration shell using Equation 15, which accounts for the volume of intersection between two spheres and takes advantage of a reduced van der Waals radius R_i^r[24]. The van der Waals radius is reduced to adjust for the error implicit in only determining the intersection volume of double overlap in a series of spheres.

$$(VHS)_i \approx \frac{4\pi}{3}\left(R_i^{h^3} - R_i^{v^3}\right) - \sum_{j\neq i}\left[D\left(r_{ij}; R_i^h, R_j^r\right) - D\left(r_{ij}; R_i^h, R_j^r\right)\right] \qquad (15)$$

The function $D(r_{ij}; R_i^h, R_j^r)$ is defined in equation 16. This function serves to measure the volume of the double-sphere overlapping sections between atom *i* and the reduced radii of all other atoms *j*. This equation is derived from using a hard-shell model to represent the atoms, such that they have a constant and uniform density within the van der Waals radii.

$$D(r_{12}; R_1, R_2) = \frac{2\pi}{3}\left(R_1^3 + R_2^3 + \frac{r_{12}^3}{8}\right) - \frac{\pi r_{12}}{2}(R_1^2 + R_2^2) - \frac{\pi}{4r_{12}}(R_1^2 - R_2^2)^2 \qquad (16)$$

Equation 16 holds true when $|R_1\text{-}R_2| \leq r_{12} \leq R_1\text{+}R_2$. The constant parameters δ_i were found for each unique atom and applied to the hydration shell for each atom to obtain the net approximation of the nonpolar solvation energy.

Optimization Methods

There are several different methods that can be used to minimize the energy function. Only the torsion angles are optimized in this program, with the bond lengths, bond angles, and relative dihedrals among nearby atoms held constant. The data for these bond lengths are taken from the protein database.

This program runs simpler modes of numerical optimization on polypeptide chains that are built sequentially from the N-terminus, similar to the manner in which it is manufactured in the ribosome. As each smaller polypeptide sequence is optimized, the next amino acid residue as added to the optimized chain and the chain is optimized again. This method employs the gradient search to attempt to mimic the effects of the intramolecular forces as they apply to each atom in a polypeptide. Because the derivative of energy with respect to distance is, by definition, the force on a body, the gradient search was chosen in the hope that it could provide a pseudo-dynamic simulation of the polypeptide chain growing and could still converge to provide a realistic structure.

Gradient Search

The gradient with respect to each torsion angle is determined by a forward-order difference equation with a step size of 0.1 degrees. To run this equation, the energy must be computed twice by the software and divided by the step size. Once the gradient is found, Equation 17 is applied to the system where the constant γ is the step size for each iteration.

$$\omega_{n+1} = \omega_n - \gamma_n \nabla E(\omega_n) \qquad (17)$$

This method has the downside of being a very crude method of optimization. It is notorious for taking a very long time to converge, even for smooth surfaces. Over the energy landscape, which is expected to be a very non-uniform environment, using the wrong step size in the gradient search runs the risk of moving to a state of higher energy as it overshoots the minimum. On the opposite end of the spectrum, a step size that is too small could lead to premature convergence.

Golden Search

To reduce computation time, the value of γ was also optimized in between every determination of the energy gradient. The Golden Search method was used to optimize the step size and find the local minimum in the direction of the gradient. However, this was found to lead to premature convergence to an unrealistic local minimum, and this method was abandoned in favor of a constant, small step size.

Genetic Algorithm

The conformation that results from a unique set of dihedral angles constitutes a chromosome, while the angles within the set are considered genes. The total potential energy calculated for a given conformation is used to rank its 'fitness' where a lower energy is considered 'more fit.'

Initialization

Initially, the user inputs the amino acid sequence of interest as a one letter code. The mutation rate and selection rate are also input. Then, the program randomly generates the genes (acceptable angle values are between -180 and 180 degrees) to define a unique chromosome. Because each member of the population is unique, the fitness of each member will vary randomly resulting in a population that spans the energy landscape.

Selection

Using the energy functions described earlier, the potential energy of each conformation is calculated and ranked in order of fitness. The best members of the population are selected to produce offspring and the rest are killed off. Selection occurs during each iteration to allow the population to evolve over the generations to favor the fit members.

Reproduction

Reproduction is simulated with crossovers. In a crossover, an offspring is the result of a combination of two parents. In this genetic algorithm, crossovers can occur via one of two methods. The simplest is the uniform crossover where the program goes down the chromosome and determines randomly which parent will provide the angle value, as indicated in Figure 6. Each colored bar is simply a visual representation of an individual torsion angle.

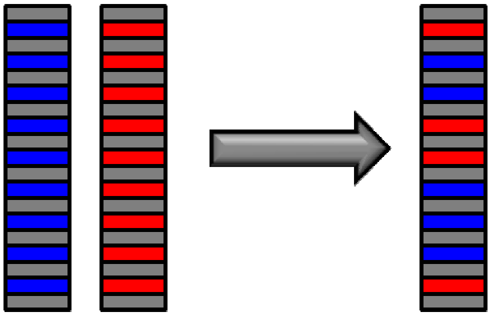

Figure 6. Uniform Crossover

The blending method uses crossovers in a way that also increases the diversity of the population. In this method, the angle values of the parents are used as boundary conditions and a value between that of the parents is selected. This is governed through the use of a blending factor, β. The blending factor is a random number between 0 and 1 called for each gene in the offspring's chromosome such that a number close to 0 would provide a value closer to the father's value while a number close to 1 would pass on a value closer to the mother's value.

$$p_{offspring} = \beta * p_{mother} + (1 - \beta) * p_{father} \tag{18}$$

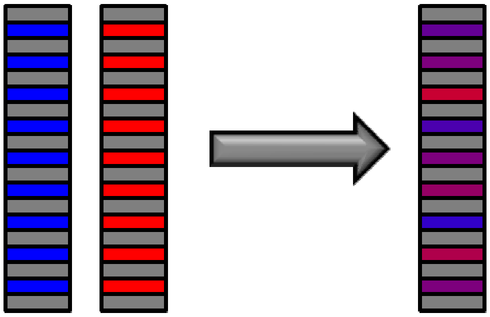

Figure 7. Blending Crossover

Mutation

Care must be taken so that the program does not converge too quickly into a local minimum. Therefore, it is necessary to further increase the diversity of the population in order to explore more possibilities. This is accomplished though mutation where existing angles are chosen to be randomly mutated by changing their values to between -180 and 180 degrees. Of course, this could also ruin any of the more fit chromosomes retained from previous generations.

