Protein Folding Prediction

Rufei Lu

Lauren Yarholar

Warren Yates

Dr. Miguel Bagajewicz

The University of Oklahoma

March 14,2008

Abstract

The search for an accurate and efficient protein conformation predicting method started around the end of the twentieth century with Anfinsen's (Anfinsen, 1972) theoretical study on predicting the ribo-nuclease folding conformation. However, little progress has been made towards the empirical prediction algorithm for predicting the three-dimensional protein conformations. The three most commonly accepted methods used in the past decade consisted of homology prediction, folding recognition, and ab initio. Our research takes the ab inito approach using computational genetic algorithm (GA) search and optimization method. The GA is based on the concept of natural selection for the "fittest" individuals, or most stable protein conformations. Previously, other groups have explored the similar GA-based method. Unlike nature, each step of the GA has many degrees of freedom that can be altered based on the preferences of the user. The most significant alteration to the genetic algorithm in our study was the fitness function and one additional genetic operator, adaptation. Our fitness function is based on the potential energy calculation of an isolated protein. We have successfully developed a modified GA VBA program (Genetic Algorithm-based Protein Structure Search, GAPSS) that minimizes the potential energy of target sequence and generated a corresponding Cartesian coordinates for each atom. By using a three dimensional graphing software, Accelry's, we were able to visualize the predicted conformation of a pentapeptide, enkaphlin, and compared to natural conformation. However, there were still discrepancies between the predicted and theoretical conformations, which suggests a more refined fitness function and perhaps a survival function should be applied.

Table of Contents

Introduction

Despite the effort to optimize the GA-based search and optimization method, little progress has been made toward the ultimate goal. According to Yang and Liu (Yang & Liu, 2006), there are two major problems associated with the prediction of three-dimensional protein structures from the amino acid sequence. The primary problem is that there is yet to be an efficient strategy for discriminating the native fold conformations from all the other misfolded or unstable immediate conformations. By using the ab intio approach, the generated structures are being discriminated through the comparison of the free energies. A protein has numerous possible intermediate conformations that could exist by slightly varying the positions of the backbone and side chains. The second problem associated with protein folding prediction is the time factor. A protein folds in a matter of milliseconds, meaning that there is no possible way a protein has the time to apply the trial and error method of finding the correct fold conformation with the lowest amount of energy. Therefore, a protein has been proved to have a folding pathway that allows the protein to reach its designated state in the short amount of time provided. The protein folding pathway is also known as the energy landscape taking the shape of a funnel. In summary, the problem with protein folding prediction is that there is yet to be a time efficient method that effectively scans an extremely high dimensional space to find the native conformation.

Protein Background

Amino acids are the fundamental building blocks of the protein. Every single amino acid in protein polymer has the same basic structure. The amino acid structure contains a centralized carbon that is bonded to the following elements: a carboxyl group, an amino group, and a side group usually referred to as the R-group (Residual). The R-group is used as an important identification tool for the amino acids.

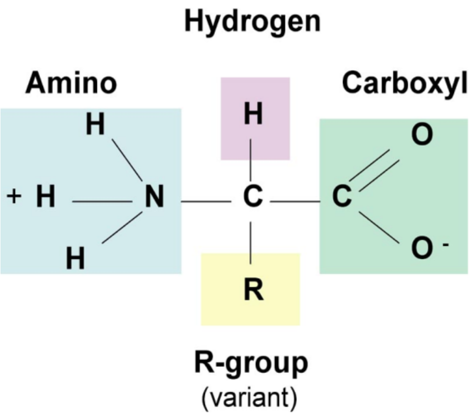


Figure 1: Amino acid with the residue

A peptide bond is formed between two amino acids through a dehydration reaction. Multiple covalently bonded amino acids form a polypeptide. The formation of a protein is divided into four primary stages beginning with the primary structure. The primary structure of a protein is the unfolded sequence of amino acids. The primary structure can also be viewed as the blue print to the formation of a more convoluted 3-dimensional protein conformation.

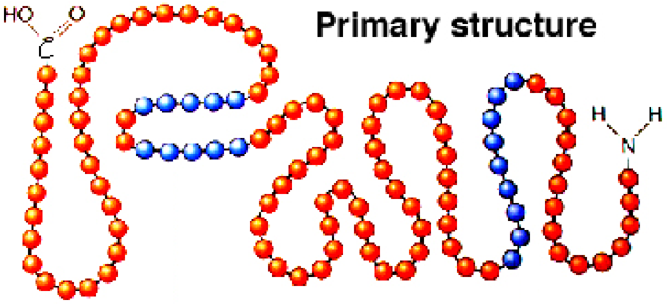


Figure 2: Each bead is representative of an amino acid

Following the primary structure is the formation of the secondary structure. The secondary structure forms by following the blue print instructions that were made during the primary stage.

Alpha helices, beta sheets, loops, and turns are all formed during this stage due to the interactive energies such as electrostatic, ionic, non-bonded, and hydrogen-bonds. All of these elements are used in sequence to form the 3-D protein structure, and are important to the structure of the protein because they help to stabilize the final protein structure. The secondary structure also initializes the beginning of the three dimensional formation through the development of the elements mentioned above. A typical β-sheet and a right-handed α-helix, a more common conformation for α-helix is shown in figure 3.

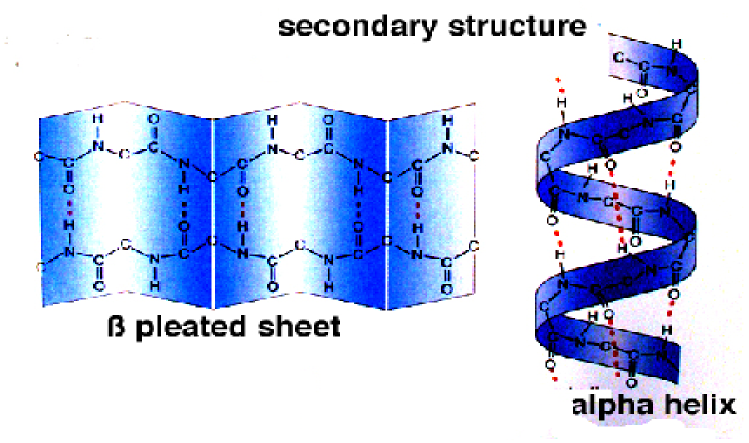


Figure 3: The secondary structure formation of Beta sheets and alpha helices

Alpha-helices form during the secondary structure of the protein because the amino acid acids in the sequence are forming hydrogen bonds with each other trying to stabilize the structure. Alpha helices may either resemble a spring or a right-handed coil that twists clockwise. Another possible secondary structure is the beta-sheet. The beta sheets are formed by the hydrogen bonding of two strings of amino acids in a parallel or anti-parallel direction. The stability of β-sheets depends intensively on the hydrophobicity, steric hindrance, partial charges of the nearby protein segment.

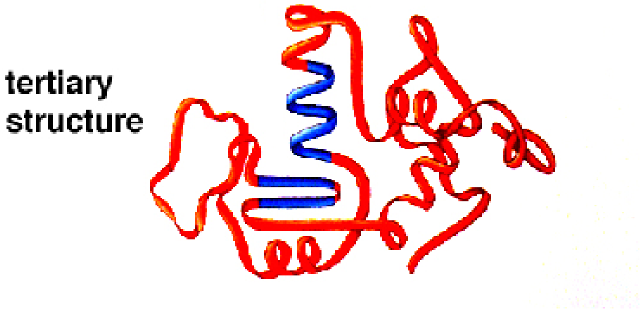


Figure 4: The tertiary structure formation with the combination of secondary structures

The third level of protein folding is the tertiary. The tertiary structure gives dimensional particular 3-D characteristics of the protein. This stage of protein folding is assumed to be the most thermodynamically stable conformation, which suggests that the conformation has the least Gibbs free energy. A portion of the commonly encountered proteins may actually carry on to what is known as the quaternary structure. The quaternary structure is the bonding of several tertiary structures to form a globular unit. The different tertiary groups are labeled as sub units that form the quaternary structure.

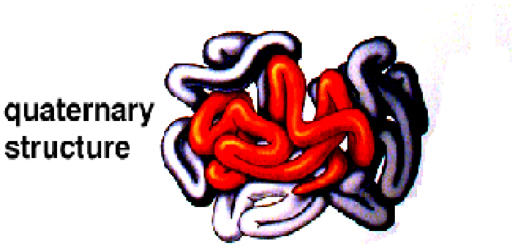


Figure 5: The quaternary structure as a combination of subunits forming a globular unit

Importance of Protein Folding

Proteins are vital for the existence of life. Proteins help to make up the formation of bones, muscles, hair, skin, and blood vessels. They are important for the immune system as well because they are able to recognize foreign invaders and send appropriate signals to stop the spreading of infection or disease. Other evidence relating to the importance of proteins is the product of a protein misfold. If the native conformation of the protein is mutated or folded incorrectly, the result can have a devastating impact on the organism. For example, the dreaded Alzheimer's disease is responsible for memory loss in the elderly generation. Alzheimer's disease is a result of a protein misfold of an unclear origin. This brings about the discussion to the importance of protein folding prediction. The reason there is yet to be a perfected solution to the protein folding problem is because of all the possible conformations the protein could form. Predicting the structure is not as easy as determining the minimum energy conformation because that brings about problems on its own. Since a protein can shift between several intermediate structures to its native conformation in as fast as a millionth of a second, so it is impossible to determine the folding path through observation. Instead other methods have been developed in order to determine the final structure of protein folding, such as ab initio.

Protein Folding Methods

The most contemporary protein folding methods can be categorized into three primary groups: 1) homology method, 2) folding recognition, and 3) ab initio. The first and most well-established method is homology method. Beginning with the discussion of the homology method of protein folding, homology folding uses a comparative modeling strategy. The protein to be examined is compared through the evolutionary progression of the related protein.

The second method is known as the folding recognition method. Folding recognition method, like the homology method, uses a comparative strategy as well. The protein to be evaluated is compared to a data bank of known proteins, and the sample protein is threaded through the already known protein to determine the structure of the sample protein. For this reason, this method is also known as threading.

Ab initio is the third folding method that is used. Ab initio uses a computer simulation to determine the final structure of the protein. The thermodynamic properties of the protein are evaluated to determine the final structure.

Genetic Algorithm

The genetic algorithm (GA) approach to solving the protein folding prediction problem has received a great deal of attention due to the promising results that have been published in the past decade. The genetic algorithm approach is based on the Darwinian theory of evolution, also known as the ‘survival of the fittest’ concept. The GA is compared to the process of mitosis beginning with two parent chromosomes. Each chromosome contains information to the chemical make-up of the parent generation. During the reproduction, the information encoded on the chromosomes may follow one of the following operators so that the diversity is maximized during the process: crossover, mutation, or, selection. The operators occur naturally in biology because they help to create diversity among the offspring. In the realm of protein folding, it is necessary that the best qualities of the parent proteins are passed to the offspring. The main goal of the genetic algorithm is to keep the diversity among the conformation while improving the overall fitness (Yang & Liu, 2006).

Problems with Genetic Algorithm

The genetic algorithm-based protein folding prediction method has yet to be perfected. Several problems with this approach are the difficulty in encoding the information, determining the fitness of the individual and insufficient reproduction process. Proteins have been coded with a binary code (a system of 1's and 0's) and even by using the alphabets. Another problem with the genetic algorithm is that it obtains a set of lower free energy states rather than a single value. The reason the genetic algorithm is forced to give a set of possible values is because a single value will force the energy force field to prematurely converge trapping all the conformations. The third and most important problem, which will be discussed in more detail later in the report, is that there is no perfect fitness function. The fitness function is used to judge the "fitness" of a protein, or in other words evaluate the total energy of the protein (Pedersen & Moult, Genetic algorithms for protein structure prediction, 1996).

