COMPUTATIONAL MODELLING OF PROTEIN FOLDING

M. Lougher, M. Lücken, T Machon, M. Malcomson, A. Marsden.

University Of Warwick

The problem of predicting, from a given amino acid sequence, a protein's function, structure and folding pathway is one of the great challenges of modern theoretical biochemistry. This report reviews the field of *de novo* protein structure prediction.

TABLE OF CONTENTS

1. INTRODUCTION

Proteins are very important molecules, which form the basis of many structures in nature [1]. Proteins are chains of amino acids, which in turn are small molecules, characterized by having an amino and a carboxyl group branched from the same carbon atom. Furthermore, they consist of a carbon-chain backbone and a unique side-chain. Twenty different types of amino acids exist in nature, which bond together by a peptide bond along the backbone to form a protein [2]. The sequences of amino acids required for a given protein are defined in an organism's genetic code [3]. Proteins can range in size from a most basic di-peptide which consists of two amino acids to the proteins such as the titins, having as many as 34,350 amino acids [4].

Proteins perform many essential functions in our and other organisms' cells, and these functions are determined by their structures [1]. This structure is determined by a complex set of interactions between both the amino acids themselves and the environment they are in (usually water molecules and other proteins) in a process known as folding. Protein structure is classified into four levels: primary, secondary, tertiary and quaternary. They refer to different stages of the protein's formation, the primary structure is the sequence of amino acids the protein is made of, secondary structures such as Beta-sheets, Alpha-helices and turns [2], are common to all proteins, and are local structures that saturate hydrogen bond donors and acceptors most efficiently. Tertiary structure refers to how these secondary structures fold giving the shape of the protein while quaternary structure refers to how several chains of amino acids interact with each other. As a protein's function is determined by its structure, a misfolded protein can be severely detrimental to an organism's health. This can be observed on the example of Apolipoprotein E, which has been linked directly to Alzheimer's disease in its misfolded state [5]. As such, the knowledge of both protein structure and the folding mechanism is of great value.

Protein structure can be determined experimentally through the use of X-ray crystallography [6] and the less popular NMR imaging (c.f. Wüthrich's Nobel laureate lecture [7]). Neutron diffraction can also be used, though the conditions required are far more elaborate [18]. The Protein Databank [8] hosts an online database of experimentally determined protein structures and despite there being 59188 entries (as of March 2010) the process remains time consuming [9]. Furthermore, direct observation of a protein's structure does not give information regarding the process of folding itself. And while some experiments (using, for example, NMR spectroscopy [10]) can be used to infer properties about the folding pathway, full knowledge of the folding process remains elusive. As such, computational models of proteins are positioned to give great insight into the physics behind this phenomenon.

Comparative modeling (commonly referred to as template structure prediction or sequence homology) uses the known structure of one protein to predict the structure of another with a similar amino acid sequence. This method relies on the fact that a small change in the amino acid sequence will usually only give rise to a small change in the overall structure [11]. For an introduction to these methods see [12]. Sequence homology is an example of one of these techniques whereby proteins are broken down into fragments, which can be matched in secondary structures by sequence likeness to protein fragments known to form these structures, thereby creating a structural prediction [13]. While these methods can be applied to a large proportion of proteins, little progress has been made recently in the methods used [14].

De novo protein modeling (also known as *ab initio* or free modeling) attempts to find the secondary and tertiary structure of the protein (and its folding pathway) given only the amino acid sequence and is the focus of this report. This approach relies much on the validity of Anfinsen's hypothesis [15] which states that the structure the protein forms in nature (the native structure) is the global minimum of the free energy, and is determined only by the sequence of amino acids. As a consequence of this hypothesis, given an appropriate protein model with a free energy associated with each structure (conformation), global minimization of the free energy will yield the correct native state [14]. However, the free energy, which consists of potential energy and entropy, poses a complex modelling problem, therefore it has become common practice to model only the potential energy surface and successively correct for the entropy term [78].

There are also approaches having elements of comparative modeling and global optimization, called threading [16]. Here, a protein configuration is matched to a library of known structures, to see which it has a highest compatibility for. The amino acid sequence therefore has to be folded to the structure it is matched to and the compatibility of this match has to be evaluated. The compatibility function in this approach is typically an energy function of some sort [17].

As *de novo* protein modelling is the focus of this report, the methods detailed here are often framed in terms of navigation of an energy surface. Proteins have a potential energy associated to each possible conformation [18]. The conformations are varied by changing individual atom positions or, more commonly, by varying certain collective variables (also called combined variables). The potential energy of a protein can then be represented as a function of these collective variables. Plotting this function (or indeed full potential defined by atomic positions) in phase space will then give a potential energy surfaces (hyper-surface), the navigation of which forms a large part of the computational modelling of protein folding. They can be used to perform many calculations in molecular science, such as structure and dynamics of molecules, as shown in [19]. Minimizing potential energy is generally a simple task, which is easily done by gradient methods [20], however potential energy surfaces, are very complicated structures. These already exhibit 10^{11} minima for an oligopeptide (met-enkephalin), consisting of just 5 amino acids [21]. As such the surfaces can be thought of as local minima with transition states that separate them [22]. In protein folding these local minima represent meta-stable states of the protein structure, and the transition states are significant changes in the structure of the protein with changes in the chosen variables. For simple on lattice models the process of finding the global minima has been proven to be an NP-hard problem, meaning there is currently no algorithm that will always find the solution ‘quickly’ [23].

The process of protein folding takes on the order of milliseconds, Levinthal’s paradox [24][25] suggests a contradiction between this folding timescale and the multitude of all possible conformations the system can explore. This leads to the assumption that the native state is determined by kinetic, as well as energy, considerations, such that it is the lowest kinetically accessible minimum, which is not necessarily the global energy minimum by Anfinsen’s hypothesis [15]. A further development of this idea is that the energy surface must take the form of a ‘funnel’ to be biased towards the native state [26].

2. MODELS OF PROTEINS

In order to determine protein structure by energy minimization it is necessary to have an appropriate energy function. This section of the report details the methods most commonly used to achieve this goal.

Ideally, to get an accurate description of the folded state of a protein, you would need to study the interactions between every atom that composes it, along with how these atoms react with other external molecules. This requires a lot of processing power and time, and so models are often used instead. There are several types of models; the most common being toy models, coarse grain models and all atom models. Also, each type of model may be considered as using either an explicit or implicit solvent. Implicit solvation, applies a potential of mean force to the protein, mimicking the average effect of the presence of water molecules [27], which severely reduces the amount of processing power needed, although as with any approximation there is a decrease in the accuracy of results. Explicit solvation means that as well as the protein, thousands of water molecules around it are also modelled, which gives a more accurate representation of how the protein would be in a natural biological environment, but adds a lot more work to the model, and uses a lot of processor power. A comparison between many of the more popular explicit solvent models can be found in [28]. Most models are done by implicit solvation, in which the average effect of the water molecules around the protein is considered, rather than the individual molecules.

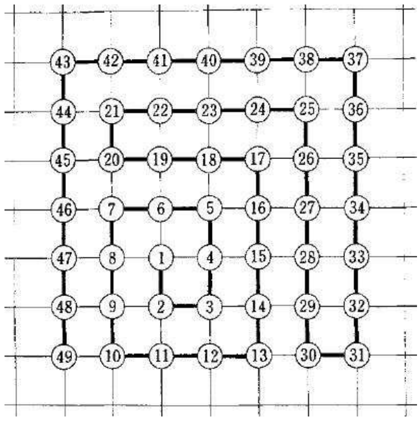


Figure 1: The native conformation of protein SP on a two-dimensional square lattice **[**30**].**

2.1. TOY MODELS

Toy models are models which have factors that limit their association with experimental results, e.g. they are often in two-dimensions, or based on a lattice, or some other property that makes them unrealistic. However, they do represent the interactions accurately, and so many more advanced models are based on these simple models. The most basic coarse grained models are based on Gō-like, or structure based models [29]. These models, named after Nobuhiro Gō who first developed the technique in the early 1980s, use either a two or three dimensional square lattice depending on the simulation being run [30]. Each amino acid is taken to sit on one of the lattice points, fixing the distance between them. The protein is also modelled as self avoiding, so the protein chain never crosses itself. An example of this can be seen in Figure 1, which demonstrates the native configuration of the protein SP on a two-dimensional lattice. In order for the model to accurately describe a protein, it is necessary to take into account both long and short range interactions; long range interactions being those between parts of the chain that are close in space but far apart sequentially (e.g. amino acids 14 and 29 in Figure 1**Figure**), and short range being interactions between parts that are both close in space and sequence (e.g. any consecutive amino acids in Figure 1**Figure**). Long range interactions are accounted for in this model by randomly selecting pairs of amino acids in the structure and seeing if they coincide with any nearest neighbour units, while short range interactions are modelled according to whether the bond angle between the amino acids is the same as in the native state.

A limitation of Gō-like models is that they can only be used to characterise protein landscapes which have negligible frustration [29], where frustration incorporates kinetic effects which may prevent the protein folding into its native state; for example if there are several low energy metastable states which have high potential walls preventing the free energy reaching the actual lowest value in the native state. Hence other coarse grained models have been developed. The following is an overview of several of these models.

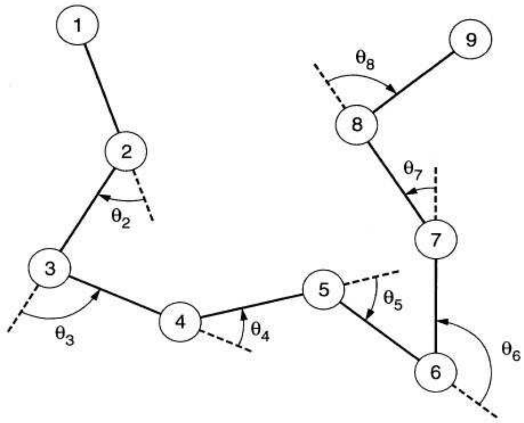


Figure 2: An example of how the AB model can be used to represent a chain made of 9 amino acids [31]

2.1.1 AB MODEL

First published by Stillinger, Head-Gordon and Hirshfeld in 1993 [31], this model simplifies protein folding by using hard spheres to represent only two different “residues” rather than the full range of 20 amino acids. These are denoted A and B and linked together by rigid bonds of uniform length. In this way, there are only three types of interactions to consider between AA, AB and BB pairs, rather all the possible interactions when considering all 20 amino acids, with the fixed bond length also reducing the degrees of freedom of each molecule. In the original two dimensional model, any polypeptide group can be represented as a chain of n residues characterised by the $n-2$ bond angles between them, as seen in Figure 2. Each bond angle has permitted values between $\pm\pi$, with an angle $\theta = 0$ corresponding to the chain being straight. Because of this, this model is referred to an off-lattice model, as opposed to an on-lattice model such as the Gō model.

The interactions in this model were originally modelled as being either due to backbone bend potentials (V_1), which are independent of the sequence of residues, or nonbend interactions (V_2), which varies with sequence and has contributions from each pair of residues not directly attached by a backbone bond. Hence, by defining a term ξ_i to be +1 if the i^{th} residue in the chain is A, or -1 if it is B, the intramolecular potential energy function Φ can be calculated for any protein of length n to be [31]:

$$\Phi = \sum_{i=2}^{n-1} (V_1(\theta_i) + \sum_{j=i+2}^{n} V_2(r_{ij}, \xi_i, \xi_j) \tag{1}$$

And as the backbone bonds between the residues all have unit length, the distances between any two of them, r_{ij}, can be written solely as a function of angle

$$r_{ij} = \sqrt{[\sum_{k=i+1}^{j-1} \cos(\sum_{l=i+1}^{k} \theta_l)]^2 + [\sum_{k=i+1}^{j-1} \sin(\sum_{l=i+1}^{k} \theta_l)]^2} \tag{2}$$

This model uses the above geometry to determine the strength of the potential at each point in the chain. Neglecting intermolecular interactions, the model defines V_1 as a trigonometric potential and V_2 as a species dependant Lennard-Jones potential [31] as follows:

$$V_1(\theta_i) = \frac{1}{4}(1 - \cos(\theta_i) \tag{3}$$

$$V_2\big(r_{ij}, \xi_i, \xi_j\big) = 4(r_{ij}^{-12} - C(\xi_i, \xi_j) r_{ij}^{-6}) \tag{4}$$

where:

$$C\big(\xi_i, \xi_j\big) = \frac{1}{8}(1 + \xi_i + \xi_j + 5\xi_i\xi_j) \tag{5}$$

In this way, the coefficient C has only 3 possible values; +1 for an AA pair, +½ for a BB pair and -½ for an AB pair, such that an AA pair is interpreted as being strongly attracting, a BB pair as weakly interacting and an AB pair as weakly repelling. In fact, results from the initial study done by Stillinger et al [31] imply that this model has the unforeseen property that the residues labelled A and B behave in hydrophobic or hydrophilic manner respectively. This gives an easy method of allocating a label of either A or B to any of the 20 real amino acids for use in this model.

This model can be expanded into three dimensions with little modification save for the actual physical interpretation of the model [32]. Instead of thinking of the protein as a chain of n circles linked by rigid rods of unit length, it is necessary to describe them as a series of n spheres, each of radius ½, and still labelled A and B depending on whether they are hydrophobic or hydrophilic as before. The modelling process then becomes a way of finding the optimum position of these balls in three dimensional Euclidean space such that any two consecutive balls within the chain slice mutually and the energy of the entire system reaches a minimum. The mathematics of this three dimensional model is exactly as described above, but with the position vector of each residue now represented by three components instead of two (i.e. the only change is that all vectors now have a z direction).