To prevent the loss of beneficial traits, the population is first cloned and the mutations are applied to the copies.

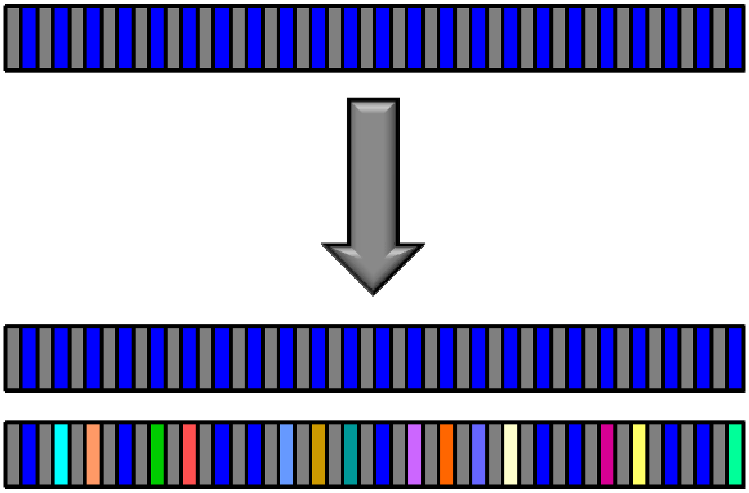

Figure 8. Cloning Mutation

Termination

The process continues in a loop of selection, crossover, and mutation until a solution is obtained. The criterion for this program is when 80% of the population is within a specified deviation, the program has converged and exits. The program stores the 50 best chromosomes of the final iteration. Usually, the top ten of these 50 are subjected to visual inspection via molecular imaging software including the Tinker/FFE package and Accelry's DS visualizer.

Local Optimization Results

Computation Time

Initial benchmarking of the software measured the time required to run the program for different inputs. The run time required to process a polypeptide grows exponentially; a 15-residue polypeptide requires about two and a half hours, while a 19-residue polypeptide requires sixteen hours. Running the amyloid β-peptide, which has forty residues, requires a full week for a single simulation. Figure 9 shows a graph of some completed simulations with run times included. This exponential trend makes logical sense, as the calculations run for one residue must be repeated with each new addition to the chain, and there is no criterion to determine when a residue's contribution to the overall energy becomes negligible.

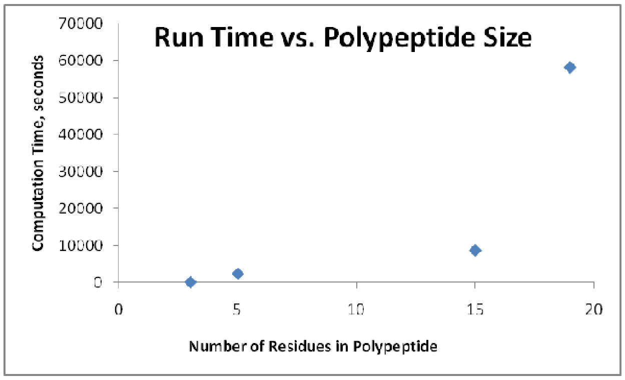

Figure 9: Due to the nature of the simulation, the time required increases dramatically.

Tyr-Gly-Gly-Phe-Met

The pentapeptide YGGFM has a well-known structure in the Protein Database, obtained from NMR measurements. However, due to its small size, it also has a large degree of variability as shown in Figure 10, although a close examination of the left image in Figure 10 shows that the side chains have some reliable approximate locations: when viewed from the perspective that the C-terminus is facing downward with the methionine side chain on its left, the phenylalanine side chain should be in the center-right of the image, and the highly variable glycine residues will curve the amino acid such that the tyrosine appears to be slightly behind the methionine side chain.

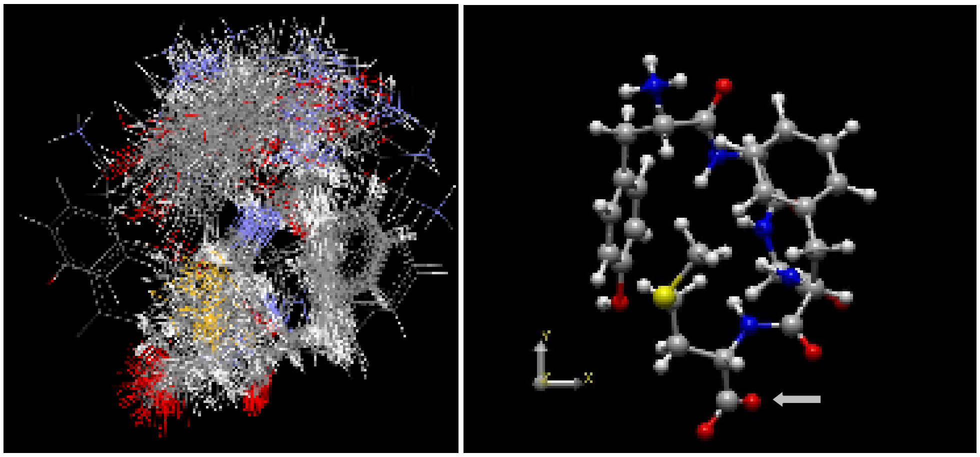

Figure 10: The conformation of YGGFM is widely variable, as seen by the image on the left. On the right, a snapshot of the average structure with the noise removed. The arrow points at the C-terminus.