Genetic Algorithm Conventions

The genetic algorithm starts with an initial population of random protein conformations. The population numbers vary depending on the preference of the researcher. The initial population generation methods also vary from researcher to researcher. After the initial population has been determined, an objective function is applied to the initial population. The objective function is subjective to the user, but usually it uses a potential energy function that calculates the energy and the fitness is determined as a result of the objective function. Next, the reproduction process takes place with the possible occurrence of one or all three operators: selection, cross over, or mutation. Operators are rules that modify individuals and the population to include diversity to the process. The user has the option of altering the properties of the operators. After the operators are implemented a new generation is produced. The fitness of the new generation is evaluated, and based on the criteria of the program the process could go back to the objective function evaluation for another process. The user also has the option of including a maximum amount of iterations to be used. The GA stops on the occurrence of or two occasions one is that there is a solution or the GA has proved the impossibility of the reproduction.

Fitness Function Comparison

Each step of the Genetic Algorithm has the possibility of being altered slightly depending on the user of the program. Four different approaches will be discussed and evaluated. All four GA's follow the same convention, but all are altered slightly through the initial population, the objective function, fitness criteria, and operators.

The first method that will be discussed is that of Agostini and Morosetti (Agostini & Morosetti, 2003) based off their paper written in 2003. The purpose of their paper is not to perfect the genetic algorithm, but to effectively weight empirical potentials in fitness function. Agostini and Morosetti start with an initial population size of 150 to 250 proteins weighting the random choice of phi and psi angles for each residue using results based on the Brookhaven Protein Data Bank. Agonstini and Morosetti also include a sharing function in their genetic algorithm. The sharing function is included as a way to keep the diversity as high as possible among the different conformations. The sharing function is defined as the distance between two structures. The distance is found by subtracting the fraction of common residue conformations from the number one. When the equation is equal to 0, this means that the two conformations being compared are identical, and if the equation is equal to 1 the two conformations are completely different. The fitness function (discussed below) is divided by the sharing function.

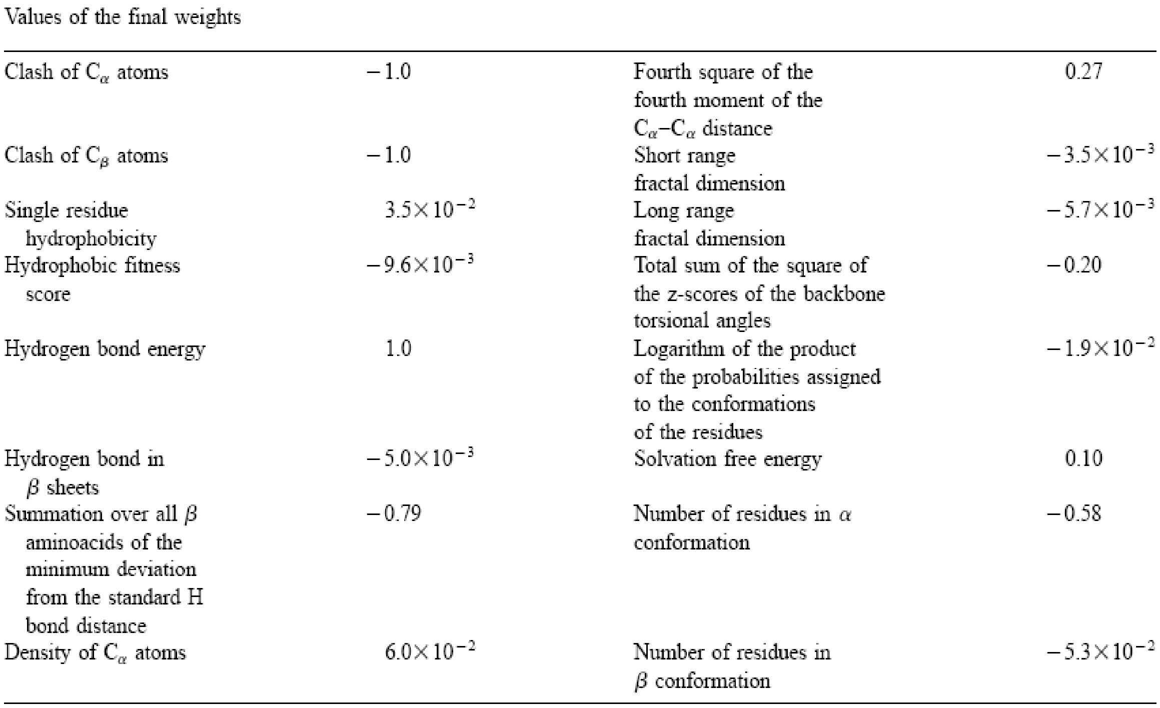
Next, crossing over and mutator operators are used to create a new generation. Within the new generation a method called 'elitist generation replacement' is used. This means that they rank the parent individuals and the offspring's based on their fitness level. Only the conformations with the highest fitness level are used for the new generation. Another strategy used for the selection of the fittest is the injection of new structures. The injection of new structures involves a new value known as multiplicity. The multiplicity value is found by taking the mean value of the sharing values divided by the population size. The user picks a multiplicity value to be the standard (i.e. .50), and if the value of the structures are under the standard a new structure is injected in its place. This method was included as a way of keeping diversity among the conformations. The evolution process sometimes loses diversity, and as a generalization if diversity is lost then the efficiency of the genetic algorithm decreases.

Agostini and Morosetti's fitness function is essentially their chosen objective function (potential energy function) with each individual term multiplied by a coefficient. The coefficient is a constant that is represented as the weight the term has towards the final conformation. The equation shown below is what they have used for the fitness function and for a description of the terms used see Table 1 below.

$$\begin{aligned}\mathrm{fitness} = \{ & a_1 * \mathrm{clash}\mathrm{C}\alpha + a_2 * \mathrm{clash}\mathrm{C}\beta \\ & + a_3 * \mathrm{hydrophobicitySDH} \\ & + a_4 * \mathrm{hydrophobicity}HF + a_5 * \mathrm{total}HB \\ & + a_6 * \beta HB \ \ + a_7 * \beta HB\mathrm{distance} \\ & + a_8 * \mathrm{density}C\alpha + a_9 * \mathrm{fourthmoment}C\alpha \\ & + a_{10} * sfd + a_{11} * lfd + a_{12} * \mathrm{zscore} \\ & + a_{13} * \log \pi \mathrm{probabilities} \\ & + a_{14} * \mathrm{solvationenergy} + a_{15} * \mathrm{number}\alpha \\ & + a_{16} * \mathrm{number}\beta\end{aligned}$$

Equation 1

The results Agostini and Morosetti produced were quite impressive. The table shown below are the results they received. The ultimate goal to test the weights determined for each potential energy term was to reproduce a structure with either a better fitness value or an equal fitness value to an experimental structure. They succeeded in producing both 1000 to 10,000 iterations.



Values of the final weights

Clash of C_α atoms	-1.0	Fourth square of the fourth moment of the C_α–C_α distance	0.27
Clash of C_β atoms	-1.0	Short range fractal dimension	-3.5×10^{-3}
Single residue hydrophobicity	3.5×10^{-2}	Long range fractal dimension	-5.7×10^{-3}
Hydrophobic fitness score	-9.6×10^{-3}	Total sum of the square of the z-scores of the backbone torsional angles	-0.20
Hydrogen bond energy	1.0	Logarithm of the product of the probabilities assigned to the conformations of the residues	-1.9×10^{-2}
Hydrogen bond in β sheets	-5.0×10^{-3}	Solvation free energy	0.10
Summation over all β aminoacids of the minimum deviation from the standard H bond distance	-0.79	Number of residues in α conformation	-0.58
Density of C_α atoms	6.0×10^{-2}	Number of residues in β conformation	-5.3×10^{-2}

Table 1: Values of the final weights

The next method discussed will be that of Cui, Chen, and Wong. (Cui, Chen, & Wong, 1998) They follow the conventional GA procedure, but they have added a few unique tweaks to the procedure. They encourage taking the route of supersecondary structure prediction. A supersecondary structure is a term that they have given to a secondary structure connected to a second secondary structure by a peptide containing one to five residues. The one to five peptide chain connecting the two secondary structures are thought to play an important role because they are what influence the protein to fold. The conformations of the residues can be classified into five major types. The benefit of predicting the supersecondary structure is that the predicted structures will be used as restraints as a way of limiting the conformation space.

Another tweak they add to the model is that in their potential energy equation they have only included two terms. A hydrophobic interaction and a van der waals interaction term. They have determined that these are the most important terms to include because the hydrophobic interactions are what drive the peptide chain to fold, and the van der waals forces are used to reject the incorrect compact structures during the hydrophobic collapse. E_{HH} is the hydrophobic term and E_{vdw} is the van der waals term.

$$E_{Total} = E_{HH} + E_{VDW}$$ *Equation 2*

Their GA procedure start with an initial population size of 500. Although, it should be noted that their initial population was hardly chosen at random. Their procedure is described in detail in their paper. The potential energy of all 500 initial parent individuals was calculated and mapped onto a fitness scale. The fitness function used is shown below:

$$Fitness_{Gl} = 1 + C_{gn} \frac{E_{\max} - E_{gn1}}{E_{\max} - E_{\min}}$$
$$C_{gn} = C_0 + Iner \cdot gn$$ *Equation 3*

$E_{gn,max}$ is the highest individual's potential energy in the *gnth* generation; $E_{gn,min}$ is the lowest individual's potential energy in the *gnth* generation; $E_{gn,I}$ *is* the *ith* individual's potential energy C_0 is a constant that is set to be .02; *incr* is increment of the ratio of fitness of the best individual (with lowest energy) to the worst individual (with highest energy) in each generation.

After each generation, the individual with the lowest fitness had its fitness value set equal to one, the best individual is given the value 1 *C*0 1 *incr* · *gn*. This strategy was implemented as a way of focusing on the ration of the fitness of the best individual and the average fitness. Another added feature to their GA procedure is their crossover operation method. The probability that an individual would be selected was determined by dividing the individual's fitness by the summation of the fitness of all the individuals of the population. Another operation that is included is mutation. There were two mutation operators that could take place. The first operation had the possibility of changing the entire conformation and the second mutation operator focused on a more local search of conformational space.

The next method discussed is proposed by Cox, Morimer-Jones, Taylor, and Jonston (Cox, Mortimer-Jones, Taylor, & Johnston, 2004). This group has included creative genetic operators to help improve the efficiency of the GA model. They begin with an initial population of 200 individuals that is formed by the constructor routine, which generates a number of valid conformations at random. Next, the fitness of each individual is evaluated by using the following equation where E_i is the energy term used (or the potential energy calculated)

$$F_i = -E_i + 0.01 \qquad \textit{Equation 4}$$

The .01 term is added so that even "open" structures with energies equal to zero will have a nonzero fitness; therefore, an equal opportunity to be selected for the crossover operation. Crossover and mutation operators follow the conventional format, but the team added a few new terms to the GA procedure. The first additional operator is called the duplicate predator. This operator works exactly how it sounds. The operator is defined in the simulation as DPL (duplicate predator limit). The DPL signifies the maximum number of times that a given structures is allowed to appear in the population. For example, if the DPL is set equal to zero the operator is essentially turned off meaning that there is no restriction on the number of identical individuals. The purpose of this added feature is to prevent the premature convergence of the population on a non-optimal solution.