The sum of interactions, be they attractive or repulsive from a non-bonded pair or from the backbone bend, creates a vast array of possible ground state energies, each with an associated geometry describing how the protein could fold to reach that energy. This is the property of the model that links it so closely with protein folding, and by running a simulation of each possible folding pathway, a minimum ground state energy can be found, and hence the native structure determined

A further feature of this model is how it can be used to imitate the tendency of proteins to fold into a compact globular form [31]. Consider two parallel strands of residues, each of which is linear (to simplify the maths of this example calculation) with a separation D large compared to the bond length.

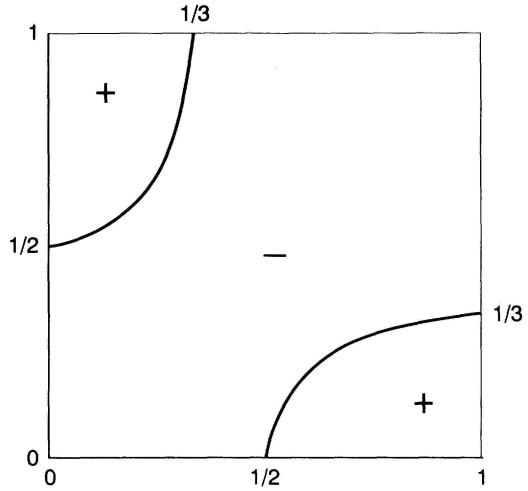
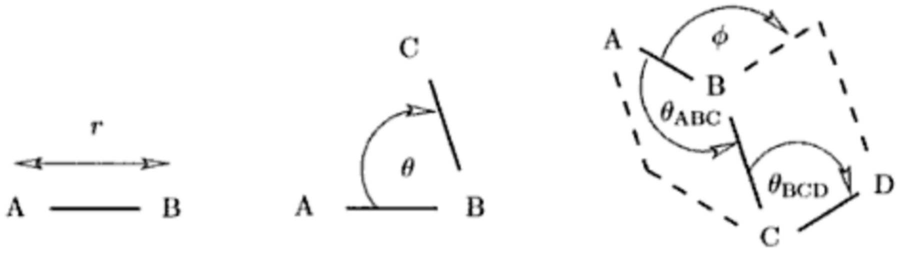


Figure 3: A map of regions of net attraction (-) and net repulsion (+) for two parallel strands, each of which is linear and has composition of residue A of x or y. [31]

Define x and y as the fraction of each strand which is comprised of A residues (there is no need of any further knowledge of the strand composition or sequence than this due to the assumption that D is large). It can be shown by integrating the attraction of an A residue on the x strand with all of those on the y strand and vice versa that the sign of the net interaction between the two strands is bounded by a hyperbolic function [31], as can be seen in Figure 3**Figure** . From this plot, it can be interpreted that if (x, y) lies anywhere in the net attractive region, and the two strands are long enough, then the protein will find it energetically favourable to fold. If, however, (x, y) lies in the net repulsive region, then the likelihood of a fold depends on how long the strands are, as the energy required for a u-turn is fixed, and so the amount of repulsion when weighted with this fixed turning energy must be calculated to conclude whether or not the protein will fold in this instance. However, if the strands are long enough, and the composition of A is taken to be normalised over such a long strand, then the protein will always fold as the line x = y lies within the attractive region [31].

An on lattice variant of the AB model is the HP model. Where the amino acids are described as polar rather than hydrophilic, meaning that the only interaction modelled is the attraction between pairs of hydrophobic residues, with other interactions neglected due to being weak comparatively [33].

For off lattice models in three dimensions, the models are necessarily specified by more than one angle. The following diagrams describe these angles.



$$r = |\mathbf{X}_{\mathrm{A}} - \mathbf{X}_{\mathrm{B}}| = |\mathbf{r}_{\mathrm{AB}}| \qquad \cos\theta = \hat{\mathbf{r}}_{\mathrm{AB}} \cdot \hat{\mathbf{r}}_{\mathrm{CB}} \qquad \cos\phi = \frac{(\hat{\mathbf{r}}_{\mathrm{AB}} \wedge \hat{\mathbf{r}}_{\mathrm{BC}}) \cdot (\hat{\mathbf{r}}_{\mathrm{BC}} \wedge \hat{\mathbf{r}}_{\mathrm{CD}})}{\sin\theta_{\mathrm{ABC}} \sin\theta_{\mathrm{BCD}}}$$

Figure 4: Geometrical definition of internal coordinates with bond length on the left, bond angle in the middle and dihedral angle (torsion) on the right [18].

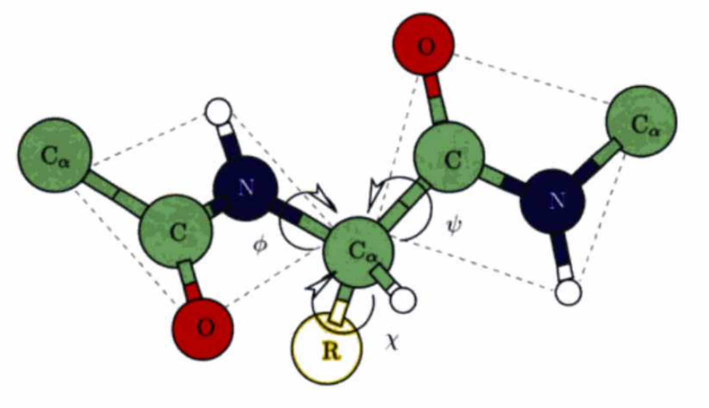


Figure 5: Diagram showing, in three dimensions, the dihedral angles ϕ, ψ and χ [18].

2.1.2 BLN MODEL

An extension from the AB model is the BLN model, first proposed by Brown, Nicolas and Head-Gordon in 2003 [34]. The aim of this model was originally to improve on the resolution of the AB model so that the differences in the folding mechanism between two proteins which end up with the same topology could be studied.

The basis of this model is similar to that of the AB model; however a third type of residue has been added. Residues are either labelled as hydrophobic (B), hydrophilic (L) or neutral (N), hence the name of the model. It adds further characterisation than the AB model did, in as much as some amino acids are very weakly hydrophobic or hydrophilic compared to others, so much so that they are effectively neutral. The addition of this third residue type modifies the maths somewhat from the AB model, with the total potential energy function Φ now written as [34]:

$$\Phi = \sum_{\theta} \frac{1}{2} k_{\theta} (\theta - \theta_0)^2 + \sum_{i} \left[V_1(\varphi_i) \sum_{j \geq i+3} V_2(r_{ij,} \xi_i, \xi_j) \right] \tag{6}$$

It can be seen that this equation is similar in form to that for the AB model, with the addition of a bond angle term representing a still harmonic potential, where k_θ is a force constant, and θ_0 = 105°; the bond angle for which there is no added potential. Also, there is defined a separate dihedral angle φ which is the angle of between the two bonds formed with each residue.

The forms of V_1 and V_2 are also different in this model, and are defined as follows:

$$V_1(\varphi_i) = A(1+\cos\varphi_i) + B(1-\cos\varphi_i) + C(1+\cos 3\varphi_i) + D(1+\cos[\varphi_i + \tfrac{\pi}{4}]) \tag{7}$$

$$V_2(r_{ij},\xi_i,\xi_j) = 4\varepsilon_H S_1(\xi_i,\xi_j)[\left(\frac{\sigma}{r_{ij}}\right)^{12} - S_2(\xi_i,\xi_j)\left(\frac{\sigma}{r_{ij}}\right)^{6}] \tag{8}$$

The additional terms in the backbone bend potential V_1 represent the change in the form of the potential for different structures. This is designed to deal with three different angle types: helical (A = 0, B = C = D = 1.2ε_H), extended (A = 0.9ε_H, B = 0, C = 1.2ε_H, D = 0) and turn (A = B = 0, C = 0.2ε_H, D = 0), as each has a slightly different impact on the total potential. Figure 6 (modified from [35]) shows a visual representation of each of these three types of angle. The term ε_H is included to set the energy scale, and is related to the strength of hydrophobic contact.

The nonbend potential V_2 also differs from the previous case by the inclusion of ε_H and the constant σ, which is the length of the bonds, usually set to 1 for simplicity, and the two coefficients S_1 and S_2. These perform much the same job as the coefficient C in equation (4), with their values being dependent upon the types of residue in each particular bond: for a BB pair $S_1 = S_2 = 1$, for an LL or LB pair $S_1 = ⅓$ and $S_2 = -1$, and for all NL, NB or NN pairs, $S_1 = 1$ and $S_2 = 0$. In this way, the attractive forces in the model are due to interactions between pairs of hydrophobic residues, while interactions between any other pair of residues exhibit a varying degree of repulsion.

As shown by experiment by Brown et al [34], this improved model can distinguish between different folding sequences that result in near identical native states, and so can be used to study misfolding in detail; one discovery being that intermediate metastable states can have a heavy influence on the geometry of the final fold.

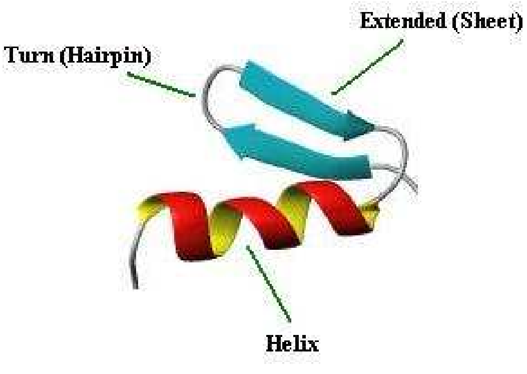


Figure 6: Image showing example of helical, extended and turn secondary structures. After [35]

2.1.3 TUBE MODEL

This model, introduced by Hoang et al [36] in 2004 models a protein as a tube rather than beads attached by rigid bonds. This model only considers interactions that are universal between all proteins [37], such as steric (i.e. space) constraints, hydrogen bonding and hydrophobicity. The model consists of a chain of amino acids, each only represented by their C_α atoms (the carbon atom at the centre of the backbone of the amino acid which links to other groups) which lie along the axis of a self avoiding flexible tube at a set distance of 3.8 Å, which is an accurate approximation for most proteins. Also, chemical constraints mean that the bond angle between any three consecutive C_α atoms along the chain is limited to between 82° and 148° [36].

The tube has a finite thickness of 2.5 Å, and so more accurately describes the packing of the side chains of the amino acid than the ball and rod models. Also, nearby tube segments interact in a way that mimics anisotropic interactions of hydrogen bonds [36]. The tube configuration is determined by ensuring that the radius of a circle drawn through any triplet of C_α atoms is smaller than the tube radius, which at the local level ensures the radius of curvature of the tube cannot be smaller than the thickness, preventing sharp corners, while at the nonlocal level it forbids self interactions. Other constraints on the model include steric considerations, which ensure that no two non-adjacent C_α atoms are allowed to be close than 4 Å and hydrogen bonds being limited to 2 bonds per C_α atom.

This model is useful for studying generic hypothesise of protein formation (e.g. [37]) as it is designed to only use interactions universal to all proteins. It is also useful for studying formation of amyloid fibrils, which are insoluble aggregates of proteins that can form in the body, as these all share common condensation-ordering mechanisms [38]. However, the model does not include biases towards specific configurations, and therefore has the disadvantage that it cannot be used to look at unique or uncommon interactions in specific protein species.

2.1.4 OTHER MODELS

This section briefly details the basics of three other popular models, the CHARMM [40], AMBER [41] and ECEPP [42] models. For a comparison of these three models applied to a simple peptide see [39]. It is instructive to note that ECEPP was the best performing of the three, particularly in simulations without adiabatic relaxation. It should also be noted that both the AMBER and CHARMM models are a part of software packages that include simulation tools.

The CHARMM model (Chemistry at HARvard Macromolecular Mechanics), developed by the chemistry department at Harvard, is an example of an extended-atom model, as instead of modelling entire amino acids as coarse grains, the only reduction from modelling every atom is that hydrogen atoms are joined with whichever heavier element they are bonded to into a single coarse grain. Thus this model takes up more processing power than the previous coarse grain models, but still reduces computation time by not counting hydrogen atoms as separate entities [40]. This is a very early example of a coarse grain model, but has been continually updated since its inception.

The AMBER (Assisted Model Building with Energy Refinement) model has specific coarse grains associated with different common types of molecular groups, e.g. aromatic carbon groups, double bonded carbons, nitryle groups etc [41]. Hence, this model is again not as sophisticated as others in the way it averages interactions, and it has over 30 different types of grain, so uses more processor power than the AB or BLN models, but as it has a finer resolution, so to speak, the results obtained from this model have a greater parallel with reality.

The ECEPP model, which is less like a coarse grain model and more like an all atom model, and has led to many variations since its inception in 1973, which are still being developed today [42]. The complexity of this model renders this report an unsuitable place for its explanation; as such the reader is referred to [43] for further details.

3. Fundamentals of Simulation

Having constructed a satisfactory protein model, the potential energy surface it generates must be explored. Many of the methods detailed in sections 4 onward use, as their base, either molecular dynamics or Metropolis Monte Carlo methods. This section gives a summary of the basic ideas behind these two ubiquitous approaches.

3.1. Molecular Dynamics

Molecular dynamics (MD)[44] consists of computer simulations of molecular and atomistic interactions via the laws of physics and statistical mechanics represented by a force field, i.e. an equation for the forces acting on each part of the molecule, which modifies over time as it changes the configuration of the molecule [45].