Figure 11 shows the results run from three different simulations: using the gradient descent with no solvation energy term, using the gradient descent with a solvation energy term, and using the line search method with the solvation energy term. Run times varied from about 2500 – 5000 sec for these experiments.

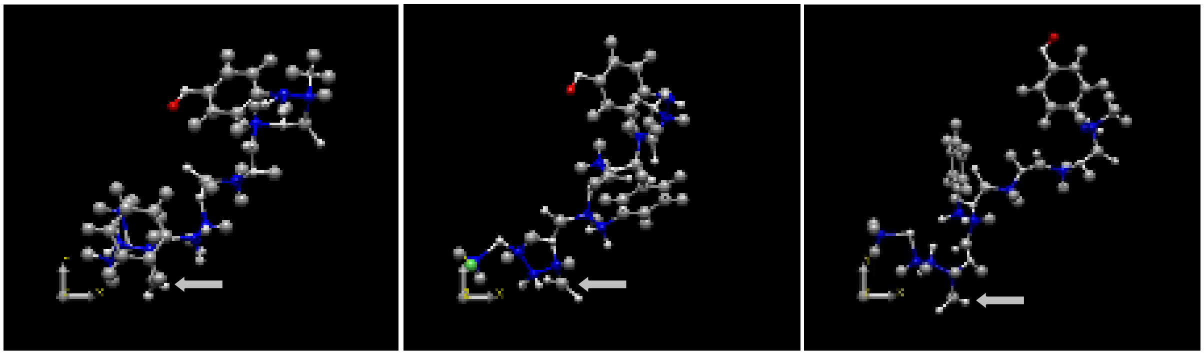

Figure 11: The results from the simulation run without solvation energy (left), with solvation energy (center), and with the line search and solvation energy (right) show little resemblance to the shapes shown in Figure 9. The arrow in all cases points to the C-terminus.

The snapshots in Figure 11 show that none of these optimization methods are correctly predicting the structure of the polypeptide being simulated. The polypeptide chain, as it grows, is failing to fold sufficiently to appear similar to the known structure of the polypeptide of interest. Adding solvation energy appears to cause the protein to fold inwards on itself slight more, but this still does not match the desired output; the phenylalanine and tyrosine residues fail to reach the desired regions of space that the original image predicts. Additionally, the peptide bonds do not show the desired planar conformation with a trans configuration. Because this conformation is most commonly seen in nature, this suggests that the simulation is not successfully modeling the torsion angles affecting these peptide bonds. Simulations where the peptide bond plane was fixed failed to converge to realistic structures.

Amyloid β-peptide

The amyloid β-peptide was measured by M. Coles et al. and is provided publicly in the protein data bank. This polypeptide, which is present in the brain and believed to play a role in the development of Alzheimer's disease, is forty amino acids long and has the sequence DAEFRHDSGY EVHHQKLVFF AEDVGSNKGA IIGLMVGGVV. In addition to

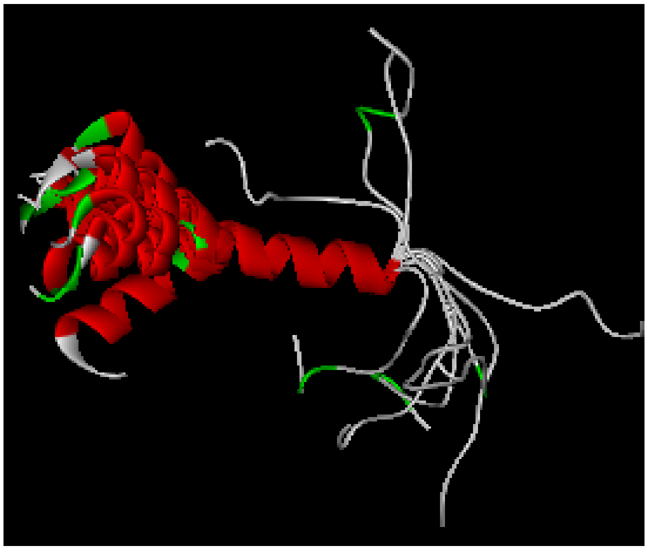

Figure 12: The structure of the Amyloid β-peptide, as measured using NMR techniques and displayed in ribbon form. This image shows eight possible models superimposed on one another, with the helix in red.

its comparatively small size, this polypeptide was chosen for its recognizable α-helix that is present on the C-terminal side of the chain. As seen in Figure 12, the N-terminal side is a highly variable region with no recognizable secondary structure. This polypeptide was also selected for use due to its lack of cysteine residues, removing the need to account for disulfide bonds.

To ascertain whether the program holds promise to predict secondary structure or higher levels, the first criterion it must achieve is to develop an α-helix, which is a fundamental unit of secondary structure.

When growing the polypeptide chain with local optimization, it is important to note that this method uses a deterministic starting value that is chosen to prevent the program from having a bias towards a certain type of secondary structure. If the gradient search fails to change the initial values significantly, a repeating pattern as seen in Figure 13 will be seen, with the planar peptide bonds forming a repetitive motif. This pattern does not commonly occur in nature, and retention of this form after convergence implies an optimizer failure. Figure 14 shows the results of the growing chains simulation applied to this polypeptide.