The next added operator is labeled 'Brood selection'. This operator is beneficial because instead of creating only two offsprings as a result of crossover it creates a pool of offspring where only the fittest offspring are selected to join the new generation. There are two possible functions of the brood selection that could be used. The first possible implementation is that the only the individuals in the ‘brood’ (or pool) are compared and the best two are passed on to join the new generation. The second implementation involves not only the ‘brood’ population but the parents as well. The whole family is evaluated and the two best (or fittest) members are chosen to move on to the next round.

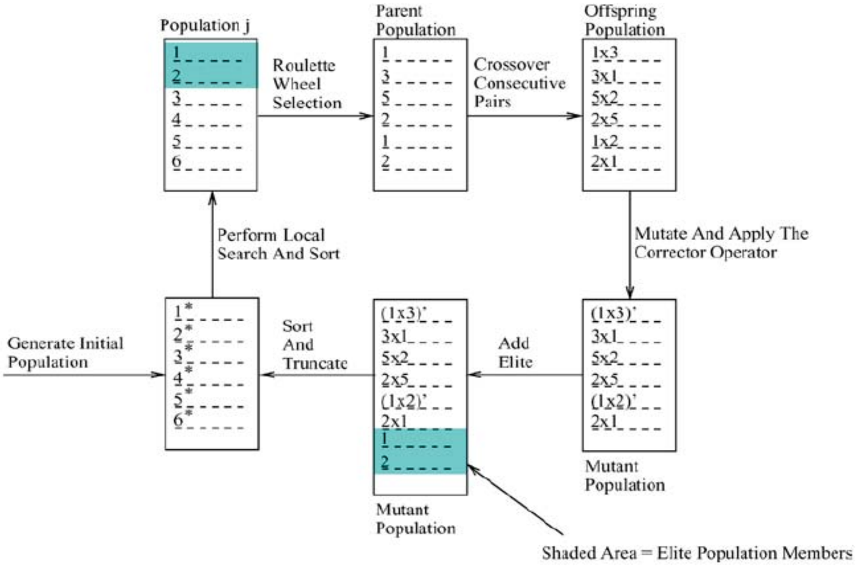


Figure 6: Schematic of GA used by Cox, Mortimer-Jones, Taylor, and Johnston

Methods

Position Calculation

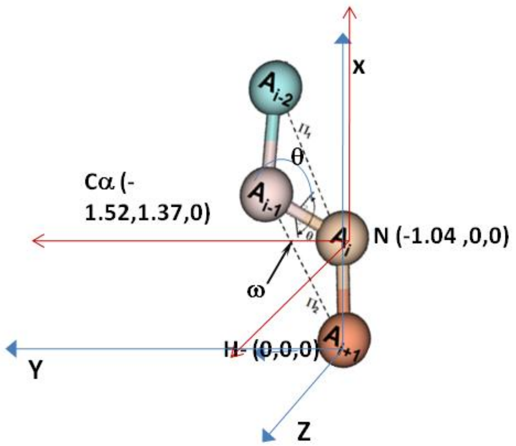


Figure 6: Cartesian coordinate system illustrates the first four atom positions in the polypeptide chain

The Cartesian coordinates were calculated using the dipole moment vector algorithm proposed by Erying in the early 1930's. By neglecting bond stretching and bond angle bending during protein folding, the bond angle and bond length values could be held constant in the calculations. All bond angle and bond length parameters for each atom used in atomic position calculation are proposed and confirmed by Scheraga(Sippl, Nemethy, & Scheraga, 1984). By fixing the origin of the coordinate system on the first hydrogen atom attached to the N-terminus, a series of coordinate system transformations by Euler's angle and distance calculation could be utilized to locate the relative position of every single atom in the target molecule. The transformation and distance calculation matrix shown in equation 1

$$Bn = \begin{bmatrix} -\cos\theta_{ij} & -\sin\theta_{ij} & 0 & -r_{i+1,j}\cos\theta_{ij} \\ \sin\theta_{ij}\cos\omega_{ij} & -\cos\theta_{ij}\cos\omega_{ij} & -\sin\omega_{ij} & r_{i+1,j}\sin\theta_{ij}\cos\omega_{ij} \\ \sin\theta_{ij}\sin\omega_{ij} & -\cos\theta_{ij}\sin\omega_{ij} & \cos\omega_{ij} & r_{i+1,j}\sin\theta_{ij}\sin\omega_{ij} \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \textit{Equation 5}$$

simplifies the computational process of the coordinate calculation. Using the modified position calculation method shown below, the locations of each atom in the target molecules could be determined in angstroms (Å).

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \\ z_{i+`1} \\ 1 \end{bmatrix} = B_1 B_2 B_3 \ldots\ldots \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}$$

Equation 6

Modified Genetic Algorithm

The Genetic Algorithm (GA) is an search and optimization algorithm derived from the basic concept of natural selection process to refine a random search. Due to the user-directed searching algorithm, the GA is a more efficient searching method than the conventional Monte Carlo method. With the assumption of constant bond angles and lengths, the dihedral angles (torsion angles) are the only variables used in the optimization. Each set of torsional angles are treated as a single "chromosome" because each set contains a genetic "blue print" to the folding of a protein. All chromosomes will undergo a genetic operation as a chance to increase the diversity in the population. The first generation, or seeding generation, is randomly generated between -180^{o} and 180^{o} with the exceptions of ring structures (e.g. tyrosine, tryptophan, and proline). The initial average and minimum fitness, or the total potential energy of the molecule at a certain folding state, is obtained from the total energy calculation consisting of electrostatic, van der waal, and hydrogen bond energy terms. The details of energy calculation can be found in sections III – VI. Due to the nature of this program, only mutation and crossover evolutionary operators are applied to the precedent generation. The offspring generation is pooled with the parent generation, and only the top 100 best fit molecules are selected for the subsequential genetic operation. The global energy minimum is obtained when no improvement can be made to the minimum energy level after 20 sequential generations. The complete process flow diagram of the GA operation is shown in ***Figure 7***.

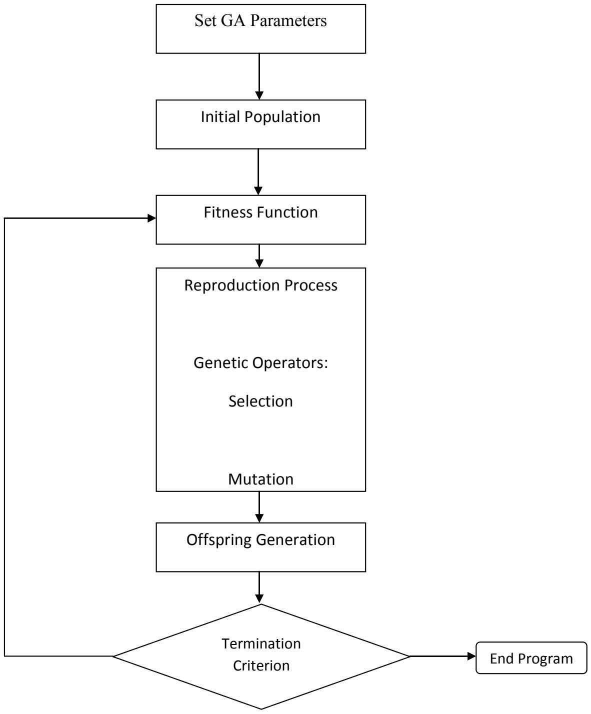


Figure 7: Schematic of Genetic Algorithm

I. Mutation and Adaptation Operator

Mutations are defined in biology as naturally occurring random changes in the chromosome. The mutation operator in the protein folding prediction GA procedure, randomly chooses 10 to 30% of the total populations to mutate. Each mutating chromosome, or set of dihedral angles, randomly adds or subtracts 0^o to 20^o from the current dihedral angle value at four random points within the set of dihedral angles. The modified GA also compares the fitness of the mutated set of dihedral angles to that of the original set; by natural selection, the set with the higher fitness or lower total energy survives, whereas the one with the lower fitness is deleted. A detailed schematic of the mutation process is illustrated in the ***Figure 8***. Adaptation is a unique operator employed only in this modified GA program. Drawing analogy to the adaptation in nature, the adaptation operator in the modified GA utilizes the linear minimization method to further improve the fitness of the overall population in every generation. The linear minimization method uses the gradient calculation to minimize energy based on each dihedral angle within the molecule to optimize locally for a particular set of dihedral angles.

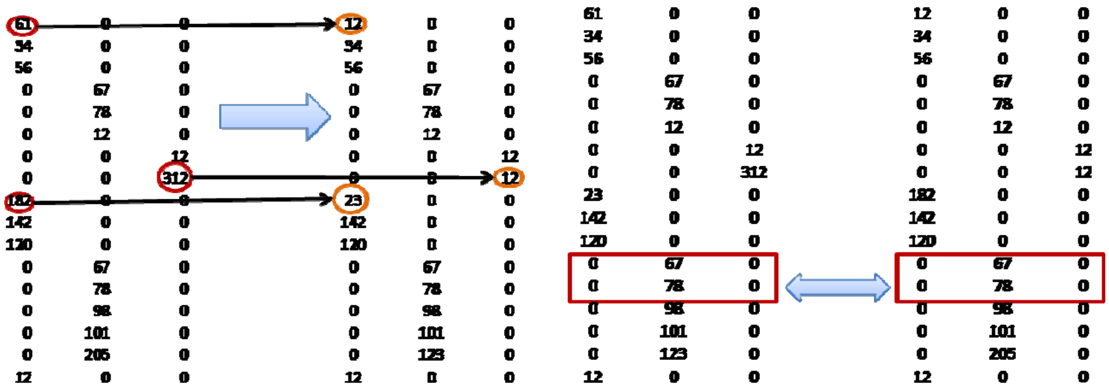


Figure 8: Mutation Operation on a set of dihedral angles (Left); Crossover Operation on the Right

II. Crossover Operator

Crossover also known as the mating process is applied to each generation to improve the overall fitness of the population. The crossover process is illustrated in ***Figure 8***. Crossover allows the offspring to combine the characteristics of both parents. Crossover operation used in this study selects two members of the population at random, and then performs an exchange of dihedral angle. Finally, the parents in the modified GA have only two adjacent “genes” or torsion angles that are exchanged between the parents at two random exchange points. The short sectioned crossovers are optimal in keeping the integrity of the good folding segment for both short and long peptides and in searching for the global minimum energy for both short peptides.