In basic molecular dynamics an update equation is created by discretizing and integrating Newton's laws. An example would be:

$$\vec{X}(t+\Delta t) = \vec{X}(t) + \Delta t\,\vec{v}(t) + \Delta t^2 \vec{\nabla} V(\vec{X}) \tag{9}$$

Where $\vec{X}(t)$ is a vector describing the positions of the atoms in the model, $\vec{v}(t)$ the velocities, $V(\vec{X})$ the potential (generated by the model) and Δt the timestep. The timestep must be chosen small enough to ensure the dynamics of the system remain physical [46]. Equation (9) can be used for as long a time as the simulation needs to run. In order to incorporate the effect of temperature (and hence thermal fluctuations) into the system Langevin dynamics [47] are often used, these methods introduce a stochastic variable into the update equation, so introducing an element of randomness into the motion. MD was first applied to a protein (bovine pancreatic trypsin inhibitor) in [48]. Because MD simulations are in effect solving the equations of physics, their results give large amounts of information, including the folding pathway [47].

3.1.1 FOLDING@HOME

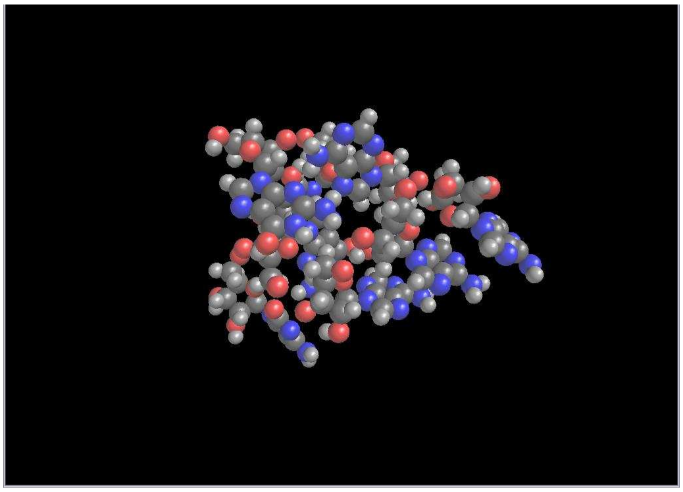


Figure 7: An example of a simulation being run on a home computer using the Folding@Home software.

Perhaps the best example of MD is Folding@Home, which is a project originating from the Pande research group at the University of Stanford [49]. Even supercomputers can only simulate up to 1 μs of the model used, so the Pande group came up with a novel solution; a free to download piece of distributed software which uses idle processor power on any home computer (or Playstation 3) it is run on. This gives the simulation access to hundreds of thousands of processors, and allows high resolution study of folding pathways.

The software works by downloading individual work units onto the users' computer, which the program experiments on before sending the results back to the main server and receiving another work unit. The amount of idle processor power which is allocated to the program is optional, and the simulation being run can be seen by the user, see Figure 7 for an example. In 2007 the Folding@Home software receiving a Guinness World Record for achieving a combined computing power of 1 petaflop, equivalent to 1 quadrillion floating point operations per second. At the time, it was only utilising 670,000 computers [50], but at the time of writing, there are over 1.36 million registered users, each of which may be using any number of computers to run the software.

This incredible amount of processing power has lead to over 70 papers being published at time of writing, which can be found through the project’s website [49]. One of the projects’ biggest successes was the characterization of the folding pathway of the Villin Headpiece; a small, 36 residue alpha-helical protein [51]. As the folding time of this protein is in the region of 10 μs, previous simulations had not been able to follow the entire folding process. However, using results from the Folding@Home software, the group were able to simulate a total of nearly 500 μs, achieving many complete examples of the folding pathway. At the time, the software was only being used on approximately

200,000 computers, but an accurate description of the folding process was still obtained. A movie of the folding process may be seen at [52], with stills demonstrating the folding pathway shown in Figure 8.

Normally explicit-solvent all-atom MD algorithms can simulate events in the range of 10^{-9} to 10^{-8} s for typical proteins [53]. Folding@Home uses between 5600 and 6000 molecules per simulation, with each simulation lasting 500 μs. The fact that this simulation was accurately done including explicit water molecules is further evidence to the power of distributed software in protein folding research.

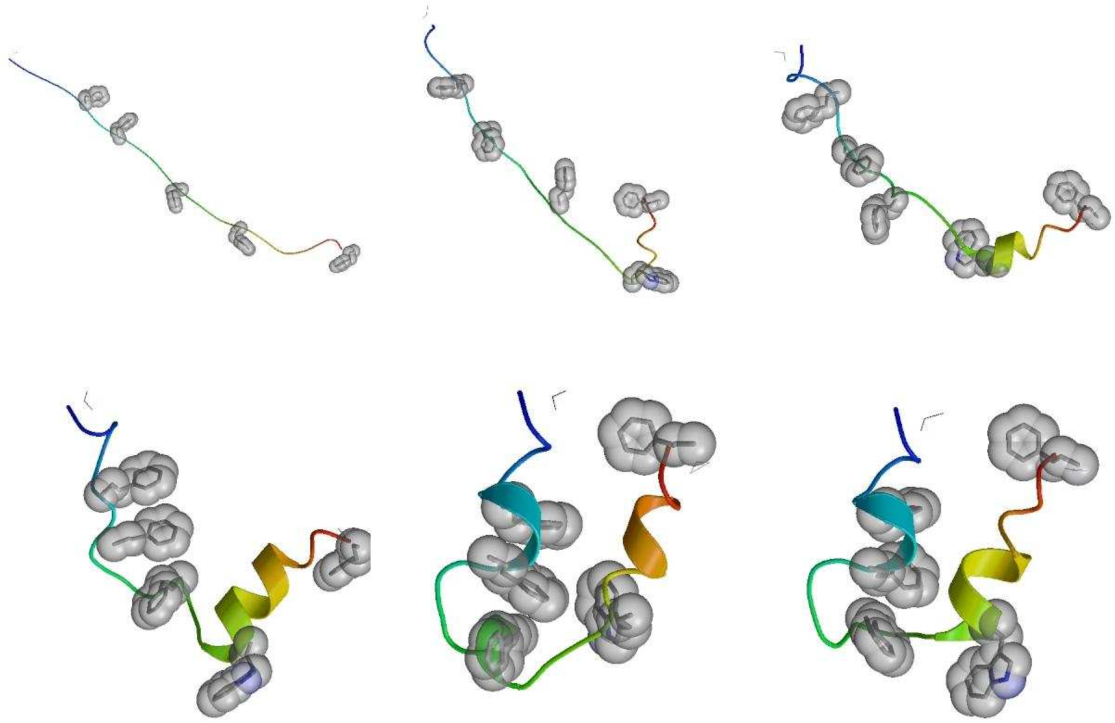


Figure 8: Stills from a movie of the folding process of the Villin Headpiece as simulated using Folding@Home.

3.2. MONTE CARLO METHODS

Monte Carlo Methods are a class of methods that, in the context of protein folding, can be used as the machinery for many different ways of exploring the statistical properties of a given model (including location of the global minimum energy). A detailed overview of the principles behind the methods is given in [54][55][56]. Monte Carlo methods are stochastic and rely on randomly sampling states from the system under consideration.

Many of the Monte Carlo algorithms used to explore energy landscapes are based on the Metropolis Monte Carlo method (MMC). This method, originally presented in [57] is a general technique allowing sampling from an arbitrary probability distribution. For proteins, this usually (but not always) means sampling from Boltzmann distributed conformations. MMC creates a Markov chain (a chain of states, where a given state depends only on the previous one) of states that has the following properties [56]:

- Ergodicity: namely that it is possible for the system move between any two possible states, given enough steps.
- Detailed Balance: Requiring that $p_a P(a \to b) = p_b P(b \to a)$ where p_a is the probability of the system being in state a and $P(a \to b)$ is the probability of the transition from state a to state b.

If these two conditions are satisfied, then the chain of states created will, if run for a sufficiently long time, will be governed by the intended distribution.

The actual algorithm is then constructed as follows:

1. Select a new state for the system
2. Accept or reject the new state with probability:

$$P(a \to b) = \text{Min}(1, \exp\left(-\frac{E_a - E_b}{kT}\right)) \tag{10}$$

Where E_a is the energy of state a etc. The chain of selected states then forms the Markov chain. As long as the trial moves (selecting the new states) are chosen such that the ergodicity condition is satisfied this will generate, given enough time, conformations with Boltzmann distributed probabilities. The issue of generating suitable trial moves is discussed in [58] where it is shown that for a model specified by angles, small changes in angles at the centre of a medium sized protein will result in large changes in distances at the edge of a protein. As such the authors propose restricting moves such that the protein can only ‘wiggle’, i.e. considering the location of the coordinate before adjusting it.

We can see from equation (10) that if the new state is lower energy, it is automatically accepted. Though moves to lower energies are always accepted, the probability of moving to higher energy allows the process to escape local minima. The choice of temperature clearly is important, with low temperatures causing the system to be reluctant to escape from local minima (the quasi-ergodicity problem) and high temperatures causing the system to accept many high energy states. Because of this, there have been many adaptations of the Metropolis algorithm.

4. ACCELERATED MOLECULAR DYNAMICS

Accelerated Molecular Dynamics is a class of methods attempting to lessen the computing power required for MD simulations. The protein folding process very quickly settles on a local minimum and moving out of this state can have a very low probability and as such are termed rare-events. Accelerated molecular dynamics can increase the likelihood of these rare-events taking place in a multitude of ways. The two ways described here are metadynamics and hyperdynamics, but these are not the only two available [59][60][61].

The number of variables defining a particular conformation for a model protein is large. Collecting these many variables is a key part of accelerated MD. The large numbers of variables are grouped into a finite number of dependants called collective variables (CV's) referred to as $s(x)$. The free energy landscape is then a function of these collective variables, $F(s)$. In protein folding, these are often taken to be gyration radius, backbone H-bonds, contact order etc. with the choice of these being a key part of these simulations.

4.1. METADYNAMICS

Metadynamics is a method used to explore a free energy surface (FES), using a time evolving artificial potential to influence the direction of the system. Gaussians are deposited at the location of the simulation, thus forming a new potential, which the simulation takes into account when moving to the next location, discouraging the simulation from revisiting sites [62].

A useful analogy is made by considering a ball sitting in an empty swimming pool. A small pile of sand is deposited at the location of the ball with the ball then rolling to a new position based on the position of this sand. Repeating this process will eventually fill the pool with sand. Given a record of how much sand had been deposited, and where, it is then possible to determine the shape of the pool. This history dependant potential is given by a sum approximating the following integral,

$$V_G = \int_0^t dt' \frac{w}{\tau_G} \exp \frac{-\left[s(x) - s\big(x(t')\big)\right]^2}{2\delta\sigma^2} \tag{11}$$

where $\delta\sigma$ is the width of Gaussian, w the height and τ_G the rate of deposition [63].

For large time scales, the simulation should explore the whole of the free energy landscape and we will have that $V_G(s,t) \to -F(s)$ [63].

The number of CV's, called d here, plays an important role in the computational efficiency of the model which depends on how many Gaussians are required to fill the well, and hence is proportional to $(\frac{1}{\delta\sigma})^d$. Hence the efficiency scales exponentially with the increase in CV's. One way to solve this in basic metadynamics is to choose Gaussians of a width comparable to the width of the current local minimum, but this contradicts the main aim of metadynamics, which can only be used to explore features that are greater in size than the Gaussians used [64].

It is not only the number of CV's but the choice of these CV's that plays a large role in the effectiveness of metadynamics. There are certain properties that the CV's should possess for them to have maximum efficiency[65]: they should distinguish between all the relevant states of the system being studied, otherwise the simulation would yield no information; there should not be too many of them due to the usual computational stresses; the set must include all the relevant CV's and those that describe relatively slow events so that they are not lost within events that happen on faster timescales.

If a relevant CV is neglected then it can cause the system to get stuck in hysteretic behaviour [65]. This is best seen with a simulation on a simple z-shaped potential surface as shown in Figure 9. If the simulation starts in B and is only biasing CV1, it will begin to fill this minimum. However as CV2 is being neglected the system will not be able to make the transition to state A until the height of the history dependant potential is much greater than the true barrier. The system will keep overfilling the minimum until it can move across, wasting valuable simulation time. This is in fact a way of testing whether relevant CV's have been left out; if the free energy grows smoothly then it is likely that a complete set of CV's has been chosen.

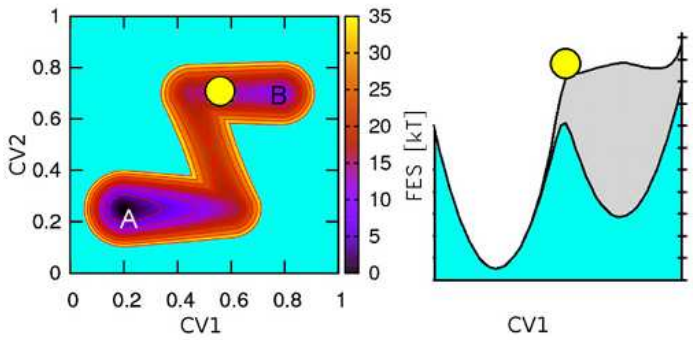


Figure 9: demonstrates the effect of ignoring a relevant CV in metadynamics. The simulation cannot make the transition to state A and so overfills minima B. From [65].

There is no method for choosing the CV’s a priori, and in nearly all cases it is necessary to use a trial and improvement to find the right combination of variables to use. Extensive studies into what CV’s can be used and situations where they are useful are further elaborated on in section [65]. There are also studies where CV’s have been designed to analyse a specific system, in this example the authors have introduced the CV’s to look specifically at protein beta-sheet structures [66]. It has been pointed out that using metadynamics can get very difficult in systems with many CV’s as is the case in protein folding.

Metadynamics has been used to determine the stability of β-sheets in prion protein present in mice [67]. It has also been used along with Monte Carlo methods to find the free energy surface of the small protein Src-SH3 [68] and on its own to explore the free energy surface of protein G Helix [69].

4.1.1 Bias –Exchange MetaDynamics

An extension to metadynamics that helps to relieve this problem is bias-exchange metadynamics. It is based upon a separate approach to other bimolecular systems, the replica exchange method (REMD) [70].