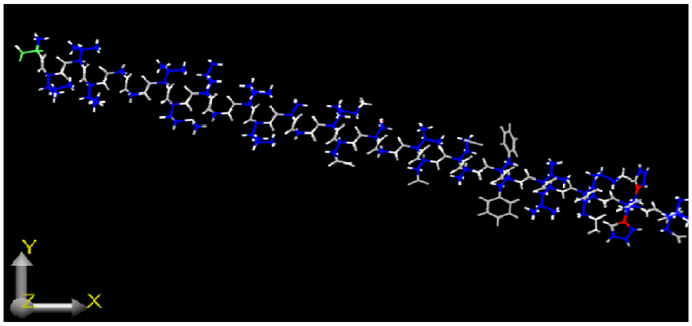

Figure 13: The cyclic motif along the polypeptide backbone indicates that the gradient search did not alter the initial angles, suggesting optimizer failure.

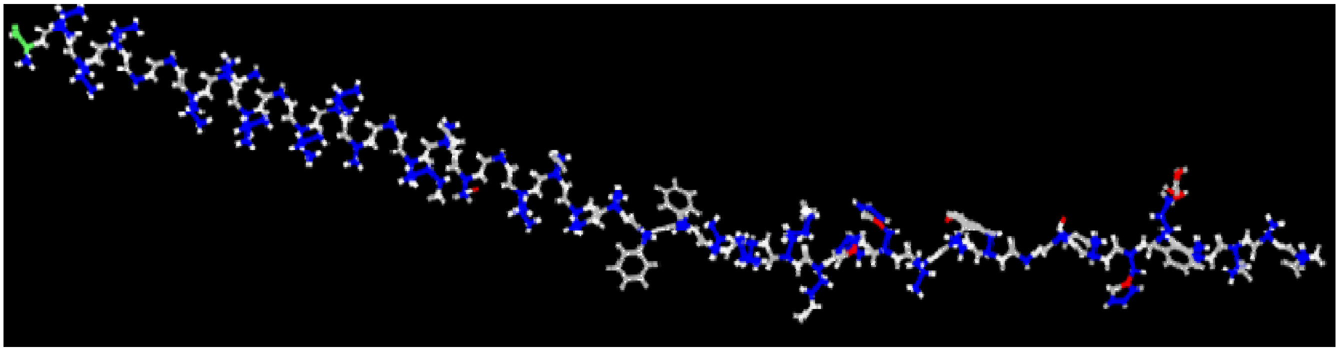

Figure 14: The predicted structure of the amyloid β-peptide using the gradient search method. To help orient the image, the C-terminus is on the left side of the picture and is highlighted green.

Figure 14 contains very revealing information. Fully half of the polypeptide was unchanged by the gradient search optimization method, suggesting that the perturbation each new amino acid residue introduces to the energy landscape becomes negligible as the polypeptide chain grows larger, causing the gradient descent to remain in the same local minimum energy well. Comparing the N-terminus is less indicative of the quality of this simulation simple because it is the more variable region in the protein's structure, however Figure 14 shows that the N-terminal side of the chain is still mostly linear with little to no bending. This is problematic, as the hydrophobic effect should be driving at least some protein folding, and the lack of this process suggests the method is wholly unsuitable for larger polypeptide chains.

Summary

The use of the gradient descent coupled with the simulated growth of the polypeptide chain appears to be unsuitable for use in protein folding predictions. In addition to the low-quality results shown here, the gradient descent method has a tendency to converge to unrealistic structures, with an unreasonable degree of overlap between atoms and large energy values greater than those found in certain iterations. Furthermore, at times it skips over the minimum values entirely. This may be because the gradient of the energy, which approximates the forces, is alone not enough to simulate the folding of a protein; without accounting for the acceleration and velocities of these atoms it is an inaccurate representation of natural phenomena.

Genetic Algorithm Results

Population Size and Mutation Rate

The following graph details the rate of convergence and level of optimization achievable for varying population sizes. Higher population sizes increase the rate of convergence and allow the program to find lower energy conformations. However, these benefits come at the cost of rapidly increasing computational time as seen in figure 15 on the following page.

Figure 16 illustrates the effect of changing the mutation rate on the convergence characteristics. Increasing the mutation rate can allow the program to converge at a faster rate and reach a better optimum. However, the benefits of increasing the mutation rate reach an optimum at around 60%. Afterwards, beneficial traits are not preserved enough and the program becomes similar in nature to a monte carlo simulation that is preserving the best results as it chances across them.