III. Fitness Function

Previous experimental studies and theoretical analysis of the peptide folding prediction have shown that the native state of protein folding, or the functional conformation, has the lowest potential energy. The general form of the total energy consisted of contribution from the Coulombic interactions, Pauli Repulsion and van der Waal Interactions, and finally hydrogen bonding interactions Different forms of intermolecular energy are discussed in more detail in sections IV – VI.

$$E_{Total} = \sum_{i=1}^{n} (coulombic) + \sum_{i=1}^{k} (LennardJones) + \sum_{i=1}^{x} (Hydrogenbonded) \quad \textit{Equation 7}$$

IV. Electrostatic Interaction

All types of electrostatic interactions (1-4, 1-5, and long range) were determined using atom-centered partial charges. These overlap normalized partial charges for each atom of every amino acid residue were chosen to represent a continuous charge distribution in the radial direction. The partial atomic charges were given by Scheraga (Sippl, Nemethy, & Scheraga, 1984), which were determined using the complete neglect of differential overlap/2 (CNDO/2) molecular orbital method. Since the CNDO theory assumes explicit interaction between all valence electrons and zero differential overlap, the slight discrepancies between experimental and theoretical values were reported by several groups. The potential energy involved in electrostatic or Coulombic interactions for any pair of atoms can be determined from

$$U_{Electrostatic}(rij) = \frac{332.0(q_i q_j)}{D(rij)} \quad \textit{Equation 8}$$

Where q_i and q_j are the partial atomic charges of the atom pair in electronic charge units, r_{ij} is the distance between the center of the two atoms in angstroms (Å), and D is the effective dielectric constant. Three hundred and thirty two (332) is the conversion factor used to give $U_{Electrostatic}$ potential in units of kcal/mol. Theoretically, the effective dielectric constant should vary with the polarization of the atoms and short and long range electrostatic interactions; however, due to the complexity of the calculations and the previous experiments, Scheraga has proven that using a value of 2 for all types of electrostatic interactions is a fairly accurate assumption.

V. Nonbonded Interaction

In integrating the nonbonded interaction between 1-4 and higher atom pairs, the Lennard-Jones "6-12" potential function was employed to represent the Pauli repulsion and van der Waals interatomic interaction. All the parameters used in the calculation were obtained from a rigid body crystal method with the assumption of no bond stretching and bond angle bending. The potential energy of any pair of atoms can be calculated from

$$U_{Nonbonded}(rij) = F\,\frac{A_{kl}}{(rij)^{12}} - \frac{C_{kl}}{(rij)^{6}} \qquad \textit{Equation 9}$$

where A_{kl} is the repulsive coefficient initially determined from crystal calculation (then modified for each type of atom, see

$$A^{kl} = \frac{C^{kl}}{2rg^{6}} < rg >^{12}$$

Equation 11, C_{kl} is the attractive coefficient obtained from the Slater-Kirkwood formalism

$$C^{kl} = \frac{3}{2}\left[\frac{eh}{m_e^{1/2}}\right]\frac{a_k a_l}{(a_k / N_k)^{0.5} + (a_l / N_l)^{0.5}}$$

Equation 10, r_{ij} is the distance between the center of two atoms (Å), and F is a penalty factor used to account for the short range nonbonded interactions (1-4 and 1-5 interaction). Since the 1-4 short range nonbonded interactions are highly restricted by torsion space due to the direct bonding through the two center atoms and dominating effect of electrostatic force, a value of 0.5 was used for 1-4 nonbonded interaction, whereas a value of 1 was used for higher interactions.

$$C^{kl} = \frac{3}{2}\left[\frac{eh}{m_e^{1/2}}\right]\frac{a_k a_l}{(a_k / N_k)^{0.5} + (a_l / N_l)^{0.5}} \qquad \textit{Equation 10}$$

where *e* is the electron charge in electronic charge unit, *h* is the planck's constant, m_e is the mass of electron in kg, a_k and a_l are the polarization constants obtained experimentally in 10^{-24} cm, N_k is a constant determined for each atom type.

$$A^{kl} = \frac{C^{kl}}{2rg^6} < rg >^{12} \qquad \textit{Equation 11}$$

where r_g^{kl} are parameters determined using mean-value law from many different crystals at different temperatures.

VI. Hydrogen Bond Interaction

In treating hydrogen-bonded dimers within the peptide structure, a general hydrogen bond potential (GHB) energy "12-10" was used instead of the L-J "6-12". Any pair of hydrogen-bonded dimmers can be calculated using

$$U_{Hydrogenbonned} = \sum_{i=1}^{x}\sum_{j=1}^{y}\frac{A'_{xy}}{(r_{ij})^{12}} - \frac{C'_{xy}}{(r_{ij})^{10}} \qquad \textit{Equation 12}$$

where A'_{xy} is the repulsive coefficient calculated from CNDO/2, and C'_{xy} is the attractive coefficient for a hydrogen-bonded dimmer obtained from similar CNDO/2 empirical calculation, r_{ij} is the bond distance between proton donor (H) and acceptor (X). Both the repulsive and attractive coefficients vary with different types of hydrogen-bonded dimer. Previous study has shown a good fit between the calculated potential and that of value from quantum mechanical (CNDO/2) and experimental dimerization results.

3-D Graphing Software

All 3-D protein structures were graphed using Accelrys DS Visualizer 2.0 from *.xyz Cartesian format. Accelrys DS Visualizers, a licensed freeware, can be obtained from www.accelrys.com with a permit. The XYZ translator, a custom-made VBA program was used to translate the GA output format to the recognizable xyz Cartesian format fit for Accelrys 2.0. All animations and superimposing structures were also created with Accelrys 2.0. Accelrys DS Visualizer 2.0 automatically connected atoms based on the proximity between atoms by recognizing the bond length with +/- 20%.

Variations of the GA

In order to find the most efficient GA, two different options have been made to the previously described GA. The two separate additions were the binary and the secondary structure options. The two implementations were added as an effort to increase efficiency in the least amount of time.

Binary Implementation

The genetic algorithm has the option of taking the binary code approach. The binary essentially uses the original VBA program, but there are a few adjustments made to the code. One difference between the original and the binary code is that the binary code includes a converter program that will convert the torsion angles to binary numbers. A binary number is a numerical notation that is written with a base of two, containing only 0's and 1's. In order to make a binary number that uses the least amount of digits, the torsion angle is divided into three different sections. Using real numbers the torsion angles will range from -180 to 180 with a max of two decimal places. The first digit in the binary code signifies whether the angle is positive or negative. The next nine digits will give the angle number ranging from 0 to 180. Finally, the last seven digits are reserved for the decimal places. Altogether, each torsion angle will be represented by a 17 digit string of 0's and 1's.

Another difference with the binary code is the way the operators are run. The binary code still includes both the mutation and crossover, but they are performed slightly different than the original GA. After converted to real numbers, all the binary digits representing each different torsion angle on one chromosome, are combined into one very large horizontal string of 0's and 1's. The mutation operator works by randomly replacing 0's as 1's and 1's as 0's on the large string of digits. The user is able to specify the percentage of digits to be mutated with the user interface. The second operator that is also in use is the crossover method. There is only one crossover method that is used with the binary code. The way that the crossover method works is by taking a random amount of digits on the chromosome string and exchanging it with another section of digits on another chromosome. After the operators have been run the long string of binary digits is divided back up into separate torsion angles. Finally, another program has been installed to reconvert the binary string back to real numbers. The idea of combing the digits into one large number is that it will help to create more differentiation among the chromosomes, and differentiation helps to achieve the ultimate goal of reaching the native conformation with the lowest energy.

Several problems have been noted with using the binary code. The first problem is that both operators create the possibility of dividing up specific numbers that are required to form rings on the side chains. For example, the amino acid proline has a ring that is a side chain, and when torsion angles are in their binary representation and combined into to the long horizontal string of binary digits the crossover and mutation operators might disrupt the specific angles needed to form the ring; however, this problem was recognized and a solution was implemented. The ring structures of the amino acids are known and are recognized by the program. Once the program recognizes the ring specifications, the program knows to preserve the specific angles. Another possible problem that could occur during either of the operators is that the digits could be arranged so that the numbers are above 180. For example, after the digits are divided back up into separate torsion angles and reconverted back to real numbers, a torsion angle might result in a value of 181 or anything above this number. Clearly, this number is not a valid angle in the program and would cause problems. This problem is solved by having the program recognize the numbers that are greater than 180 and moving the decimal place to the left to correct the potential problem. For instance, a torsion angle might have changed to 500.12, but once the program recognizes that the number is greater than 180 it will move the decimal place on place to the left so that it will become 50.012.

In theory, the binary seems like it has the possibility of creating more differentiation and therefore resulting in conformations closer to the native; however, the main problem with the binary code is that it will require a large amount of time, even more time than the original GA. The reason is that the once five digit maximum torsion angles are being converted into seventeen digit numbers. The overall idea of the project is to find a program that runs efficiently in the least amount of time possible, and the binary code does not fit the requirement.

Secondary Structure Implementation

Because the vast majority of proteins include α-helices and/or β-sheets in their secondary structures, a genetic algorithm tailored to finding these protein structures was created. The goal of this α-helix/β-sheet genetic algorithm was to provide a substantial increase in speed over the conventional genetic algorithm with which the most energetically favorable 3-D protein conformation may be found.

At its core, the α-helix/β-sheet GA works very much like the conventional GA. However, it differs by randomly assigning certain regions of each individual's torsional matrix to become α-helices or β-sheets. This step is done during the random parents generation step, at the very beginning of the GA's run sequence. The secondary structures generated in this way are protected from mutations, though torsional angles outside the secondary structures are unprotected.

Similar to the original GA, crossovers are included as well, but the crossovers are performed slightly different from the conventional GA as well. The control parameters that govern each α-helix or β-sheet within an individual are switched between the two parents of interest. Then, the newly acquired α-helix and β-sheet regions, as governed by a parent's received control parameters, are revised to reflect the structural change. Side chain and branch torsional angles remain unaffected. Regions on each parent that were formerly α-helix or β-sheet regions are revised to be random torsional angles. One of the children is randomly chosen to survive. This results in just one offspring per crossover. The crossover step is then concluded, and the fitness of each parent and child is assessed. The GA then continues as the conventional GA would have proceeded following crossover.

As stated above, the intent of this GA was to expedite the convergence of the conventional GA. By giving the search a “head start” on finding secondary structures, the actual, most energetically favorable structures of polypeptide sequences known to contain α-helices and/or β-sheets can more rapidly be found. However, due to the inherent nature of this GA, it is not compatible with as many varied forms of crossover as is the conventional GA.

Results and discussion

Single AA Structure Conformation

Trial GA simulations were tested on the 13 common amino acids to ensure the accuracy of the parameter entry and functionality of the modified GA VBA program. Each amino acid was generated from a random population of 20 sets of dihedral angles, and 20% of each generation was mutated and adapted to new generation. The GA algorithm terminates at the 10th consecutive generation with any improvement. Each amino acid was analyzed and compared to the natural state with the matched bond length and bond angle. The performance of the GA program was also analyzed based on the total number of atoms optimized per AA and the total number of generation takes to converge. The predicted structures of the 20 common AA’s were listed according to hydrocarbon residues, sulfur-containing residues, acid residues, and base residues.