We divide the system into N non-interacting replicas that are each biased by one or two different CV’s, $s^{\alpha}(x), \alpha = 1, \dots, N$. Each replica is at the same inverse temperature, β, and each of them acquires its own history dependant potential in the same way described for metadynamics. Bias-exchange then attempts to flip the configurations between a pair of replicas, a and b, like in the replica exchange model. The probability of the move being accepted is:

$$P = \min\left(1, \exp\left\{\beta\left[V_G^{a}(x^{a}, t) + V_G^{b}\left(x^{b}, t\right) - V_G^{a}\left(x^{b}, t\right) - V_G^{b}(x^{a}, t)\right]\right\}\right) \qquad (12)$$

Here $V_G^{\alpha}(x, t) = V_G(s^{\alpha}(x), t)$. Using this each trajectory can move through the high dimensions of the free energy landscape biased by low dimensional metadynamics runs. This enables us to look through the large dimensions in protein folding using the efficient method of metadynamics, but not experience the complications described earlier when metadynamics is used on systems with many dimensions. An example where this is used in protein folding is to imagine a system where there are two relevant variables, the two dihedral angles of alanine dipeptide, φ and ψ. A swap is attempted at a time t. If the swap is accepted, the values of the CV’s simultaneously perform a jump from $\psi(x_1)$ to

$\psi(x_2)$ and from $\phi(x_1)$ to $\phi(x_2)$ while also changing the direction of each of the replicas. So where original replica one was exploring the free energy space with relation to the first dihedral angle ψ it will now be exploring using the other dihedral angle φ.

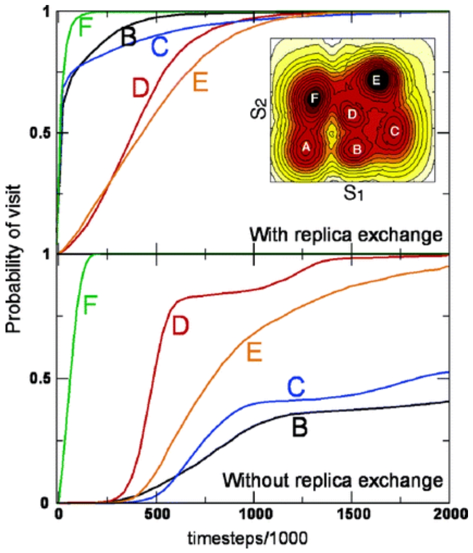


Figure 10: shows the effect of using bias exchange metadynamics on a 2D model potential. The results clearly show that BE greatly increases the probability of exploring the whole energy surface. From [63].

Bias-exchange does not give a full picture of the free energy surface in all dimensions, as in metadynamics, but instead gives several low dimensional projections of the free energy surface along all of the collective variables.

There are examples of bias exchange being used on small proteins and on some model potentials. BEMD is used on the Villin and Advillin C-terminal headpiece to find that a mutation destabilises the native fold state for the first, but the same mutation does not destabilise the native state for the latter, having important biological consequences [71]. It has also been used to find the energy surface for β-hairpin using a variety of different CV’s [72].

A model potential of 6 basins was studied, as shown in Figure 10 [63] and the bias exchange extension was tested by comparing the probability of visiting each well compared for this and normal metadynamics over a certain step size. All simulations started in well A. On the normal metadynamics, the separate CV’s, S1 and S2, are biased on two separate replicas but in the bias exchange method the simulation attempts a flip between replicas every 200 steps. Despite the advantages of metadynamics discussed earlier, the basic metadynamics simulation fails to extensively explore all the phase space mainly due to hysteresis problems. The run that biased CV S2, quickly moves into well F and then moves between A and F. The simulation biased by variable S1 also moves very quickly into well F. After the history dependant potential has built up it is able to move over the barrier and into well D, where it is able to explore both wells D and E. In these simulations, wells B and C are usually left unexplored. In the bias-exchange runs the simulation not only moves into well F quickly, but also moves over the barrier into well B and proceeds to explore the other potentials rapidly. The plots of the probabilities of visiting the wells for bias exchange and normal metadynamics is shown and this clearly shows the advantages of using the bias exchange method.

The bias-exchange extension has solved some problems with metadynamics but the problem still remains that it is very difficult to choose a set of CV's before the simulations is run, and it is still largely a case of trial and improvement. However, in this approach it is possible to choose more CV's than in standard metadynamics and so it speeds up the search of the relevant CV's[72].

4.2. HYPERDYNAMICS

Hyperdynamics [73] is another method used for speeding up rare events using a different approach that increases the value of energy minima, thus increasing the probability of escape. The energy surface is modified using an applied bias potential. This bias potential is constructed such that above a certain energy value the bias potential is the same as the true potential, and below this, it is modified.

$$V^*(r) = \begin{cases} V(r), & V(r) \geq E_B \\ V(r) + \Delta V(r), & V(r) < E_B \end{cases} \tag{13}$$

Choosing the actual form of $\Delta V(r)$ is one of the key challenges in hyperdynamics as well as choosing the best value for the bias potential [74]. One of the first potentials to use was $E - V(r)$ [75] so that the modified potential becomes$V^*(r) = E_B$ as shown in Figure 11. This gives flat potentials in the gaps that were termed as puddles. The force on this puddle is zero meaning that it is computationally inexpensive. Downsides to this method include the presence of discontinuous derivatives where the modified potential meets the real potential and the possibility of hiding transition paths beneath the modified potential.

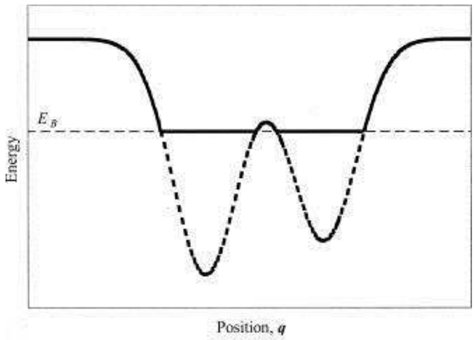


Figure 11: shows the most basic model for hyperdynamics, the puddle potential. This has the advantage of being computationally inexpensive, but has discontinuities where the modified meets the real potential and can cover some other minima. From [75].

4.2.1 Snow Drift Methods

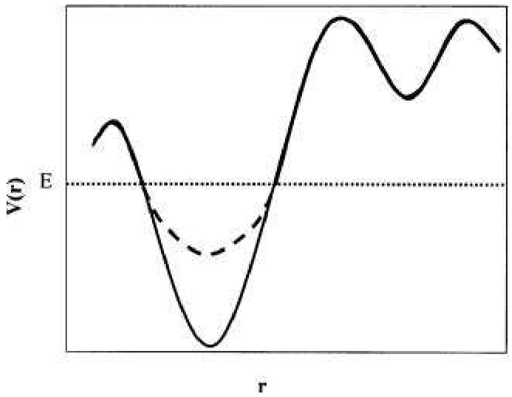


Figure 12: shows the effect of the snow drift bias potential on a model potential. From [76].

A class of methods similar to hyperdynamics are termed snow drift methods as shown in Figure 12 [76]. These introduce an applied bias potential that merges smoothly with the real potential removing the discontinuities with the additional advantage that the applied potential does not hide the shape of the minima even at high E. The applied potential ΔV is given by:

$$\Delta V(r) = \frac{(E - V(r))^2}{\alpha + (E - V(r))} \qquad (14)$$

Where α is the tuning parameter. Choosing the values of α and of E determine how aggressively the dynamics are accelerated. Figure 14 shows some choices of these parameters and their effect on a model potential. The same study tested their method on the simple protein alanine dipeptide.

In this study the variables chosen were the two backbone torsional angles φ and ψ and free energy plots were produced as functions of these variables. Initially a normal MD run was carried out and compared to three hyperdynamics simulations with varying choices for E_B, in this study called E_D, and α, as shown in Figure 13. All the simulations were carried out at 800 K so that the simulation would move efficiently over the high energy barriers. The darker regions show the potential wells. The main point to note from this figure is that the hyperdynamics simulations look very similar to the normal MD

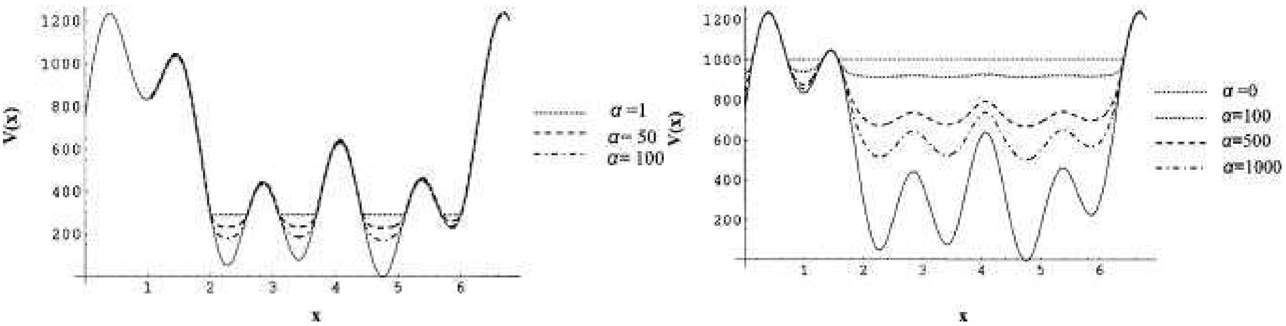


Figure 13: shows the effect of different values for E_B and α on a model potential. Low values of α have the same effect as the basic puddle potential and when used with high values of E_B minima can be hidden. However if a high value of α is chosen with a high E_B the minima can still be observed. From [76].

simulations and from this we can infer that this approach accurately samples the potential energy wells and provides the correct canonical distribution.

The values for E_D and α are a key part of getting accurate results from the simulations performed using hyperdynamics. In the same study as above, relatively high values of E_D were chosen with varying values for α and the same done for relatively low values as shown in Figure 15. With high E_D and high α, the plot looks very similar to the normal MD runs, representing an accurate simulation. The problems occur at high values of E_D along with low values of α. With low α, the potential energy surface has become effectively flat, as can be seen on the model potentials earlier. The plot has become noisy and there is little structure to it. This has arisen due to simulation experiencing a random walk on the isoenergetic surface and not sampling the potential wells properly. Alternatively, when the value of E_D is relatively low the simulations converge to the correct distributions for high and low values of α.

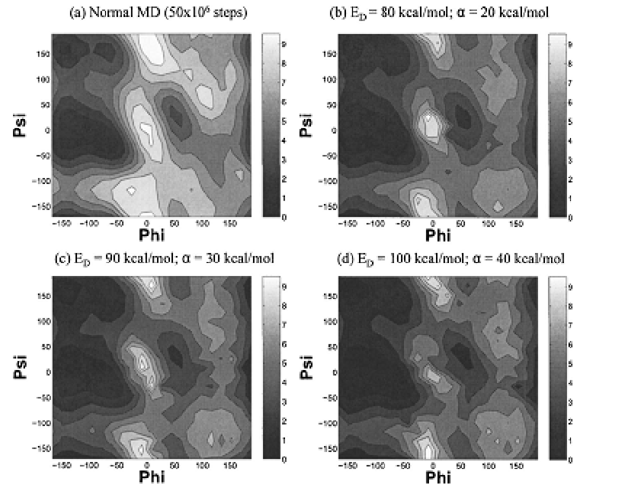


Figure 14: shows a comparison of a hyperdynamics energy surface alongside one found using normal MD simulations. In all choices of the parameters the simulations bear a resemblance to the normal MD. Showing the method is an accurate extension. From [76].

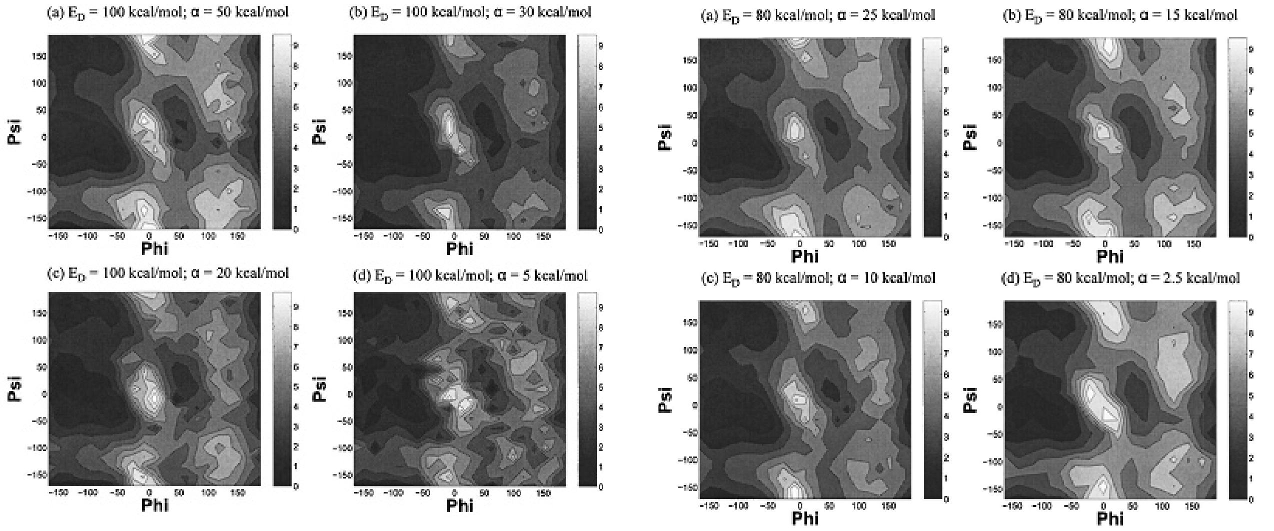


Figure 15: shows the varying choices of α and ED on the energy surface for alanine dipeptide based on the two torsional angles. With high ED and low α statistical noise dominates the surface and details are lost. Choosing the correct values of ED and α is one of the main challenges of hyperdynamics. From [76].