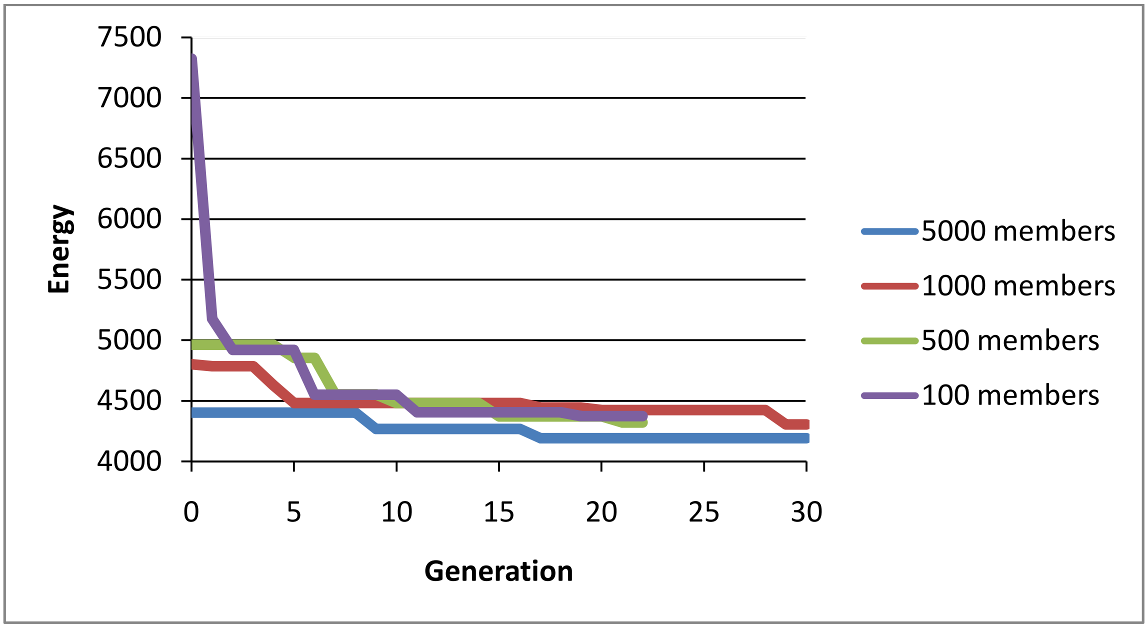

Figure 15. Convergence characteristics at different population sizes

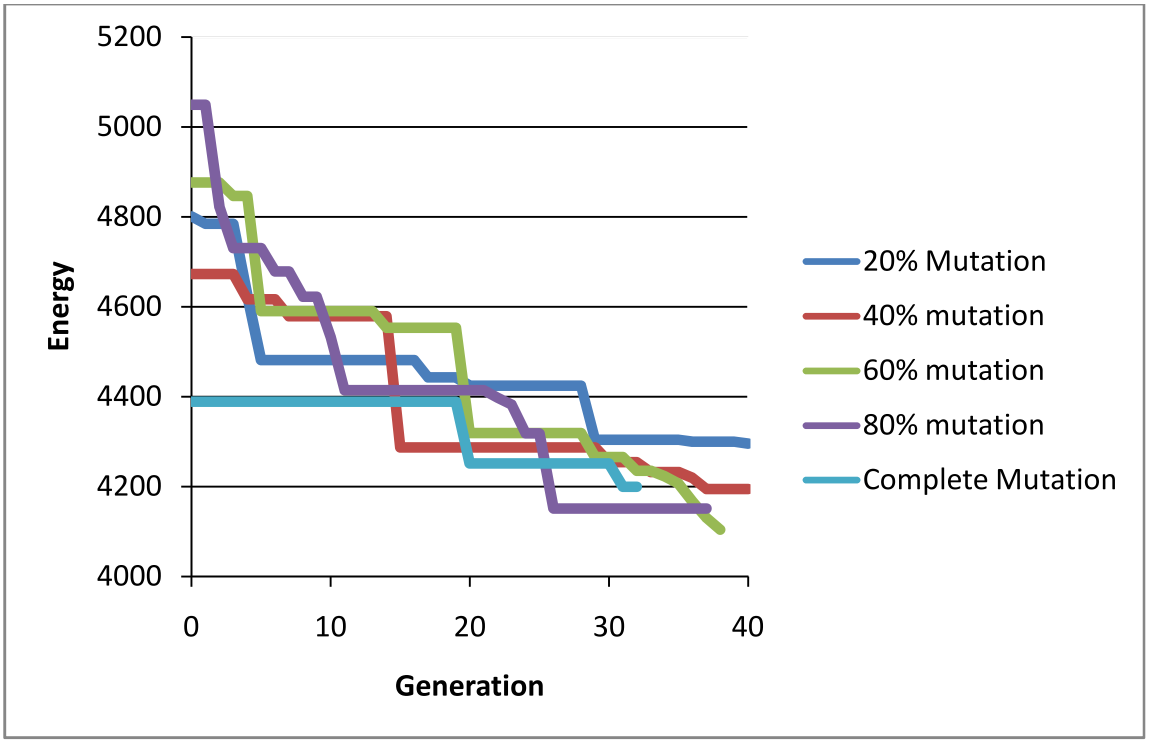

Figure 16. Effect of mutation rate on convergence characteristics

Tyr-Gly-Gly-Phe-Met

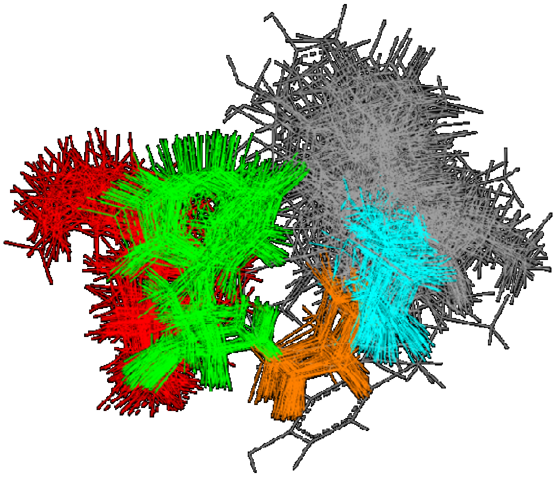

Figure 17. The NMR structure of YGGFM, colored by amino acid residue as follows: Gray = Tyr1, Cyan = Gly2, Orange = Gly3, Green = Phe4, Red = Met5.

To compare the genetic algorithm results with the NMR structures of Met-Enkephalin, it is important to use figure 17, which contains the same NMR structures as shown in figure 10, from a different perspective to better show each individual amino acid residue.

Several simulations were run to attempt to predict the structure of this pentapeptide, with common trends emerging among the best results. Figure 18 shows one representative simulation output and its fit with the NMR structures measured experimentally.