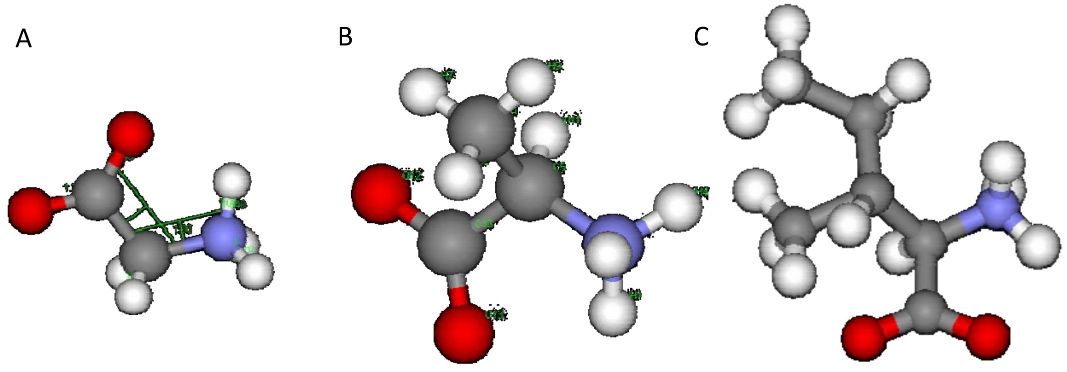


Figure 9: (A) Gly structure; (B) Ala Structure; (C) Ile Structure

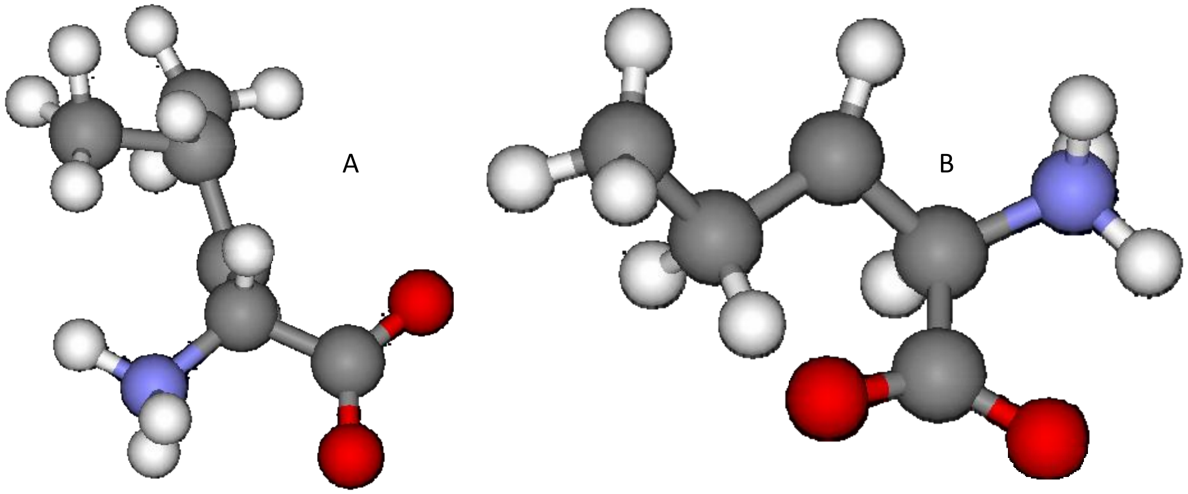


Figure 10: (A) Val Strucure (B) Leu Structure

Amino Acid	No. of Atoms	Total Generation	Minized Energy
Gly	10	7	-2.35412
Ala	13	20	-0.653100124
Ile	22	19	-2.920599024
Leu	15	18	1.178935128
Val	19	27	-4.4653873

Table 2: The performance analysis of hydrocarbon residues

As demonstrated in ***Figure 9*** and ***Figure 10***, the single hydrocarbon residue amino acid matches its natural conformation with all bond angles and bond length matched the input parameter and theoretical calculation. According to ***Table 2***, there is a positive trend between the number of atoms to be predicted and the number of total generations to converge; however, side chains and branches significantly increase the total generations needed to converge. In the case of Val compared to Leu, Val has 3 less atoms than Leu, but Val took 27 generations to converge, which is 8 more generations than that of Leu. Other single amino acids natural conformations are listed below, and all the parameters used in the optimization process matched with the input and theoretical with precision.

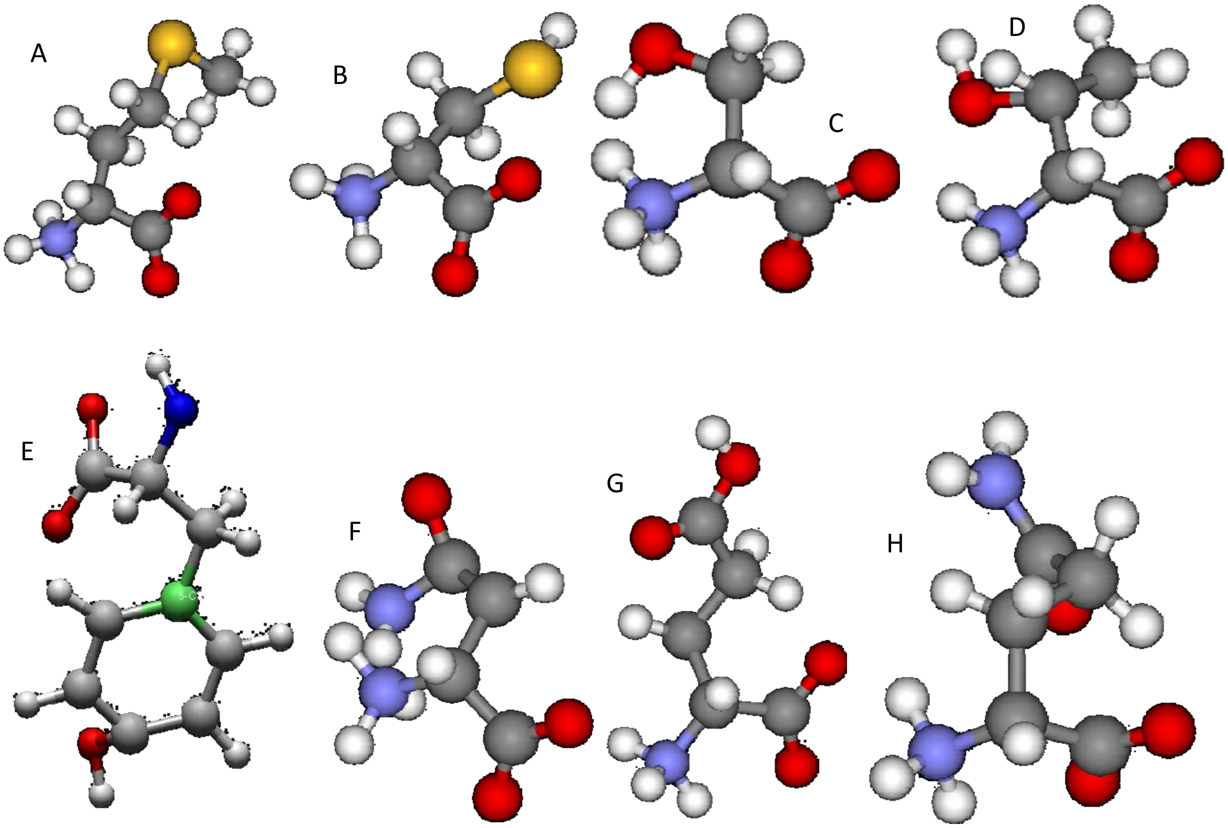


Figure 11: (A) Met; (B) Cys; (C) Ser; (D) Thr; (E) Tyr; (F) Asn; (G) Glu (H) Gln

GA parameter optimization

In order to facilitate the assessment of the GA-based prediction, the experimental fragment of methylated enkephalin (EDA), Protein Data Base ID 1plw, was tested in different conditions to optimize the best set of the conditions to simulate similar length proteins. Minimization process terminates when the potential energy of the Met-EDA can no longer be minimized after 20 consecutive generations. Met-EDA is an optimal protein to assess every aspect of the GA program, since it contains various types of amino acids that examine the program's ability to integrate branches, aromatic rings, and convoluted hydrogen bonded dimmer interactions. The parameters to be optimized include: number of sets of dihedral angles in the initial population, number of consecutive generation need to exit the loop, and percentage of each generation mutated or adapted. The duration time for 2, 5, 20, 50, and 70 sets of the dihedral angles to converge were determined to be 4.3 min, 8.7 min, 25 min, 54.4 min, and 2.1 hour. The time required to compute each set and linear gradient minimization method exponentially increased with the increase of seeding population.