This is because a low value of E_D positions the surface in the potential wells and so below most of the transition states. When this is the case the choice of α is not so important, as the all the transitions states are present.

5. GLOBAL STRUCTURE SEARCHING

Previously we have been looking at the exact interactions between elements in different models to describe the protein folding process. In these sections models were used to find the native state by exploring the potential energy surface as the proteins would when folding by molecular dynamics (MD). This exact modeling of the process is however not the fastest way to find the native state, as the entire energy surface is mapped by these simulations. In this section the protein folding process itself is neglected in order to find the native state more efficiently, an approach classified as global structure searching.

This report focuses on *de novo* structure prediction or global optimization, as mentioned in the introduction. This is a method which uses Anfinsen's Hypothesis [15], that the global minimum of the energy surface corresponds to the native state of the protein, as a basis. Therefore, in this approach, the potential energy surface is minimized to find the global minimum, and thereby the native state [77]. This is the only type of global structure searching that has the potential to produce an exact solution for the structure without prior knowledge, e.g. known structures.

There are many approaches to global optimization. These approaches can all be classified as either stochastic or deterministic. Stochastic approaches can give no guarantee that the energy minimum found is the global one as they use probabilistic models, while deterministic approaches claim an amount of certainty. However, except for some branch and bound and time-dependent deformation methods, most approaches are in effect stochastic. Even those claiming to be deterministic only function with certain approximations.

5.1. Combinatorial methods

Combinatorial methods are methods that divide the optimization problem into fragments to simplify global structure searching. The energy is minimized in these fragments and subsequently the fragments are put back together to give a possible structure for the molecule [78]. The most common method that implements this approach is the so-called build-up method [79].

The build-up method initially divides the investigated protein into fragments, which may be as small as a single amino acid. Subsequently, the energy of these fragments is separately minimized to give the lowest energy minima in each such fragment. The minima that fulfill the cutoff criterion:

$$\hat{f}_\alpha + D > f_\alpha \tag{15}$$

are recorded. Here $\hat{f}_\alpha$ denotes the lowest energy minimum found in fragment α, f_α is an energy minimum in fragment α, and D is the cutoff parameter. The cutoff parameter thus determines the amount of minima that are recorded for a given fragment.

These recorded minima are used as the starting points of further energy minimization of combined fragments of the protein. In the event that the starting points of each fragment correspond to an atomic overlap (a singularity in the energy function) when adjoined, an artificial function with a finite potential replaces the energy function until minimization drives the combined fragment away from this state. This energy minimization is carried out successively using all the starting points recorded, in all combinations, while continually eliminating those minima, similar to others, but at a higher energy [79].

The main difficulty that arises in the build-up method is the determination of the cutoff parameter. Clearly the number of minima recorded grows exponentially with the cutoff parameter, so that high values of D are unfeasible to calculate. However an insufficiently high value of this parameter will cause certain minima to be overlooked, which may have a significant effect when combining fragments. Furthermore this method tends to exaggerate the importance of short-scale interactions due to the fact that minimization is carried out on small segments and adjoined. However, this method has been shown to be especially successful when proteins exhibit strong secondary structure characteristics, such as beta sheets or helices [78].

5.2. BRANCH AND BOUND

Branch and bound methods can be described as effective methods to solve inherently combinatorial problems. Instead of starting with fragments, branch and bound methods work their way downward by continuously splitting the investigated region on the potential energy surface into smaller fragments [78].

An example of a deterministic Branch and Bound method is the so-called αBB algorithm [80]. This algorithm uses lower and upper bounds to confine the global minimum into successively smaller energy ranges until the global minimum is found when the bounds meet. This is done by continually partitioning the hypersurface into smaller regions, or boxes, to explore.

The upper bound for the global minimum energy is computed by minimizing the energy at random positions on the potential energy surface. In every iteration a new position is sampled in this way, and if the sampled position has a lower energy than the existing upper bound, it replaces this as the new upper bound [78].

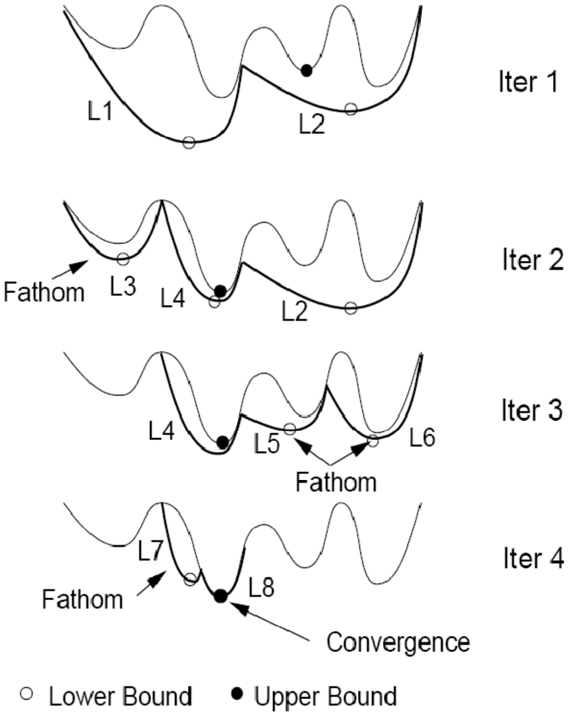


Figure 16: Example of the αBB approach to protein folding. In the first iteration the hypersurface is partitioned into two parts, CLBFs are set up for each and the minima are identified. An upper bound is found by gradient methods. As L1 is lower than L2, this region is further bisected CLBFs are fitted to each fragment. As the minimum L3 is higher than the new upper bound, this fragment is neglected (or fathomed) and region L4 is recorded. Subsequently the same is done for region L2 in iteration 3; however both CLBF minima can be fathomed as their minima are higher than the upper bound. Finally in iteration 4, region L4 is bisected and region L8 converges to the global minimum as it coincides with the upper bound. From [78]

The lower bounds pose a much greater challenge to model. These are modeled by so-called convex lower bounding functions (CLBF), which must fulfill several important criteria. CLBFs must always underestimate the hypersurface or be equal to it, and get tighter for smaller sampling regions. Furthermore, they must match the hypersurface at the box constraints and be convex within the box. These criteria ensure that the minima of the CLBFs converge to the global minimum [78]. How the algorithm uses these bounds to converge to the global minimum can be seen in Figure 16.

This algorithm has been shown to be very effective for small oligopeptide systems [81] and shows potential for the larger protein case as well, as its ability to discount large regions of the hypersurface makes it computationally effective. A challenge however still remains deriving suitable convex lower bounding functions for the larger cases.

5.3. GENETIC ALGORITHMS

Genetic algorithms are the most common of many algorithms that solve computational problems by learning from nature. They are methods that derive from natural selection and work by the three basic principles of evolution: selection, crossover and mutation [82].

Selection is the process by which a suitable partner is found. This is based on two parameters. On the one hand it depends on something called a fitness function. For a human looking for a partner this would consist of criteria such as intelligence, appearance etc. The second parameter is simply probability. In analogy this would be because the person with the highest fitness may live half way around the world.

Crossover and mutation, which can be seen in Figure 17Figure is simply the genetics of the mating process. The genes in

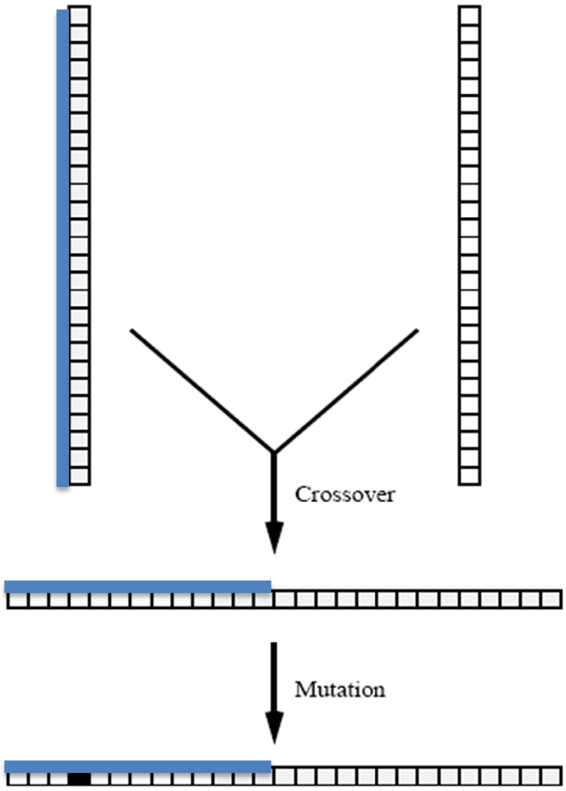


Figure 17: Schematic Diagram of the crossover and mutation mechanisms in genetic algorithms. After [78].

the chromosomes cross over to give the new chromosome of the child. Mutation is then a process that occurs additionally in evolution.

These principles are transferred to protein folding in that the genes represent the variables of the potential energy function, thus the chromosomes are a complete description of the position on the hypersurface (usually a minimum). The assigned fitness function to the chromosomes is an inverse potential energy function, so that a high fitness corresponds to a low potential energy. The processes of selection, crossover and mutation are then carried out on an initial set of minima, where the child generation successively replaces the parent generation until certain termination conditions have been met [78].

Today genetic algorithms are the basis of a lot of successful global optimization methods, of which two will be mentioned in subsection 5.7. The reason for this is that the parallel computing that these algorithms employ is a specifically efficient technique [82]. However, the often interdependent collective variables in protein folding problems create a difficulty in the exact definition of the crossover and mutation mechanisms [78].

5.4. SIMULATED ANNEALING

Simulated Annealing (SA) is a computational algorithm which simulates a crystallization process from a high temperature state to a low temperature state. Here the global minimum is essentially found by slowly freezing a sample to form the lowest energy conformation, i.e. the global minimum [83]. This can be done using either MD (c.f. section 3.1) or Monte Carlo algorithms (c.f. section 3.2).

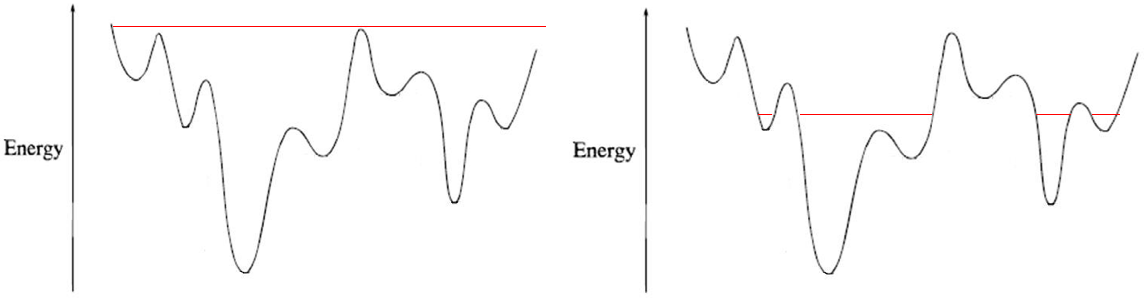


Figure 18: Diagram showing a simplification of a potential energy function explored by simulated annealing. The red line is the temperature limiting the states which can be explored by the simulation at this given temperature. In the left case the temperature is high and the search is very widespread, while in the right case the temperature is lowered, so that the simulation becomes trapped in one of the minima. After [92].

The Monte Carlo approach to simulated annealing uses the metropolis algorithm. It involves starting the system at a high temperature, running the Metropolis algorithm for a given time, and then lowering the temperature. This process is repeated until the system's state no longer changes.

The varying temperature in the algorithm means that the probability of accepting a higher energy configuration is greater at a high temperature than at a lower one. Effectively, at high temperatures the simulation can move between multiple minima basins, while as the temperature is lowered, the search is focused increasingly on the local surroundings [83]. This can be seen in Figure 18.

Using MD, the temperature of a system is modeled as the kinetic energy of the particles that are simulated. At a high temperature the particles are very fast, and therefore explore the low energy structures quickly, while at a lower temperature the structure is essentially frozen in as not enough kinetic energy is available to fold out of this structure [84].

In both methods of simulated annealing, the so-called annealing schedule plays a large role. This annealing schedule determines how many simulation moves are carried out in one temperature step, thus it determines the timescale of the cooling process. If cooling occurs too rapidly the simulation will be confined to minima basins around the starting position, while a too slow schedule takes up too much time and thus renders the simulation unfeasible [78]. SA has been proven to converge to the global minimum of energy, with sufficiently slow cooling [85], thereby allowing the method to be claimed as deterministic in the ideal case. However, a priori knowledge of the necessary cooling speed required to achieve the deterministic case cannot be known.

Further problems that arise with this approach are that the energy surface can be temperature dependent if the Free energy is modeled. In this case the global minimum that the simulation gets trapped in at high temperatures may not correspond to the global minimum at zero temperature. This makes simulated annealing only feasible if the temperature from which the global minimum is equal to the global minimum at zero temperature is lower than the temperature needed to cross the energy barrier between these two global minima at different temperatures. This can be expressed as:

$$T_{\hat{E}(T)=\hat{E}_0} \leq T_{Barrier} \tag{16}$$

Finally, multiple runs are always necessary to determine the quality of the minimum that the simulation gets trapped in at zero temperature. This is the case as the simulation may get stuck in local minima when the annealing schedule is too fast [78].

Early SA algorithms were shown to successfully generate low energy (including native) conformations of oligopeptides, from a random initial state. For example, the native structure of Met-enkephalin, mentioned in the introduction, was successfully generated using this method [86][87][88].