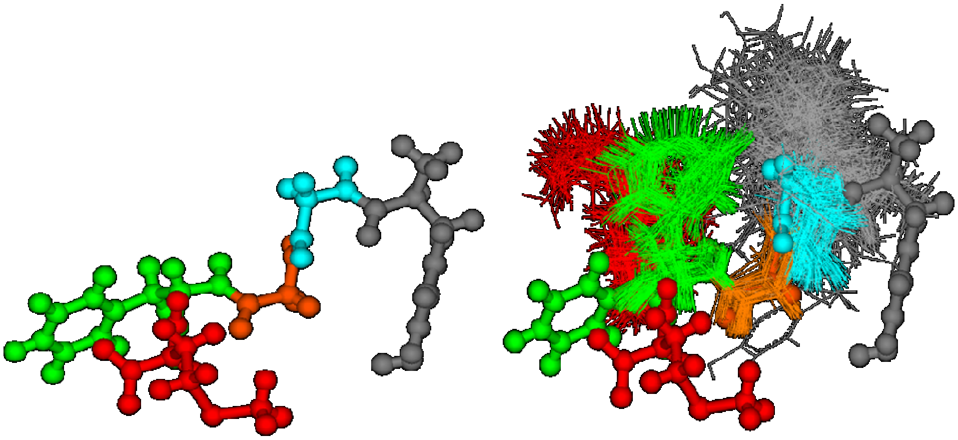

Figure 18: The left image shows the program output, rendered in the ball-and-stick model to facilitate comparison with the variable NMR structure. When the backbone is aligned, there is decent agreement, but the bulky side chains do not align well with experimental measurements.

Figure 18 indicates that the results are certainly not perfect; the bulky phenol groups on the side chains are nowhere near the large range of possibilities permitted by the NMR models. However, the backbone aligns decently well for the first four amino acids; unfortunately, the misplacement of the phenylalanine side chain prevents the methionine residue from moving towards the correct location in space.

Figure 19 shows another representative result that appears to match the natural configuration slightly better.

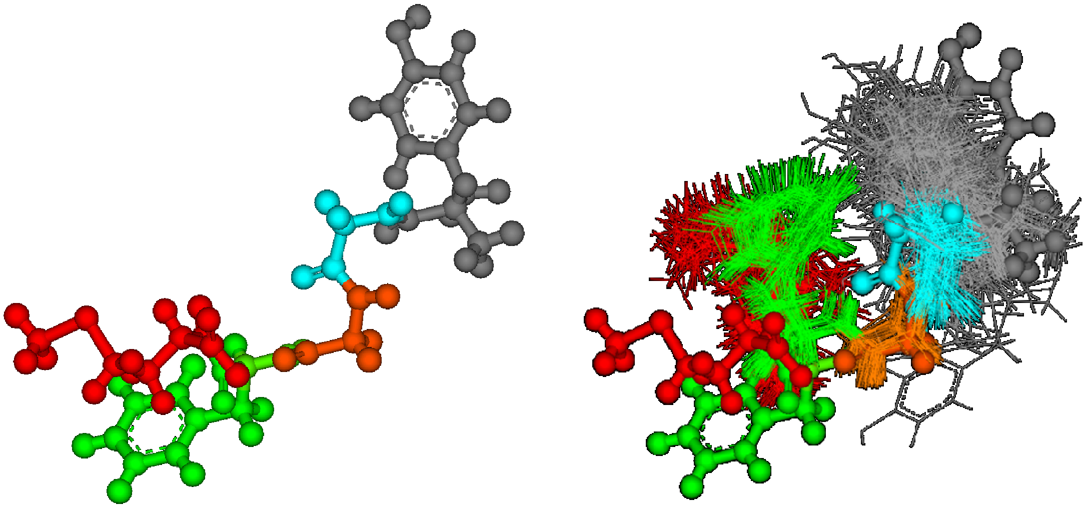

Figure 19: A second example output for the Met-Enkephalin predictions.

In neither case do the predictions align perfectly with the NMR models. The best case scenario is to see a close fit to the first four amino acid residues, but the methionine is typically in the wrong position completely.

When the genetic population is "polluted" with preset members that are identical to the measured NMR structures, these members actually find themselves being out-competed by these less accurate models. This suggests that the genetic algorithm itself is performing its job admirably. The problem appears to reside in the parameters used to calculate the conformational energy itself; the optimum energy found by the algorithm is significantly lower than those that the energy calculations determine for the native structures.

This suggests that the genetic algorithm is a robust method for analyzing the effectiveness of different force fields. In this study, the ECEPP/3 force field was used, but there are several other sets of parameters such as CHARMM and AMBER that can be comparatively evaluated.

Amyloid β-peptide

The genetic algorithm results for the amyloid β-peptide revealed a larger discrepancy than those for the met-enkephalin pentapeptide, indicating that the error and inconsistencies from the model propagate to a significant degree in larger chains. Figure 20 shows the structure of one of the eight structures of this polypeptide measured using NMR to provide as a basis for comparison. Figures 21 and 22 show how well the algorithm's predictions compare to the actual structure.

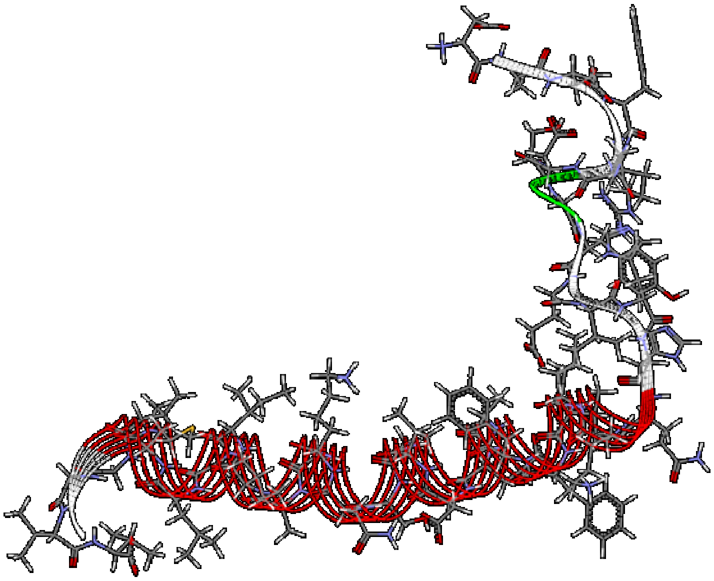

Figure 20: The ribbon model is superimposed on a wireframe diagram to show the actual positions of the atoms and how they fit into the α helix. The C-terminus is on the bottom left side of this structure.