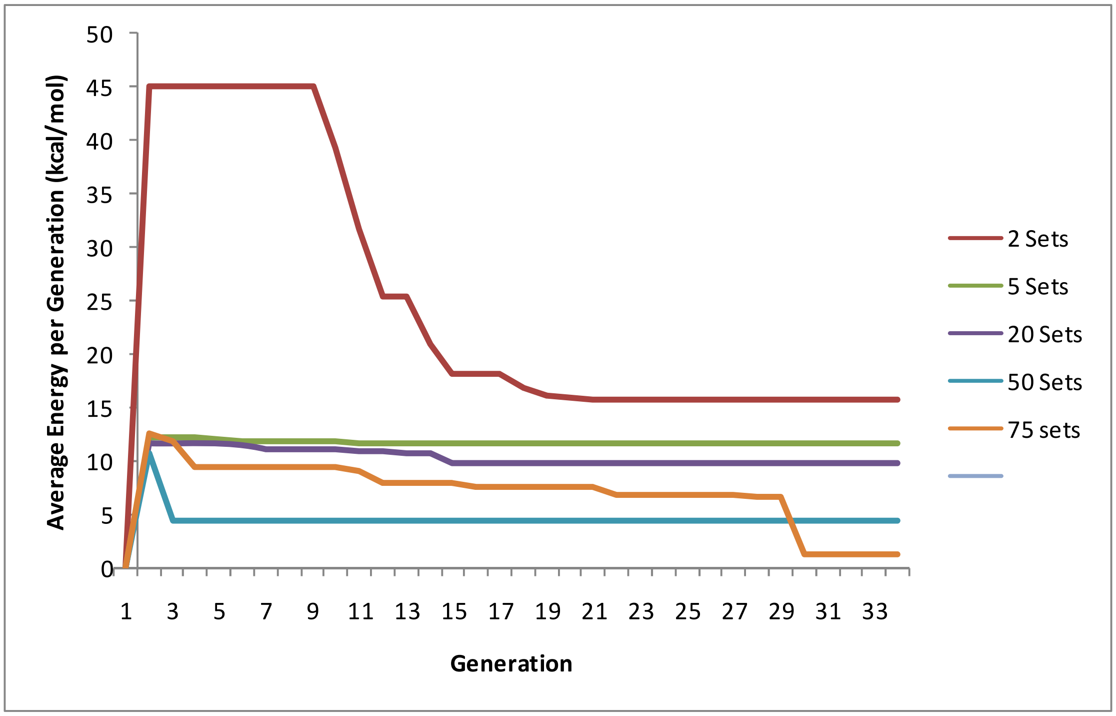


Figure 12: Energy minimization process in each generation

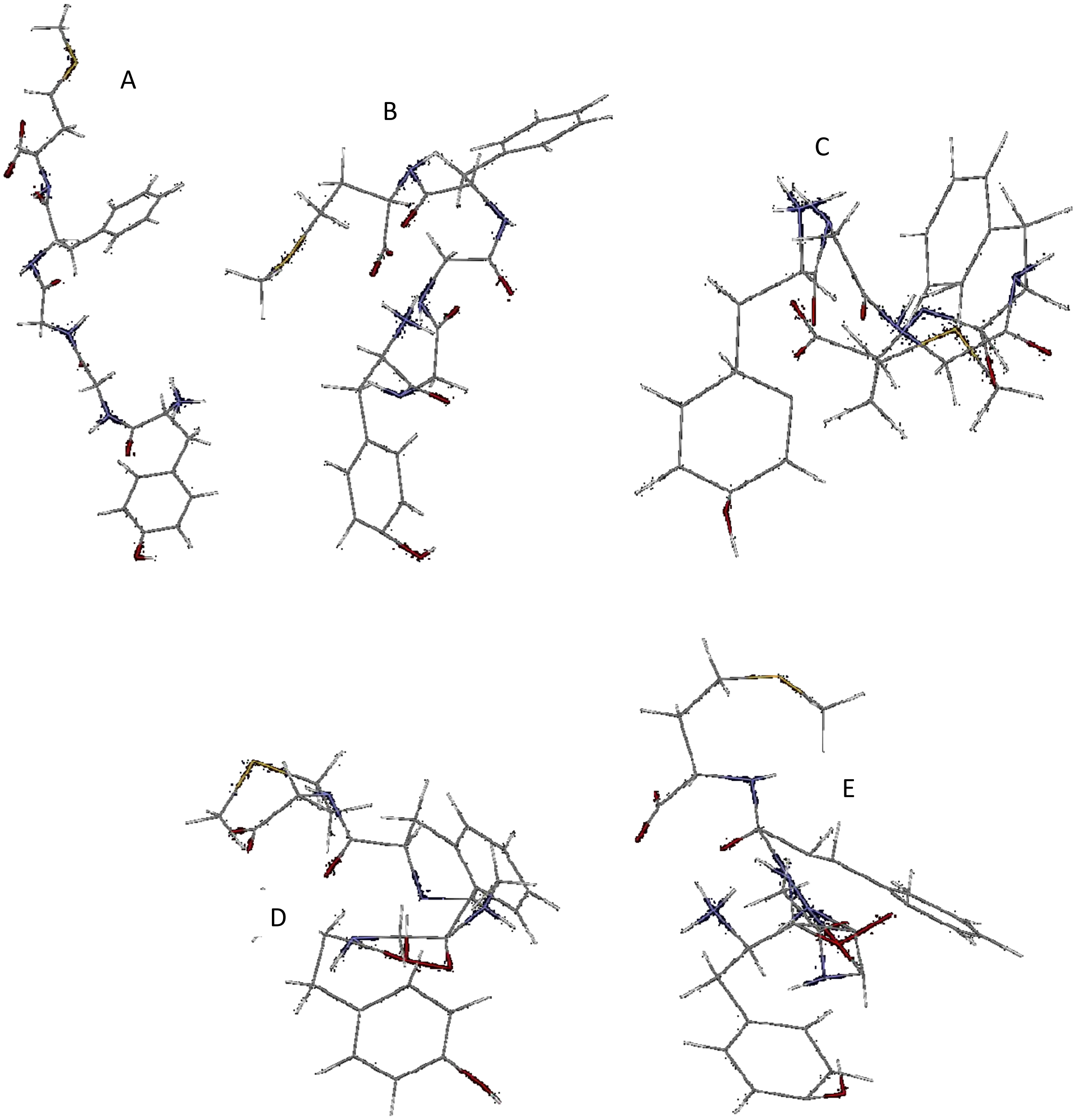


Figure 13: Predicted folding of Met-enkephalin with different initial population: 2 (A), 5 (B), 20 (C), 50 (D), and 75 (E)

By holding the generation limits and percentage of mutation and adaptation in each generation constant, we optimized the initial population for the modified GA. As the seeding population increased, the average energy of initial generation and overall minimum energy decreased dramatically from the initial seeding population of 2 to 5. However, higher the initial seeding, longer it takes for each reproduction process. Therefore, the initial population of 50 was chosen for the sub-sequential optimization steps to obtain the folding conformation, since it performed most efficiently according to *Figure 12*. As shown in Figure 13, with the increasing initial population, the protein takes more and more complex conformations, since there were more chances to mutate and more models to compute with.

After the optimization of initial seeding population of 50, generation limit was tested from range of 5 to 25 with increment of 5. The mutation and adaptation in this optimization step was still 20%. The data was obtained and analyzed. The optimal generation limit was found to be 15.

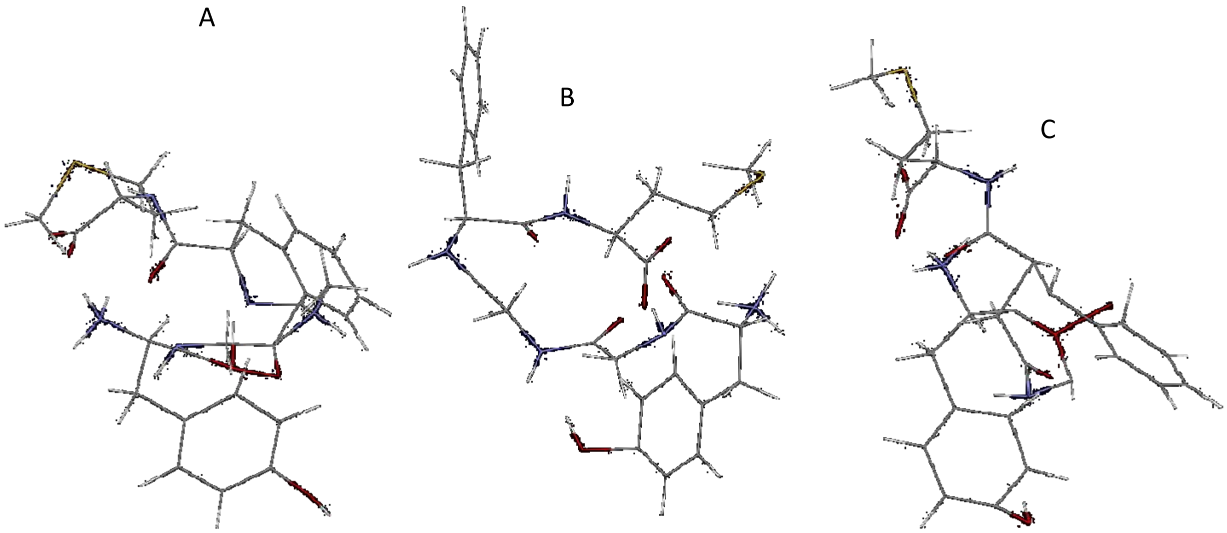


Figure 14: Predicted folding of Met-enkephalin with different seed population: 10 (A), 15 (B), and 20 (C)

Finally with the initial seeding population and generation limit fixed, the mutation and adaptation rates were varied from 10% to 30%. The 30% was slight better than the 20% at minimizing the potential energy of the molecule; however, at high population optimization process 20% was significantly faster than the 30% mutation rate.

Short protein structure prediction

After determined the optimal operating conditions for the modified GA procedure and confirmed the energy and coordinates calculation employed by the modified GA, a series of short peptide chains were ran using the modified GA program and compared with the theoretical folding. The preliminary test was conducted mainly to test the capacity of the GA program, the GA prediction was found to be at very low resolution, since improper torsion angle penalty function was not included and the spatial overlapping can easily occur at short range due to the dominating effect of electrostatic interaction.

Poly-Gly's Chain

A sequence of 5 Glycines (GGGGG) was simulated with the modified GA first to assess the consistency of the GA program. Since Gly is one of the most flexible common amino acids, a long chain of Gly's can take many different conformations. The sequence of 5 Gly's was simulated with an initial seeding population of 50 with 15 generation limit and 30% mutation and adaptation in each generation. The simulation was repeated 3 times and the 3 conformations were obtained from the GA program shown below. The 3 conformations showed certain similarities in folding of the backbone and slight different folding at the C-terminus. The results suggest that random mutation can lead to numerous local minimums, a more strict generation limit should be applied to ensure the consistency of the GA process.

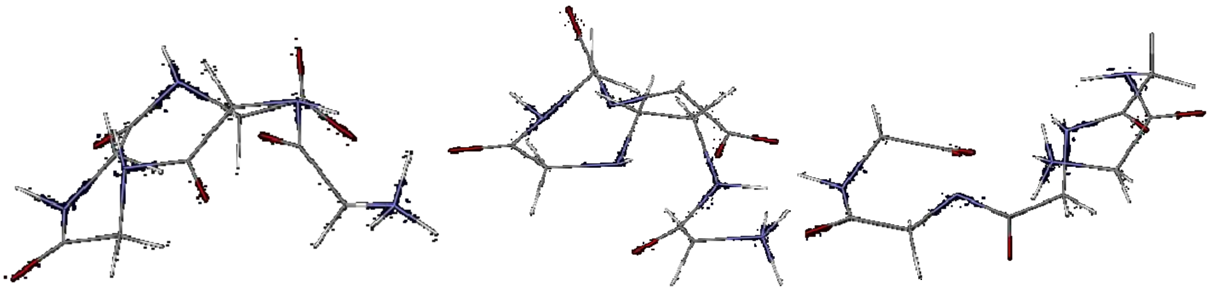


Figure 15: Three conformations used same GA parameters and processing conditions

Enkephalin

Methylated enkephalin-1 is a 5 amino-acid polypeptide known as L-neruopeptide, it is an analog of morphin. Understanding the enkephalin conformations could immensely aid the understanding of human pain perception mechanism. Due to the simplicity of Enkephalin, many literatures have used Met-enkephalin to confirm the optimization methods. The predictions were obtained from 20, 25, 30, 50, and 75 individual set of dihedral angles in the initial population with 15 to 20 generation limits and 10% mutation rate each generation. ___ different structures were obtained from the GA calculation, however, there is no positive match against the theoretical and experimental results. After entering the theoretical dihedral angles into the GA energy fitness function calculation program, the fitness of the theoretical was -11.354 kcal /mol. Compared to the predicted models have the minimum potential energy of -3.1872 kcal/mol, which is 30% lower than the theoretical potential energy. The discrepancy between the theoretical and the predicted could be the reason of the conformational difference of dihedral angles within the residues of Met-enkephalin, a dihedral angle difference at the Phe position.

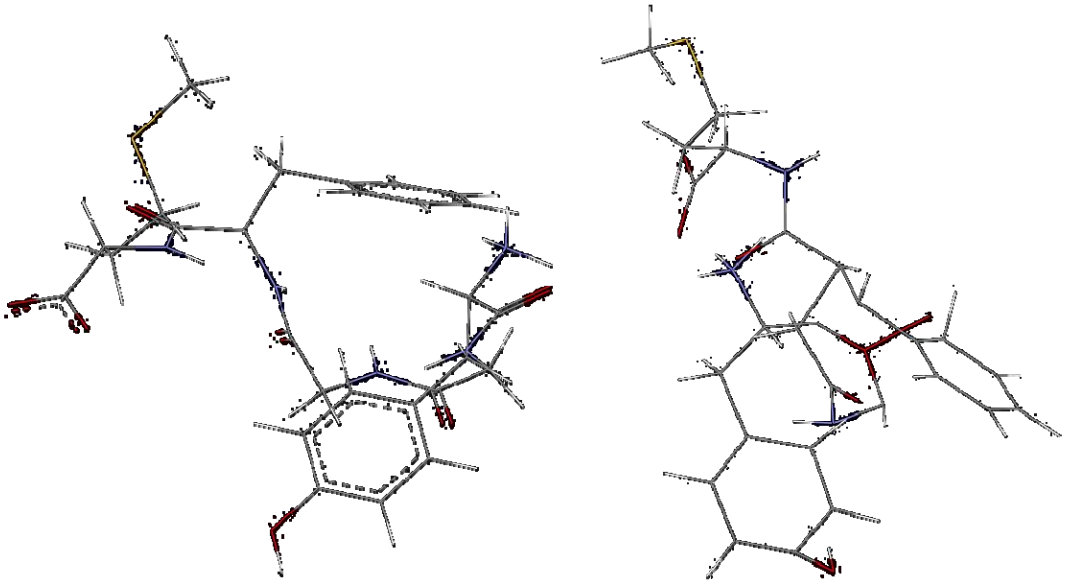


Figure 16: Theoretical conformation of Met-enkephalin on the left and the predicted model on the right