5.5. MONTE CARLO WITH MINIMIZATION

Monte Carlo with minimization [21] is an adaptation of MMC for global minimization. This approach [89] combines the MMC with conventional, calculus based, energy minimization [90]. This technique then generates a random walk across all energy minima. Early uses of this technique successfully calculated the native conformation of a pentapeptide. The Markov chain created with this technique does not, however, satisfy the condition of detailed balance and so cannot be used to calculate thermodynamic quantities (the Markov chain does not satisfy the Boltzmann distribution). This method was further enhanced in [91]. Where the energy is replaced with the harmonic free energy (using a harmonic approximation of the Entropy, such that the harmonic entropy of a state *i* is given by:

$$S_i^{har} = -\frac{k_b}{2}\ln|\boldsymbol{H}| \tag{17}$$

where $\boldsymbol{H}$, the Hessian, is the matrix of the second derivatives of the energy with respect to the dihedral angles.) This method was shown to improve the performance of Monte Carlo Minimization methods and was successfully applied to oligopeptides.

5.6. DEFORMATION AND SMOOTHING

Deformation and smoothing processes are approaches to the protein folding problem which attempt to simplify the search for the global minimum by simplifying the hypersurface itself. The challenge in this approach is to map the deformed hypersurface global minimum back to the original hypersurface [92]. There are many methods of doing this, of which two will be further looked into here.

One such method is the diffusion equation method (DEM) [93]. This method introduces a time parameter into the hypersurface equation, which is used to solve the diffusion equation (18), and thereby create a smoothed hypersurface.

$$\frac{\partial^2 E}{\partial x^2} = \frac{\partial E}{\partial t} \tag{18}$$

This is done with the initial condition, that at time *t=0* the deformed surface is equal to the original hypersurface, or:

$$\tilde{E}(x, t = 0) = E(x) \tag{19}$$

The power of the DEM lies in the fact that it can deform the hypersurface without prior information about its topography. It effectively reduces the number of minima, as local maxima are lowered and local minima are raised to form a smoothed surface. This has the effect that as $t \rightarrow \infty$ the hypersurface becomes flat, however just before this, it only exhibits one minimum, which is ideally the global one [93].

The problem that arises with this method is that this may not be the case if the global minimum is in a narrow energy basin, so that a broader, but shallower basin survives for a longer time [93]. Furthermore, it becomes very computationally expensive to map the smoothed surface global minimum back to the original hypersurface for complex hypersurfaces created by larger proteins [78].

A deformation method which avoids these difficulties is Basin Hopping [92]. Basin Hopping is a more recent method which does not move the minima or alter their energies, but rather uses the deformation described by equation (20) below:

$$\tilde{E}(\boldsymbol{X}) = \min\{E(\boldsymbol{X})\} \tag{20}$$

Here the deformed hypersurface $\tilde{E}(\boldsymbol{X})$ is equal to the energy of the local minimum, which is found by gradient methods at any position on the hypersurface, ***X***. This means that the energy at each point in the catchment basin of a local minimum is replaced by the energy of this minimum. This results in the hypersurface being deformed to

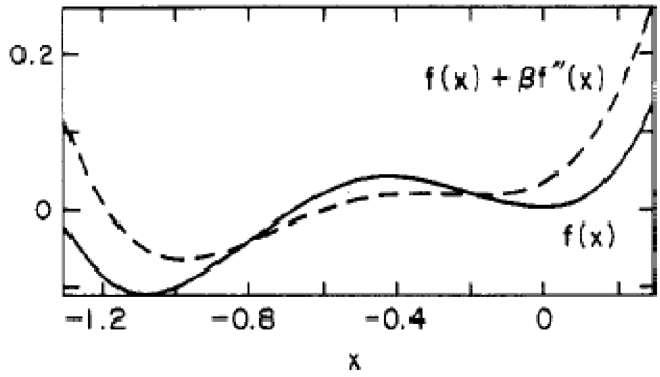


Figure 19: One dimensional example of the deformation of the hypersurface by the diffusion equation method (dashed line). It can be seen that the points of inflexion are kept the same, while the equation is transformed to reduce the number of minima. From [93].

a set of interlocking staircases as can be seen in Figure 20. This deformation eliminates the energy barriers between minima, thereby simplifying transitions between them and speeding up the searching algorithm applied to this deformed hypersurface.

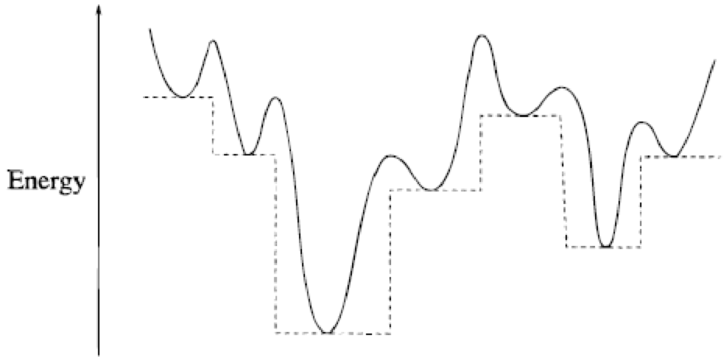


Figure 20: A schematic diagram of the effects of the energy transformation of basin hopping. The dashed line being the transformed energy. From [92]

The methods described in this section simply change the hypersurface, however do not actively look for the global minimum. Therefore these methods must be combined with a searching algorithm [13]. This can be done by either Monte Carlo methods or by molecular dynamics.

5.7. ON LATTICE METHODS

For on lattice models of proteins, (the HP model, for example), different methods become available. One such method is that of chain growth, where ensembles of chains are built such that they obey a specified ensemble [94]. Because the model is on lattice, as the chain grows there are a finite number of options for the next molecule to be placed (restricted by the condition that the chain cannot overlap). In the naïve method, based on work by Rosenbluth and Rosenbluth[95] the conformations are weighted proportionally to the number of ‘moves’ available. Later methods weight conformations with the Boltzmann weighting. The Prune Enriched Rosenbluth Method (PERM) creates conformations with this weighting. As chains are generated those with lower weight are ‘pruned’ with those remaining then increased in weight [96].The resulting ensemble of chains can be used to calculate low energy conformations in addition to an estimate of the partition function (and hence estimates of various thermodynamic quantities). Whilst these methods are efficient, they are restricted to on lattice models. A variant of the chain growth algorithm using the multicanonical ensemble (c.f. section 6) to calculate the density of states (again with an on lattice model) is demonstrated in [97]. It has been shown that one of the major weaknesses of the PERM method is folding models with a large amount of hydrophobic residues in the core [98].

5.8. REPLICA EXCHANGE

Replica Exchange Monte Carlo (also known as Parallel tempering) is a very successful protein folding algorithm and was first discussed in [99]. Usually used for energy minimization, the method involves running several copies of the protein model at different temperatures, using molecular dynamics [100] or Monte Carlo methods [101]. These different copies of the protein are then periodically exchanged with probability corresponding to a metropolis criterion where the probability of exchanging replica *i* with replica *j* is given by:

$$P_{ij} = Min(1, Exp\left(-(\frac{1}{k_b T_j} - \frac{1}{k_b T_i})(E_i - E_j\right)) \qquad (21)$$

These exchanges are usually performed with systems at similar temperature, as the probability of exchange drops dramatically with temperature difference. This algorithm outperforms many others when applied to small or simple models. Table shows the relative performance of a replica exchange algorithm against other on lattice algorithms (PERM and ACO (c.f. section 5.9) for a 3D HP model.

Sequence No.	Minimum Energy	PERM	ACO	Replica Exchange
1	-32	-32 (0.1min)	-32 (30 min)	-32 (0.1 min)
2	-34	-34 (0.3 min)	-34 (420 min)	-34 (0.2 min)
3	-34	-34 (0.1 min)	-34 (120 min)	-34 (0.1 min)
4	-33	-33 (2 min)	-33 (300 min)	-33 (0.1 min)
5	-32	-32 (0.5 min)	-32 (15 min)	-32 (0.1 min)
6	-32	-32 (0.1 min)	-32 (720 min)	-32 (0.1 min)
7	-32	-32 (0.5 min)	-32 (720 min)	-32 (0.3 min)
8	-31	-31 (0.3 min)	-31 (120 min)	-31 (0.1 min)
9	-34	-34 (5 min)	-34 (450 min)	-34 (0.9 min)
10	-33	-33 (0.01 min)	-33 60 min)	-33 (0.1 min)

Table 1: Results for global minimization with three different algorithms on the 3D HP model. 10 different residue orders (sequences) with known global minima were used. Result is given as minimum energy with time taken in brackets. Table adapted from [101].

As evident from Table 1 Replica Exchange methods outperform even the specialized on lattice algorithms. A disadvantage to this method is that it requires storage of a large number of copies of the protein in the computer memory [119], which can become difficult with large proteins, for example both [100] and [101] use simple models (an ECEPP/2 model of met-enkephalin and the on lattice HP model respectively).

5.9. FURTHER METHODS

The approaches that have been looked at in the previous subsections can be regarded as basic components of a lot of methods that are used today to perform global structure searching. Since the turn of the millennium most of the research in this area has gone towards creating improved methods based on these approaches.

A common method that has not been mentioned so far is Tabu Search [102]. Tabu Search is a general global optimization method, which introduces memory of a searching algorithm. It steers the searching function away from sites it has already visited by prohibition based on this memory. The length of the memory of a Tabu Searching algorithm is an important parameter, as a too short memory will cause the search to become trapped in minima basins too quickly, while a long memory may have the effect that the search will never get stuck and go on a random walk instead. More advanced methods of Tabu searching, such as Energy Landscape Paving [103], use punishment instead of prohibition. This punishment is a way to make the revisiting of these sites more difficult, but not impossible. It can be interpreted as a type of hypersurface deformation, which deforms minima that have been visited for the time period of the memory. Other advances of this method have also been made recently, and can be found in [104].

An example of a method, which uses the approaches mentioned in previous sections as a basis, is evolutionary algorithms [105][106]. These combine a genetic algorithm with parallel tempering. The algorithm works by simulating a population of Markov chains, each with a different temperature. The population is updated by mutation (standard Metropolis update), crossover (exchange of partial states) processes from genetic algorithms, and the exchange (exchange of full states) from parallel tempering. It has the ability to search a large area of state space due to the crossover operations, and the range of temperatures used gives it the fast mixing ability of tempering algorithms.

The evolutionary algorithm was successfully applied to a 2D HP model [107] and was shown to outperform the genetic algorithm (using 10-30% of the computational power) and regular metropolis Monte Carlo (using 1-3% of the computational power).

Other methods which combine these basic approaches are conformational space annealing (CSA) and genetic annealing. CSA combines the ideas of genetic algorithms and the build-up method. It essentially starts with a certain number of initial conformations (positions on the hypersurface), which are minimized and used as the parent generation. Crossover and mutation is simulated while conformations are successively replaced by those similar with a lower energy. A more detailed description can be found in [108]. Genetic annealing on the other hand combines simulated annealing with the use of genetic algorithms as is further described in [109] and extended in [110].

Model specific methods, specified in the on lattice method section, are also subject to further development. One such development is ant colony optimization (ACO) [111]. ACO is based on the foraging behaviour of ants, where a population of candidate conformations is repeatedly generated, based upon the relative strength of previously conformations. ACO optimisation has been applied to the HP model [112] and though its performance is not as good as the basic on lattice method PERM, it can have a slight advantage for sequences with a large amount of hydrophobic residues in the core.

All in all there are a multitude of methods, some of which are based on specific models, others on search algorithms or methods, and even more combined between the existing methods, that explaining them all is beyond the scope of this report. Mostly the methods can however be traced back to general approaches specified here.

5.10. METHOD COMPARISON

Comparing global structure searching methods is generally a very difficult task, as the systems new methods are tested on are very rarely the same. If the tested system is too simple, this cannot be a good comparison as the methods are designed to solve complex protein systems and different methods will cope differently with comparatively simple cases. However if the system is too complex, the simulation will take too long and due to this there will not be sufficient data from different methods to set up a proper comparison. Even in the case that the same simple protein is probed, there can still be very large differences, such as the model used to set up the hypersurface, i.e. the number of variables used in the energy equation. Finally if exactly the same model has been used on the same protein, the system used to simulate the problem must still be taken into account. The processing power of systems varies strongly, so that a fast processor will naturally take less time than a slow processor. To complicate this further the processing power diminishes with use, so the age and average use of a system plays a role as well, although a minor one. The entirety of data needed to make an exact comparison of methods is usually not given in research papers, so every comparison must be treated with certain skepticism. The subsequent comparison should be evaluated with this in mind [78].

Computational speeds are however not the only success criterion for comparison. As multiple runs are done in all global structure searches, the probability of finding the global minimum is an equally important criterion. Yet, in order to compare probabilities very large sample sizes are needed, as the variation may be very large. Taking into account the complexity of the problem and the already inherent difficulties in comparisons, these cannot be given.

A commonly tested protein is met-enkephalin, a five amino acid oligopeptide, mentioned in the introduction, so a form of comparison can be conducted on this sample. The table below compares the time taken to find the global minimum using various aforementioned methods, noting the number of free variables used in the model, *N*, and the computer system used. Normally this protein is modeled by using 24 combined variables, however some searches where done fixing five of these. The comparison is done by CPU processing time taken, denoted as *CPU*.