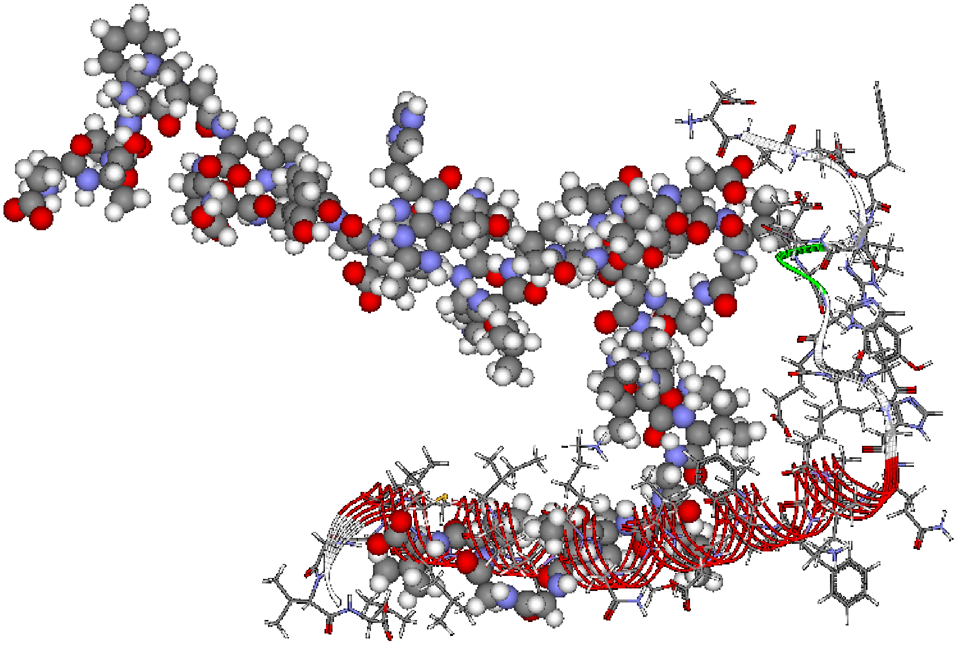

Figure 21: The predicted structure is shown using a space-filling representation, superimposed over the image in figure 20. The images were aligned at the C-terminal end to see how closely they match up. No characteristics of the α helix are manifest in the predicted model.

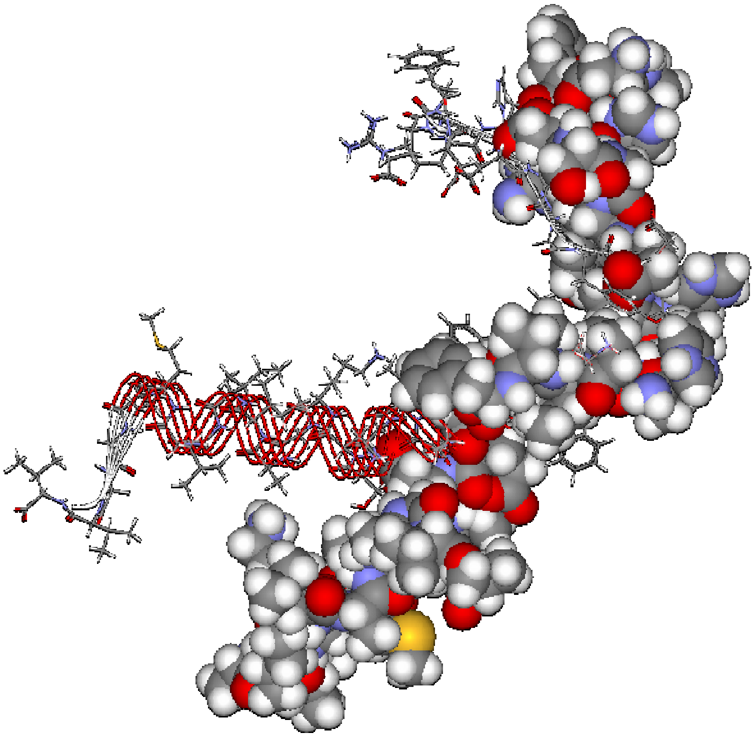

Figure 22: The closest-matching NMR model (wireframe) to this predicted structure (solid atoms) is slightly different to the image in figure 20. In this image, the molecules are aligned by their centers of geometry. However, there is still no α-helical structure on the C-terminal side of the predicted structure.

The predicted structures are hugely inaccurate. As stated previously, variability in the free N-terminus is to be expected, while the α helix should reliably form. However, in none of these cases did the α helix arise naturally from simulation. In fact, more often than not, the backbone amide hydrogen atoms were not even aligned in the proper direction to participate in hydrogen bonding with the backbone carbonyl oxygen atoms. With no repeatable helical pattern, there is no reason to assume that the energy calculations can successfully determine this optimal configuration that occurs frequently in nature.

Conclusions

The use of the gradient search as a local optimization technique has proven to be woefully inadequate when compared to the genetic algorithm. In terms of the reliability of the results, the accuracy of results, and the computation time, the genetic algorithm outperforms the use of the gradient search. This is not to disqualify the use of the simulated growth of the polypeptide chain completely, however; there is a good possibility that an alternate local optimization method that is more well-suited to rougher landscapes may reach closer to the mark. Unfortunately, this will still not correct for the velocities and accelerations of the molecules at any given point, so a large amount of work will need to be done to reach this point. The genetic algorithm, however, is capable of successfully reaching the optimal thermodynamic conformation as evidenced by its repeatable convergence to the same range of values. Further study should focus on the genetic algorithm as the optimization method and analyze other energy parameters and force fields to find the most accurate method to model the natural forces in the protein molecules.