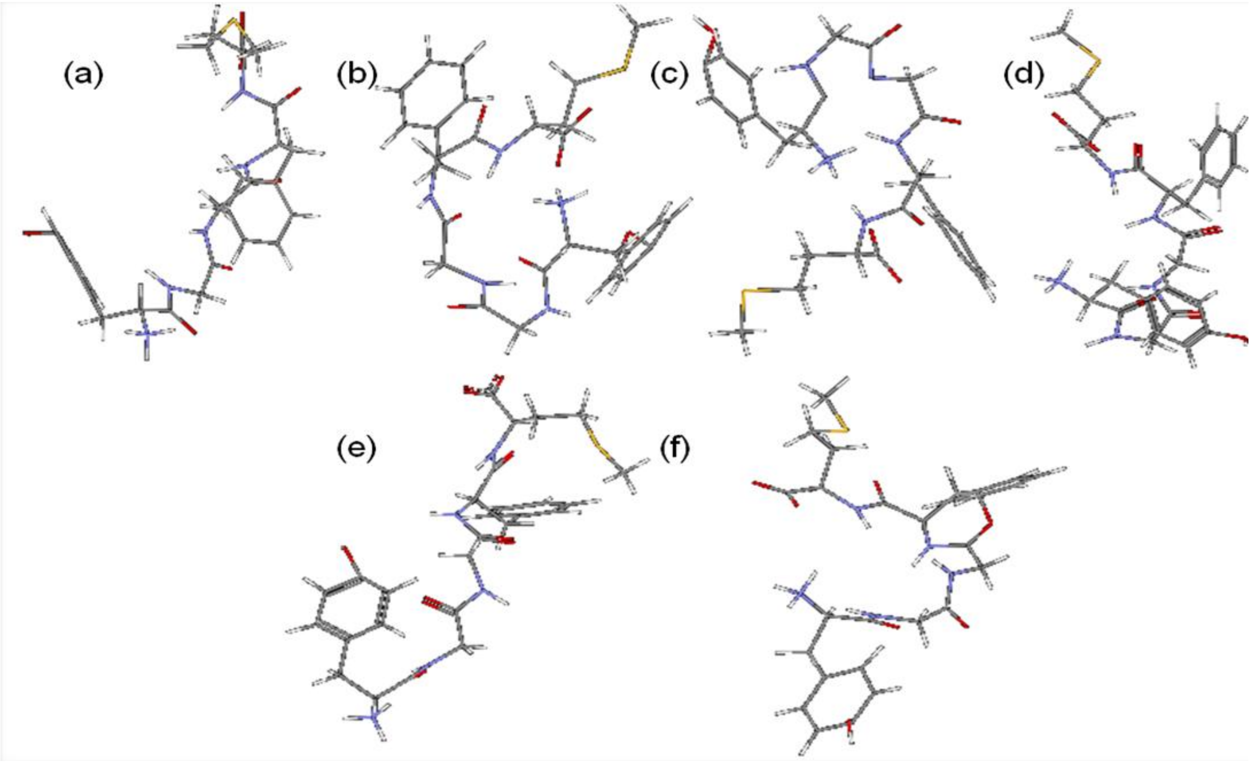


Figure 17: Local minimum conformations predicted by the GAPSS

GAPSS was also able to find numerous structures have linear gradient of zero, which suggests that the GAPSS was able to successfully locate the local minimum potential energy for enkephalin. These predicted conformations could be some stable intermediate folding structures after dissolving in water. However, most of these folding showed one common and puzzling feature: they all display the tendency to force the N-terminus and C-terminus close together. Because the fitness function used to evaluate the potential energy of each individual conformation overlook the contributions from the entropy and salvation energy. The possible formation of hydration shells that hinders the coiling of the protein chain structures were not present in the GAPSS prediction. With the inclusion of the entropic contribution and the salvation energy between the protein and the solvent, we would expect much more zero-gradient conformations that show the various backbone conformations.

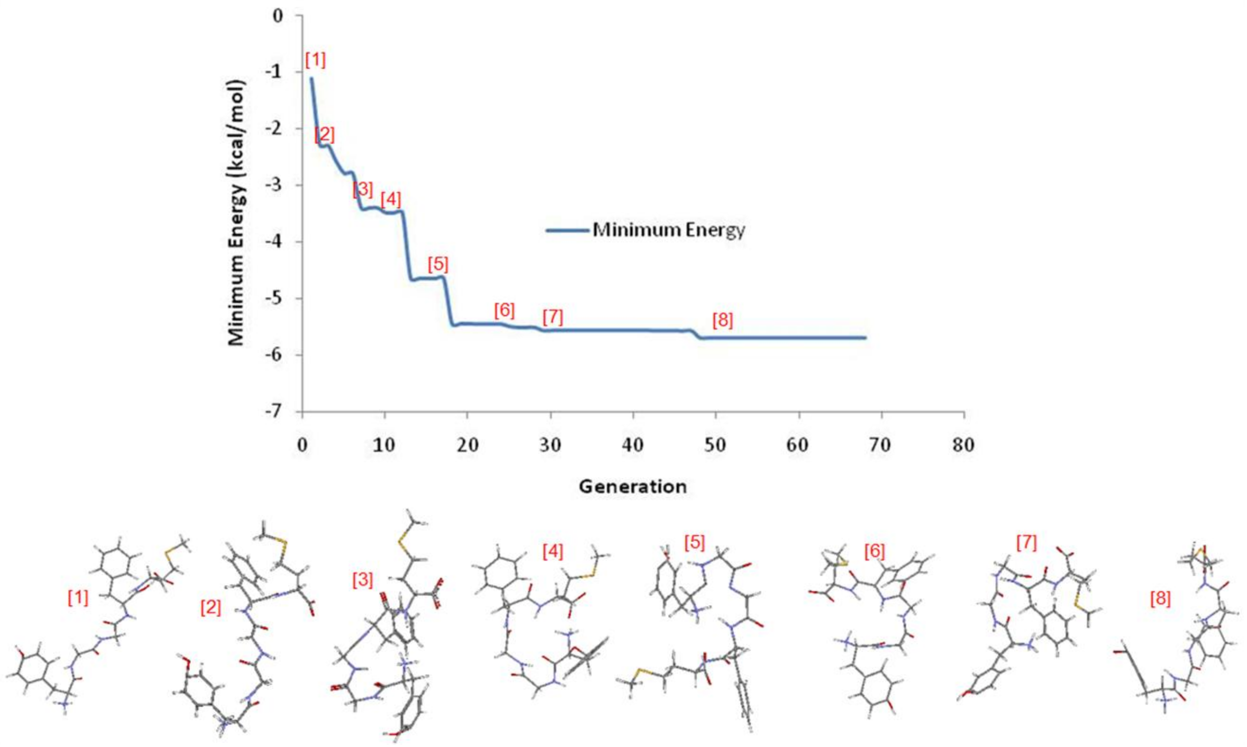


Figure 18: Energy profile and enkephalin conformations

GAPSS with the adaptation operator can easily map out the potential energy profile of the protein folding. The energy profile of enkephalin folding for the best prediction shows an exponential decrease in potential energy as the GAPSS minimizing the potential energy. As shown in figure 18, a clear trend of folding from a rather linear form to the more coiled up conformation. This demonstrates GAPSS's ability to plot out the energy minimization pathway. The results also demonstrated that GAPSS's energy minimization mechanism is working properly. And with the energy decrease as the generation increase, the improvement becomes smaller and smaller. However, due to the less stringent GA parameters set at the beginning of the run. The continuous decrease of potenetial energy suggest that the energy located thus far might be a local minimum and more stringent GA parameters and more intelligent guided search could eventually lead to the global minimum conformation which is the most stable and natural state of the enkephalin conformation.

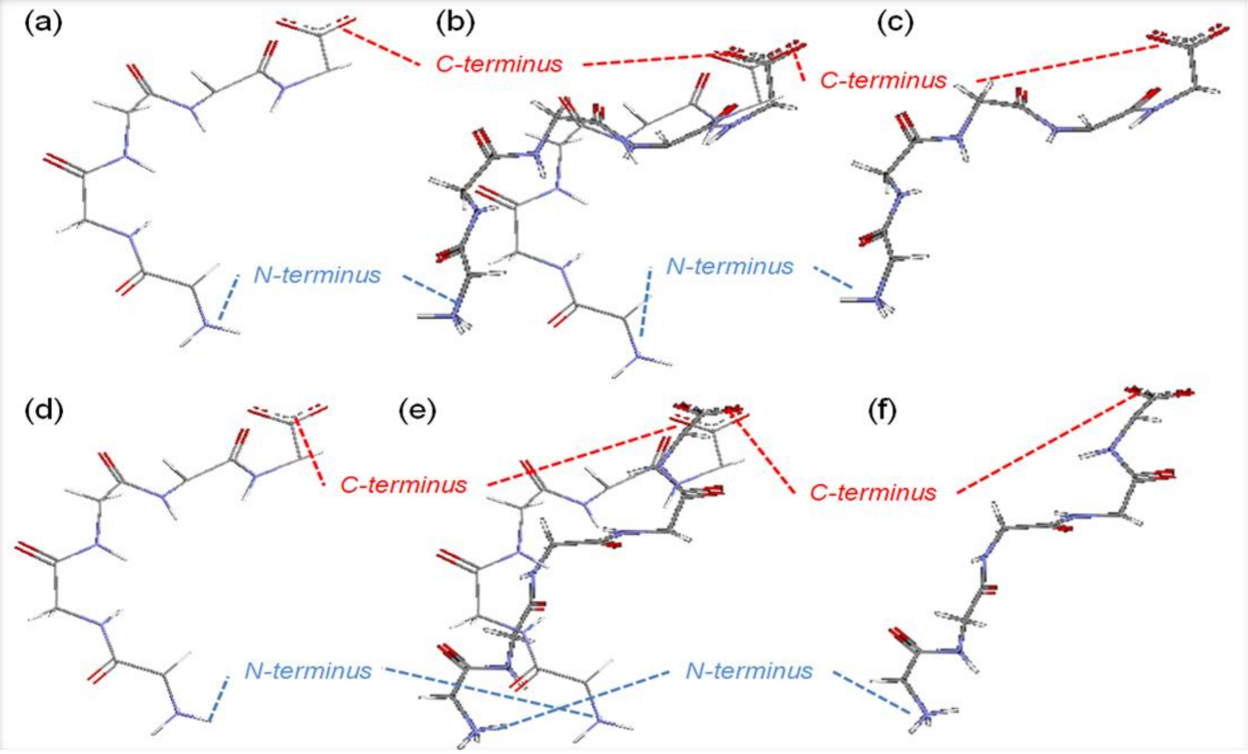


Figure 19: Energy profile and enkephalin conformations

GAPSS was able to predict one conformation that show resemblance of the backbone folding when compared to the theoretical conformation. The RSMD was about 65.43 between backbone dihedral angles with one exceptionally large different occurring on psi-3. However, the side groups conformations are quite different between the theoretical and the predicted.