Method	Reference	Variables, N	CPU	System
Genetic Algorithms	[113]	24	2 hrs	SGI IRIS 4D/220
Simulated Annealing	[114]	24	2.5 hrs	Apollo DN 1000
Simulated Annealing	[115]	24	1.5 hrs	Apollo DN 1000
Monte Carlo Minimization	[21]	19	2-3 hrs	IBM 3090
		24	10 hrs	IBM 3090
Diffusion Equation	[116]	19	20 min	IBM 3090
αBB	[81]	24	1.3 hrs	HP–730

Table 2: A comparison of various global structure searching methods of met-enkephalin. It must be noted that the general methods are named, however small adaptations are usually made to improve these (After [78])

The Critical Assessment of Techniques for Protein Structure Prediction (CASP) (from [117]) is a biennial competition for which groups are challenged to predict the structure of a protein given only the amino acid sequence. The structures obtained by simulation are compared to experimental results and scored depending on the degree of structural agreement. The amino acid sequences given always correspond to proteins whose structure is known experimentally, but of which computational results are yet to be published. The results from the 6th such contest (in 2004) are shown in Table to give a rough overview of the different methods being used.

Model Name	**Group Name**	**Sampling Method**	**Number of high scoring folds**	**Average score For new folds**	**Average Score for all folds**
UNRES	Scheraga	Molecular Dynamics, Monte Carlo	2	16.65	24.97
ROSETTA	Baker	Monte Carlo	3	27.38	27.38
CABS	Kolinski-Bujinicki	Replica Exchange Monte Carlo (REMC)	4	25.18	25.18
REFINER	Boniaki_pred	REMC	2	17.46	19.64
FRAGFOLD	Jones-UCL	Simulated Annealing	4	24.43	27.49
CAS	Skolnick-Zhang	REMC	3	23.97	23.97
PROTINFO	Samudrala AB	Monte Carlo Simulated Annealing	2	14.02	23.97

Table 3: Partial results from the 2004 CASP, showing the model, method score of various groups. The score is the percentage of residues whose difference from the ideal position is below a certain cutoff. The two categories of scores reflect the fact that some groups attempted prediction of a selected subset of new fold targets. Table adapted from [**118**].

6. Generalized and Extended Ensembles

These are general methods to improve the performance of both Monte Carlo and Molecular Dynamics simulations. In statistical mechanics, the probability that a system in the canonical ensemble will be found in a state with energy E is given as:

$$P(E) = \frac{1}{Z} g(E) Exp(-\frac{E}{kT}) \qquad (22)$$

Where g(E) is the density of states (spectral density) or degeneracy, in the case of systems with discrete energy levels, and Z the partition function. The basic idea behind generalized ensembles is to change the Boltzmann factor in the expression for P(E) to an artificial one [119]. It is important to note that in general this artificial expression need not be a priori known, and can be calculated using an iterative procedure such as [120]. The main advantage of these methods is that they enable one to specify how ‘easy’ it is for the simulation to cross energy barriers, and so can counter the main weaknesses of the metropolis algorithm, such as getting stuck in local minima.

Umbrella Sampling [121] was the first description of this method, with its early applications to protein models proving successful. See [122] for an application to protein models using molecular dynamics.

One of simplest examples of a generalized ensemble is adaptation of Tsallis statistics [123] for use in an MMC algorithm. Here the Boltzmann factor is replaced with the following weight, $w(E)$:

$$w(E) = (1 - (q-1)\frac{E}{kT})^{\frac{1}{1-q}} \tag{23}$$

Which reduces to the Boltzmann distribution in the limit $q \to 1^{+}$. This has the effect of raising the probability of the Markov chain moving to higher energy states. This ensemble was shown to give significant improvements over regular MMC when used in a Simulated Annealing algorithm [124].

Other similar ideas, whereby $w(E)$ is proposed such that higher energy states are more likely, have been detailed, and applied to protein folding. These methods involve making a transformation in $P(E)$ where $E \to f(E)$, for some function f [125].

Most of these simple ensembles increase the likelihood of sampling high energy states. This makes them suitable for calculating thermodynamic quantities with reweighting procedures [126].This can be done using the following equation:

$$< A > (T) = \frac{\int A(E)P_g(E)w_g^{-1}(E)e^{\frac{-E}{k_bT}}dE}{\int P_g(E)w_g^{-1}(E)e^{\frac{-E}{k_bT}}dE} \tag{24}$$

Where A is the quantity to be obtained, $P_g(E)$ is the probability to find a state with energy E in the generalized ensemble and $w_g(E)$ is the modified weight. These ensembles can also be incorporated into a simulated annealing algorithm.

Multicanonical Sampling uses a different ensemble. In this method, the new weight is chosen to be the reciprocal of the density of states, such that *P(E) = const*. This allows a random walk on the energy surface, and so the simulation will not get trapped in any minima [127]. This method is an example of the aforementioned case where the weight is not a priori known. In this particular case it is possible to use algorithms [128] specifically designed to calculate the density of states in order to compute the weighting.

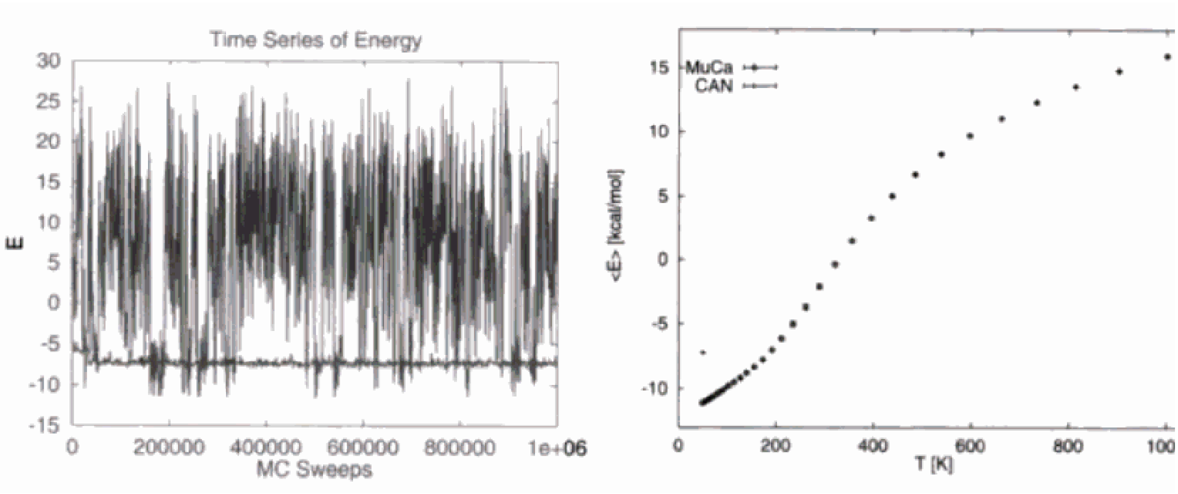


Figure 21: Time series of energy for a multicanonical and regular (canonical) MMC simulation of met-enkephalin using the ECEPP/2 model. The multicanonical results show a random walk in energy whilst the canonical simulation gets stuck in a local minimum. Right: Results of multicanonical simulations reweighted to calculate $< E >$ (T) for the same model. The single outlier is an equivalent point for the canonical MMC simulation, showing its relative weakness. From [127].

This algorithm performs an MMC (with a dynamic density of states) in fixed energy ranges. The amount of states in each energy range is continually monitored, with the density of states being updated for the next iteration. This step forms part of using the Multicanonical Monte Carlo method, and can often take a significant proportion of the computational effort for a simulation. Simulations using the multicanonical ensemble with molecular dynamics, with similar results, have also been done [129].

Two very similar techniques exist called 1/k sampling [130], where the probability of finding a state with a given entropy is made constant, and adaptive umbrella sampling [131] [132], which focuses on reaction coordinates (c.f. section 8).

An algorithm with similarities to both simulated annealing and the multicanonical ensemble is simulated tempering, which was first introduced in [133]. Simulated tempering treats temperature as a dynamical variable in a similar way to 1/k sampling such that temperatures are sampled uniformly. This technique is very useful for calculating free energies of proteins, though reweighting techniques do have to be used. For an application to a model protein see [134]. Simulated tempering has also been applied to the AB model in [135]. Many more examples of generalized ensemble algorithms exist, for a recent review see [136].

7. DISCONNECTIVITY GRAPHS

As has been stated already in the introduction to this report, simply finding the global minimum of the hypersurface, according to Anfinsen's hypothesis [15] (c.f. section 5), is not sufficient to determine the structure of a protein. According to Levinthal's paradox [24] (c.f. section 1) the time taken for a protein to fold into a minimum energy state plays an equally important role in structure determination. This is the case as if folding into the global minimum is kinetically hindered, then this structure becomes less likely than other low energy minima, into which folding can occur quickly.

While a full MD simulation of a protein will yield the folding pathway, methods exist to find path knowing only the start and end points. One such method used to investigate this is by creating a disconnectivity graph to show the topography of the energy landscape [137]. These graphs can be used to qualitatively investigate the folding kinetics, however lack the information to model these accurately.

Disconnectivity graphs are set up by mapping the partitioning of the potential energy surfaces at different temperatures or energies as can be seen in Figure 22 [138]. This means that the energy needed for transitions between basins are investigated. Here the energy or temperature step used to map the hypersurfaces determines the resolution or complexity of the graph. It must be noted that a high resolution graph often exhibits too much information and may obscure significant features. Further information can be obtained from [137].

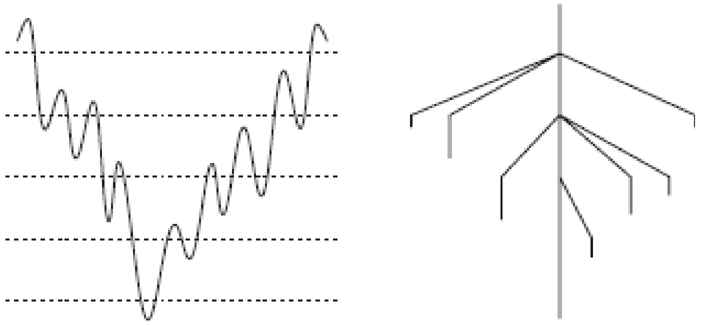


Figure 22: Schematic diagram showing the correlation between an energy landscape and the corresponding disconnectivity graph. The vertical axis represents energy or temperature, while the horizontal axis has no representation. The dotted lines correspond to the temperatures at which the partitioning of the energy surface was mapped. The disconnectivity graph is a steep funnel. From [138].

It has been postulated that a funneled disconnectivity graph as shown in Figure 22**,** is important for protein folding [139].This means that a disconnectivity graph which shows a rough energy landscape, exhibiting multiple funnels or a lack thereof, will exhibit multiple stable structures [140], while a funneled disconnectivity graph will exhibit only the native state as the global energy minimum. It is in this way that disconnectivity graphs can be used to predict whether the structure found by global optimization is valid. However, calculations of the probabilities for different structures can only be done by Transition Dynamics investigations.

8. Folding Pathways

Most of the methods so far are aimed at determining the minima of a system, this section will deal with the modelling the folding pathway. There are two main questions to ask about protein folding paths [141]:

- How does a protein fold?
- How often/quickly does the reaction occur, i.e. what is the rate constant?

While a rate constant can be measured through experiments, it is difficult for experiments to describe the process of folding in atomic detail. Hence experiments primarily yield quantitative information on the rate of folding while little on the dynamics of the folding itself [142]. Despite the ability to measure them experimentally, theoretical determination of rate constants is a worthwhile endeavour, allowing conformation of the model’s accuracy and predict rates for those reactions that are difficult to measure amongst other things.

The question as to how a protein folds can be separated into four smaller questions, namely:

- How many typical reaction channels exist?
- Which transition states and metastable states are visited by the system on its way from the reactants to the products?
- What are the prominent characteristics of the reaction mechanism?
- Is it possible to describe the reaction by a small set of reaction coordinates?

While MD can, in principle, simulate the entire folding event, this is usually computationally infeasible. One of the worst situations for MD to study is the transition between two stable states with a high energy barrier between them. As transition rates decrease exponentially with the activation barrier height, the expectation time for an event can exceed current computer capabilities by many orders of magnitude [143]. While the fastest proteins fold on the microsecond to millisecond time scale, atomistic molecular dynamics simulations are typically constrained to the nanosecond time scale [142]. This results in almost every path simulated using MD being a fluctuation about a stable state [144].

8.1. TRANSITION STATE THEORY

Any reaction can be defined by progress along a reaction coordinate, which is an abstract collective variable representing progress along a reaction pathway. Consider a system with 2 stable states A and B separated by a high energy barrier. One method of investigating this kind of system is to look at the dynamical bottleneck, which is the part of the reaction that limits the rate. For a specific reaction this can be defined by a value of the reaction coordinate. In protein folding this is equivalent to the "transition state". In relation to the energy surface, this is the top of a saddle point between the two stable states A and B. Once the system has reached this point, the system will almost certainly fall into one of the stable states within a computationally feasible timeframe. To analyse this bottleneck, assuming the location of the transition state is known, one moves the system reversibly from the initial state to the bottleneck finding the probability of reaching this bottleneck, then sets off many short trajectories from that state [144][145].

From these trajectories the probability of falling into either A or B from the transition state is then found.