References

1. Voet, Donald, Judith G. Voet and Charlottle W. Pratt. *Fundamentals of Biochemistry*. 3rd ed. USA: 2008.
2. Baker, David and Andrej Sali. "Protein Structure Prediction and Structural Genomics." Science 294 (5540), 93. (5 October 2001) [DOI: 10.1126/science.1065659]
3. William Ramsay Taylor. "Identification of protein sequence homology by consensus template alignment." *Journal of Molecular Biology*, Volume 188, Issue 2, 20 March 1986, Pages 233-258. (http://www.sciencedirect.com/science/article/B6WK7-4DN3YG4-4M/2/9bb15110c7e39a748e8700247ca314a2)
4. Guex, Nicolas and Manuel C. Peitsch. "SWISS-MODEL and the Swiss-Pdb Viewer: An environment for comparative protein modeling." *Electrophoresis.* Volume 18, Number 15. © 1997, p. 2714-2723.
5. Duan, Yong and Peter A. Kollman. "Pathways to a Protein Folding Intermediate Observed in a 1-Microsecond Simulation in Aqueous Solution." *Science* **282** (5389), 740.
6. Kihara Daisuke, Hui Lu, Andrzej Kolinski, and Jeffrey Skolnick. "TOUCHSTONE: An ab initio protein structure prediction method that uses threading-based tertiary restraints." PNAS 2001 98:10125-10130.
7. Phillips, A.T., J.B. Rosen, and V.H. Walke. "Convex global underestimation for molecular structure prediction." From Local to Global Optimization, 2001.
8. Thomas Dandekar, Patrick Argos, Folding the Main Chain of Small Proteins with the Genetic Algorithm, Journal of Molecular Biology, Volume 236, Issue 3, 24 February 1994, Pages 844-861, ISSN 0022-2836, DOI: 10.1006/jmbi.1994.1193.

9. Bäck, Thomas, Gunter Rudolphy, and HansPaul Schwefel. “Evolutionary Programming and Evolution Strategies: Similarities and Differences.” University of Dortmund, Dept of Computer Science.
10. Holland, John. *Adaptation in Natural and Artificial Systems*. 1992.
11. Haupt, R. L. *Practical Genetic Algorithms.* 2004.
12. Thompson, Bradford H. “Calculation of Cartesian Coordinates and their Derivatives from Internal Molecular Coordinates.” *Journal of Chemical Physics*. Volume 47, Number 9. 1967: p. 3407-3410.
13. Trigub, L. P. and Yu. A. Kruglyak. “A universal program for calculating the Cartesian coordinates of atoms in molecules.” *Journal of Structural Chemistry.* Volume 24, Number 1. 2004. p. 161-164.
14. Lavor, Carlile. “On Generating Instances for the Molecular Distance Geometry Problem.” *Nonconvex Optimization and Its Applications* 84. 2006, p. 405-414.
15. Zimmerman, S. Scott, Marcia S. Pottle, George Némethy, and Harold A. Scheraga. “Conformational Analysis of the 20 Naturally Occurring Amino Acid Residues using ECEPP.” *Macromolecules* Volume 10, Number 1. 1977: p. 1-9.
16. Klepeis, J. L. and C. A. Floudas. “Free energy calculations for peptides via deterministic global optimization.” *Journal of Chemical Physics* Volume 110, number 15. 1999: 7491 – 7512.
17. Momany, F.A., R. F. McGuire, A. W. Burgess, and H. A. Scheraga. “Energy Parameters in Polypeptides. VII. Geometric Parameters, Partial Atomic Charges, Nonbonded Interactions, Hydrogen Bond Interactions, and Intrinsic Torsional Potentials for the Naturally Occurring Amino Acids.” *The Journal of Physical Chemistry*, Vol. 79, No. 22, 1975. p. 2361 – 2381.
18. Némethy, George, Kenneth D. Gibson, Kathleen A. Palmer, Chang No Yoon, Germana Paterlini, Adriana Zagari, Shirley Rumsey, and Harold A. Scheraga. “Energy Parameters in Polypeptides. 10. Improved Geometrical Parameters and Nonbonded Interactions for Use in the ECEPP/3 Algorithm, with Application to Proline-Containing Peptides.” *J. Phys. Chem.* 1992, 96, 6472-6484
19. Némethy, Geroge, Marcia S. Pottle, and Harold A. Scheraga. “Energy Parameters in Polypeptides. 9. Updating of Geometrical Parameters, Nonbonded Interactions, and

Hydrogen Bond Interactions for the Naturally Occurring Amino Acids." *J. Phys. Chem.* 1983, 87, 1883-1887.

20. Tjong, Horatio, and Huan-Xiang Zhou. "GBr6: A Parameterization-Free, Accurate, Analytical Generalized Born Method." *J. Phys. Chem*. 111(2007): 3055-3061.
21. Zhou, Ruhong. "Free energy landscape of protein folding in water: Explicit vs. implicit solvent." Proteins: Structure, Function, and Genetics. Vol 53, 2. 148-161.
22. Hawkins, G. D.; Cramer, C. J.; Truhlar, D. G. J. *Phys. Chem.* **1996,** *100,*19824.
23. Schaefer, M.; Karplus, M. *J. Phys. Chem.* **1996**, *100*, 1578.
24. Augspurger, Joseph D. and Harold A. Scheraga. "An Efficient, Differentiable Hydration Potential for Peptides and Proteins." *Journal of Computational Chemistry*, Vol. 17, No. 13, 1549-1558 (1996).
25. Lu, Rufei, Lauren Yarholar, Warren Yates, and Dr. Miguel Bagajewicz. "Protein Folding Predictions." The University of Oklahoma, 2008.