Interleukin-1 bet A, 2I7B

Interleukin-1 beta A, Protein Data Base ID 2I7B, is a polypeptide known as L-signaling protein. Only a segment of this rather long protein was used in the protein folding simulation. The segment range from 20 to 32, KIEINNKLEF, was predicted in this study. The predictions were obtained from 30 individual sets of dihedral angles in the initial population with 15 generation limits and 10% mutation rate each generation. As shown in ***Figure 20***, the structure predicted is mainly stabilized by the hydrogen bonds between the H-X to O. Another dominating effect shown in the prediction was the electrostatic interaction at short range, which caused the convoluted intertwining of the initial segment of the protein. According to the Protein Data Base (PDB), this segment of Interleukin-1 is a hair pin loop between two beta sheets. The predicted structure displays a loop-like structure, which suggests the modified GA is predicting the basic structure of the sequence.

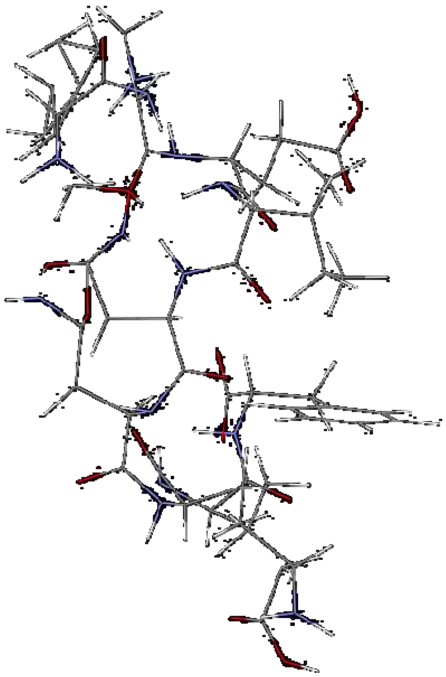


Figure 20: Interleukin-1 beta A, 2I7B , structured predicted using initial population of 75

Performance analysis of GA VBA program on large protein

Myoglobin (PDB ID 1MBC) is a macromolecule with variety of amino acids consisted of 214 atoms. The amino acid sequence of myoglobin is MNKALELFRKDI. This simulation run was mainly conducted to test the consistency and capacity of the GA VBA program for a large protein sequence. The molecule was simulated twice with initial seeding population of 50, generation limit of 15, and mutation and adaptation rate of 30%. Each simulation was run on different computers, the specification of each computer and running time are listed below.

Specification	Computer 1	Computer 2
CPU Speed	2 Ghz	1.3 Ghz
RAM	4 GB	512 MB
Time Taken	4 hr	6 hr

Table 3: Performance analysis of the GA VBA program.

As shown in *Table 3*, the computer 1 with 8 times the Random Access Memory as computer 2 could only out-perform the computer 2 by 2 hr. This suggests the limitations of the processing speed and VBA compiling ability. To further improve the capacity of the modified GA, coding in Visual C++.Net could improve the overall speed of the calculation due to C++'s direct interface with windows. The minimization method used in each reproduction process can also be improved by using the larger steps then refine to the smaller one, instead of taking minute steps in one direction constantly.

The two myoglobin conformations are displayed in *Figure 21*. The two cores are similar with slight dihedral angles changes on the outer core of the protein folding. The mean squared difference (MSD) between two structures was found to be 15.234 Å. The data suggests that the modified GA VBA program is fairly consistent even at calculating large molecules.

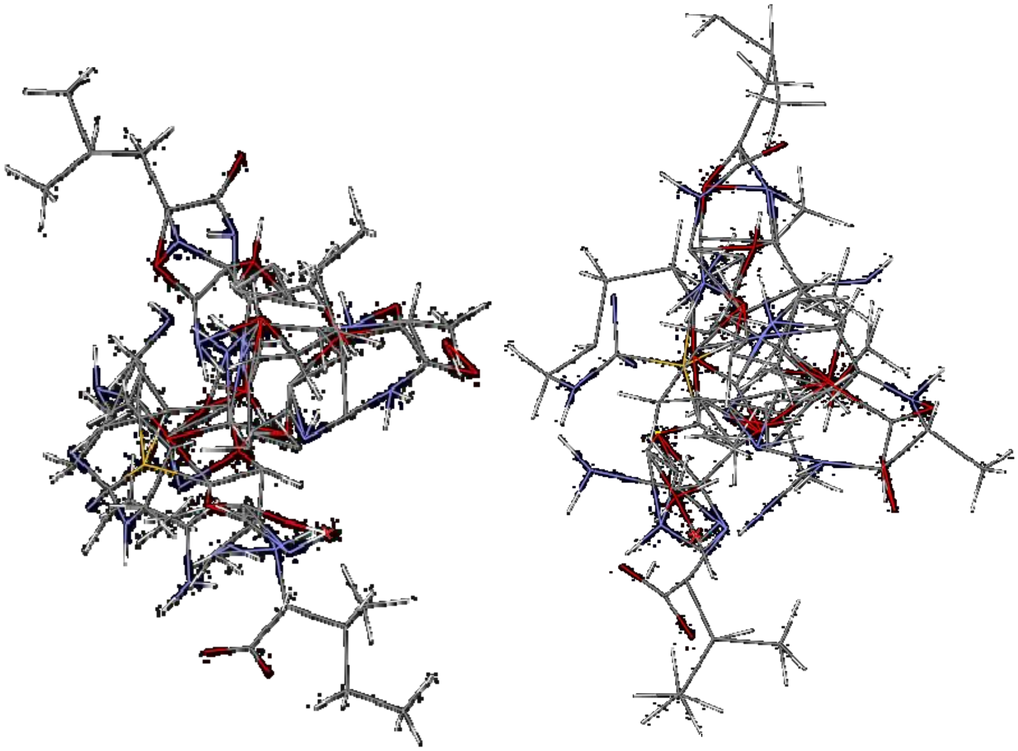


Figure 21: Myoglobin structure predicted with same GA parameter and conditions

Conclusion

In conclusion, the GA algorithm has demonstrated its capability of obtaining the global minimal energy and ability to generate plausible natural conformations for the unknown proteins. However, the resolution and accuracy of the GA depends largely upon the fitness function chosen for the operation and the GA parameters optimization process. To further improve the prediction by the GA, more refined fitness function with improper torsion angle penalty, bond stretching, and bond angle bending should be used. Also, weighting each individual term in the potential energy function brings about an interesting perspective. By determining the contribution of each term in the potential energy function and applying it to the fitness function may help to improve the efficiency of the genetic algorithm. Other possible alteration of the GA with the use of a various combination of operators may also improve the quality of the genetic algorithm.

Bibliography

A. LIWO, P. M., Wawak, R. J., Rackovsky, S., & Scheraga, H. A. (1993). Prediction of protein conformation on the basis of a search for compact structures: Test on avian pancreatic polypeptide. *Protein Science* , 1715-1731.

Agostini, L., & Morosetti, S. (2003). A simple procedure to weight empirical potentials in a fitness function so as to optimize its performance in ab initio protein folding problem. *Biophysical Chemistry* , 105-118.

Cox, G. A., Mortimer-Jones, T. V., Taylor, R. P., & Johnston, R. L. (2004). Development and optimisation of a novel genetic algorithm for studying model protein folding. *Theoretical Chemistry Accounts* , 163-178.

Creighton, T. E. (1988). Disulphide Bonds and Protein Stability. *BioEssays* , 57-63.

Cui, Y., Chen, R. S., & Wong, W. H. (1998). Protein Folding Simulation With Genetic Algorithm and Supersecondary Structure Constraints. *PROTEINS: Structure, Function, and Genetics* , 247-257.

Dandekar, T., & Argos, P. (1994). Folding the Main Chain of Small Proteins with the Genetic Algorithm. *Journal of Molecular Biology* , 841-861.

Dill, K. A. (1990). Dominant Forces in Protein Folding. *Biochemistry* , 7133-7155.

Gibson, K. D., & Scheraga, H. A. (1967). Minimiazation of Polypeptide Energy, I. Preliminary Structures of Bovine Pancreatic Ribonuclease S-Peptide. *Chemistry* , 420-427.

Gordon, M. S. (1969). A Molecular Orbital Study of Internal Rotation. *Journal of the American Chemical Society* , 3122-3130.

Jayaram, B., Bhushan, K., Shenoy, S. R., Narang, P., Bose, S., Agrawal, P., et al. (2006). Bhageerath: an energy based web enabled computer software suite for limiting the search space of tertiary structures of small globular proteins. *Nucleic Acids Research* , 6195-6204.

Klepeis, J. L., & Floudas, C. A. (1999). Free Energy Calculations for Peptides via Deterministic Global Optimization. *The Journal of Chemical Physics* , 7491-7512.

Momany, F. A., Carruthers, L. M., McGuire, R. F., & Scheraga, H. A. (1974). Intermolecular Potentials from Crystal Data. III. Determination of Empirical Potentials and Application ot the Packing Configurations and Lattice Energies in Crystals of Hydrocarbons, Carboxylic Acids, Amines, and Amides. *The Journal of Physical Chemistry* , 1595-1620.

Momany, F. A., McGuire, R. F., Burgess, A. W., & Scheraga, H. A. (1975). Energy Parameters in Polypeptides. VII. Geometric Parameters, Partial Atomic Charges, Nonbonded Interactions, Hydgrogen

Bond Interactions, and Intrinsic Torsional Potentials for the Naturally Occurring Amino Acids. *The Journal of Physical Chemistry* , 2361-2381.

Nemethy, G., Gibson, K. D., Palmer, K. A., Yoon, C. N., Paterllini, G., Zagari, A., et al. (1992). Energy Parameters in Polypeptides. 10. Improved Geometrical Parameters and Nonbonded Interactions for Use in the ECEPP/3 Algorithm, with Application to Proline-Containing Peptides. *The Journal of Physical Chemistry* , 6472-6484.

Pedersen, J. T., & Moult, J. (1996). Genetic algorithms for protein structure prediction. *Current Opinion in Structural Biology* , 227-231.

Pedersen, J. T., & Moult, J. (1997). Protein Folding Simulations with Genetic Algorithms and a Detailed Molecular Description. *Journal of Molecular Biology* , 240-259.

Pitzer, R. A. (1983). The Barrier to Internal Rotation in Ethane. *Accounts of Chemical Research* , 207-210.

Rabow, A. A., & Scheraga, H. A. (1996). Improved Genetic algorithm for the proein folding problem by use of a cartesian combination operator. *Protein Science* , 1800-1815.

Sippl, M. J., Nemethy, G., & Scheraga, H. A. (1984). Intermolecular Potentials from Crystal Data. 6. Determination of Empirical Potentials for O-H...O=C Hydrogen Bonds from Packing Configurations. *The Journal of Physical Chemistry* , 6231-6233.

Standley, D. M., Gunn, J. R., Friesner, R. A., & McDermott, A. E. (1998). Tertiary Structure Prediction of Mixed a/b Proteins. *PROTEINS: Structure, Function, and Genetics* , 240-252.

Unger, R., & Moult, J. (1993). Genetic Algorithms for Protein Folding Simulations. *Journal of Molecular Biology* , 75-81.

Yan, J. F., Momany, F. A., & Scheraga, H. A. (1969). Energy Parameters in Polypeptides. II. Semiempirical Molecular Orbital Calculations for Model Peptides. *The Journal of Physical Chemistry* , 422-433.

Yang, Y., & Liu, H. (2006). Genetic Algorithms for Protein Conformation Sampling and Optimization in a Discrete Backbone Dihedral Angle Space. *Journal of Computational Chemistry* , 1593-1602.