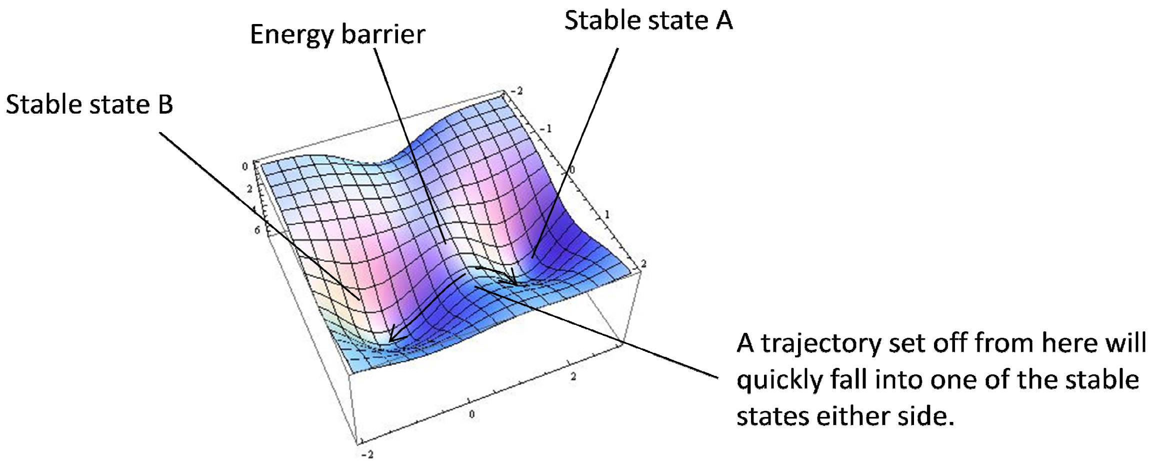


Figure 23: Diagram showing ideal energy surface for TST, the energy barrier is known and not too long so that the protein can be defined as in each stable state once over the barrier. Due to the height of the energy barrier, plain MD would have trouble analyzing this as it would take a lot of computational power waiting for a process of interest to happen. This is because most of the time a protein will simply fluctuate around its stable state.

This information allows the calculation of the probability of transition, which, combined with information on how long each path takes, can easily be converted to a rate constant i.e. the rate for the rare event. This rate describes the number of transitions, which occur per second for a given concentration of proteins and leads to the macroscopic speed of folding [144]. Another, equivalent, way of visualising this is to measure the effective flux through the transition state [143].

A Comprehensive analysis of this method can be found at [146] and [145].

8.2. TRANSITION PATH SAMPLING

To circumvent the requirement of prior knowledge of the transition state, Monte Carlo algorithms have been developed for sampling the ensemble of all transition pathways [147][141]. This method, named Transition Path Sampling, focuses entirely on the rare events of interest, but requires no prior knowledge of how the transition occurs (e.g. reaction coordinates) but can extract them from the results. The path action $S_{AB}[\{x\}]$ is defined via a relation to the probability that a path occurs, which itself is given as the product of the probabilities for each step on the path. The path of least action is then the most probable [141].

8.2.1 SAMPLING METHODS

The aim of these methods is to efficiently sample the ensemble of all paths linking the protein in its original state A to the protein in its final folded state B of a given length $\mathcal{T} = L\Delta t$. This is achieved through the use of a metropolis algorithm with acceptance criterion:

$$P_{acc}^{o\to n} = \min\left[1, \frac{e^{-S_{AB}[\{x^n\}]}P_{gen}^{n\to o}}{e^{-S_{AB}[\{x^o\}]}P_{gen}^{o\to n}}\right] \tag{25}$$

Here $P_{gen}^{o\to n}$is the probability of generating the new path from the old one and $P_{acc}^{o\to n}$is the probability of accepting the new path [141]. To find new random paths two main algorithms are used: Shooting and Reptation [141]. Both of these algorithms rely on already having a known transition path, but this can be any path linking A and B, as, like the metropolis criterion drives the system to probable regions of the energy surface, it also moves the path sampling to probable regions of path space after some "equilibration period" [144].

Another necessity for these algorithms is that the stable states need to be well defined; this is done via "order parameters". These must fill two main criteria: First they must fully distinguish between the stable states, if they do not then the path simulation will tend to produce paths that simply stay in one of the stable regions as these paths are of high probability [141]. Second, the order parameters must be characteristic for the stable states. When the system is in one of the stable states the order parameters should be within the boundary definitions. Otherwise, not all relevant paths will be sampled. These requirements can be fulfilled only by trial and error [141]. Problems with choosing an order parameter unwisely are explored in [148].

8.2.1.1 SHOOTING

The shooting algorithm takes a known path between A and B, and chooses a random time slice τ along this path. At this time slice, the point in phase space is $x_\tau = \left(r(\tau), \rho(\tau)\right)$, i.e. the position and momentum at time τ. Shooting takes this point and runs an MD simulation forwards in time. Due to the inherently chaotic and random system that describes a protein folding; this new path quickly diverges from the old path [141]. The dynamics are computed forward until time step L, if the path is in stable state

B at this point, the path is accepted based on the Metropolis acceptance criterion, otherwise shooting begins again. This creates a new path consisting of the time slices 0 to $(\tau - 1)$ of the old path and the newly generated time slices τ to L. This path is possible due to the stochastic nature of the dynamics [141].

This procedure samples the area around B well, but the area around A is sampled very slowly. This problem is solved by doing the same algorithm on the path but with reversed momenta. The action of reversing momenta to find a backward path was shown to be valid in [141].

8.2.1.2 REPTATION

Reptation is used to improve the equilibration of TPS and consists of removing a random amount of time steps from one end of the path and adds time steps to the other end. If the ends are still in the stable states A and B then the new path is accepted with the Metropolis criterion as above.

Reptation obviously does not affect any time slice in which the system is on the barrier between A and B, so it cannot be used to sample completely new transition paths. It is therefore used to supplement the shooting algorithm. Whereas shooting creates new paths, reputation simply shifts the path back and forth in time [141].

These algorithms produce a collection of transition paths called the transition path ensemble or TPE from A to B from which results may be drawn.

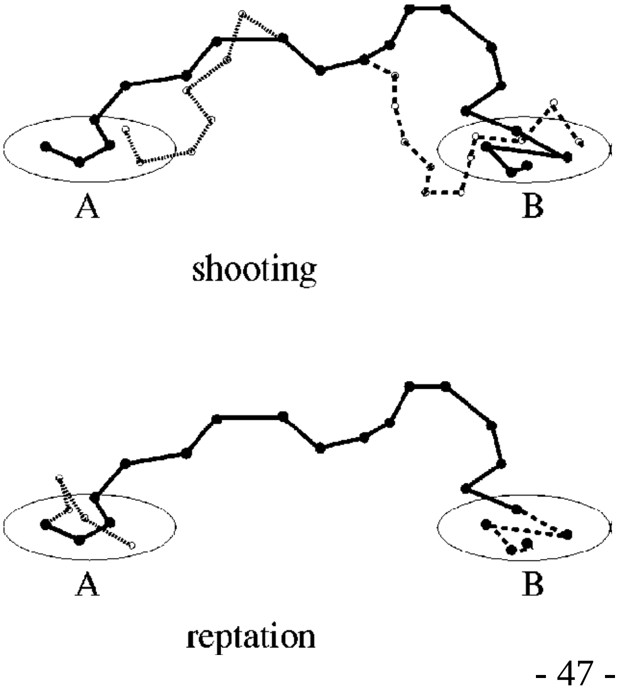


Figure 24: This picture shows the sampling algorithms method: The shooting algorithm creates new paths by shooting a path off from a randomly selected time slice either forward (dashed line) or backward (dotted line). Reptation effectively shifts the path in time by deleting time slices from one side of the path and adding new time slices onto the other side. The previous metropolis acceptance criterion ensures that paths are sampled according to their probability. From [141].

8.2.2 ANALYSIS OF THE TPE

The TPE is essentially a large collection of paths from A to B. The two main questions (how quick does the folding happen, and what is the mechanism of the folding) can be answered by analysis of this TPE.

8.2.2.1 PATH QUENCHING

The action of a path is related to the path energy by:

$$S_{AB}[\{x\}] = \beta U_P[\{x\}] + C \tag{26}$$

Where C is a constant depending on the time increment Δt, and the friction, while $U_P[\{x\}]$ is the "path energy" (depending on the path but not on temperature).

As paths of low action are favoured, this formula implies that so are paths of low energy, these can be thought of as local minima in path space, hence most paths are simply variations around some local minima [141].

Typically, paths close to each other share the same basic properties. So we can therefore regard these least energy paths as a representative for an ensemble of similar paths [141].

In a complex system there may be several distinct sets of paths connecting regions A and B. Path quenching uses the local path energy minima as representatives of all of the paths in the set. As a result, one reduces the original large set of transition paths to a small set of least action paths, which can be easily examined and viewed. This is most useful if the fluctuations around the minimum action paths are small. At low temperature, this condition is met, and the classification of paths by quenching is an effective way to explore the path ensemble [141].

8.2.2.2 RATE CONSTANTS

A Rate constant, k can be calculated via the equation:

$$k(\tau) = \nu(\tau) \cdot P \tag{27}$$

Where $\nu(\tau)$is the frequency factor and is expected to reach a plateau after the molecular time τ_{mol} when all the proteins being considered have relaxed into local minima, leaving the rate constants to be dependent on the energy barriers and P is the probability of finding the protein in state B. This plateau value is the value used in the calculation. This can be found through a single simulation. Conceptually it's approximately how often states are changing and is similar to the TST probability of finding the system at the transition state [141].The frequency factor, $\nu(\tau)$ can be found in one path sampling simulation from its definition, while the probability factor P usually involves several independent path simulations through umbrella sampling. Looking at the calculation of rate constants in TPS and TST, and using the analogy of fluxes through surfaces you can show that in the limit $t \to 0^+$ the equation for rate constants in TPS is the same as in TST [143]. TPS has been used to look at DNA polymerase β [149]

8.2.2.3 USING THE TPE TO FIND TRANSITION STATES

As previously stated, TPS does not require thorough knowledge of the transition state, but it can be used to find states in the transition state ensemble, possibly leading to a good reaction coordinate; this is done by looking at "committors" and the "separatrix". The committor, is the probability of fleeting trajectories started at configuration x to end in state A a short time, t_s, later. The committor can be used to examine a dynamic bottleneck if the short time, t_s, is of the order of the commitment time, $\tau_{mol}[142]$. This is done via recording which state the system falls into each time. If the probability of falling each side is 50/50 then the state can be thought of as a transition state. The set of these points is called the separatrix [144].

For a system with few enough dimensions, the set of points where the probability to fold is one half is the location of the saddle points of the energy surface [144]. This is the simplest definition of transition states. However, this definition can also be applied to systems of higher dimensions and hence is useful [150].

The reaction coordinate has been found for an enzymatic reaction using Transition Path Sampling in [151]

Despite the fact that TPS is designed around analysing the transition on the ideal energy surface between two stable states, it can be adapted to look at longer transitions with some stable states in between. The folding pathways of the C-terminal β-hairpin of protein G-B1 are examined by using TPS on three separate transitions involving three separate transitions [152].

One problem with TPS is that, like the Monte Carlo metropolis criterion can get stuck in local minima of structure energy, the algorithms for finding paths can get stuck in local minima of Path energy. Hence TPS has been adapted for this by adding a bias to move the system out of local minima in [153].

While TPS is valid and has more applicability than TST, it has limited use for protein folding in this form because the dynamics of many proteins have long, diffusive energy barriers, causing many paths to shoot off into irrelevant parts of phase space [141]. Another Problem is that, when reaction pathways are complex and exhibit multiple recrossings, the typical timescales for a transition can be relatively long. In that case the TPS rate constant calculation is computationally expensive, as the path length $L\Delta t$ must exceed these time scales [143]. In an attempt increase the efficiency of simulation in these circumstances, alternative schemes have been made around basic TPS framework the major being Transition Interface Sampling (TIS) and a further modification to this called Partial Path Transition Interface Sampling (PPTIS).

8.3. TRANSITION INTERFACE SAMPLING

Transition Interface Sampling (TIS) is based on a flux calculation, which can be compared to an alternate way of looking at TST and TPS presented in [143]. Its major advantage over TPS in that it measures the effective positive flux instead of a conditional general flux (flux through a surface but with a condition applied to it). Hence, only the positive terms contribute to the calculation, leading to a quicker convergence to the rate constant.

To look at the flux, a reaction coordinate or order parameter must be chosen, however this requires knowledge of the system and a bad choice of this can lead to a rate constant to be much too high [143]. TIS solves this by introducing a series of interfaces through which to measure the flux. This gives a value related to the probability factor of TPS rate constants but with less computational power needed. The method is similar to umbrella sampling [143]. To formulate a proper flux the phase space is divided into nonintersecting interfaces where the effective flux through an interface with the condition that the paths originated from A is the rate constant [143]. By recursively integrating a relation of the flux through one interface to the flux through the interface one closer to A, the rate constant can be found. Note that the final rate constant k_{AB} is independent of the choice of the interfaces as long as the first and last one are inside the basin of attraction of the stable states A and B, respectively [154].

TIS is a two step procedure, the first step is to find the effective flux out of A, which is possible by running an MD simulation at stable state A and counting the number of effective crossings of the interface. For statistical accuracy the interface must be chosen near to A [143]. The second part of the calculation is to evaluate the conditional probability that a trajectory coming from A crosses interface λ_i provided that it has passed interface λ_{i-1} for each interface λ_i (this is the relation that is recursively integrated). This involves sampling all paths from A that cross λ_i and then either go back into A, or eventually (possibly after recrossing λ_i) cross λ_{i+1} [143]. The Sampling of these paths is effectively the shooting algorithm but abandoning the path as soon as it reaches either λ_1 or λ_{i+1}. As the paths are not limited by time length, moving the path forward or backward in time (the reptation algorithm) is unnecessary [143]. The Major benefits of this method as compared to TPS is that, by allowing variable path lengths, the number of time steps needed to be computed are drastically reduced, also, by only counting the effective positive flux terms, TIS is less sensitive to recrossings and has better convergence [143].

TIS has been combined with a modification of TPS (accepting paths to multiple stable states) in a novel manner in [155].

8.4. PARTIAL PATH TRANSITION INTERFACE SAMPLING

Partial Path Transition Interface Sampling (PPTIS) is a method devised to limit the computational expense of TIS by relying on the assumption that a path does not have a "memory" of where it's been before. This assumption has led to a linear scaling in computational effort with the width of a diffusive barrier as opposed to the quadratic scaling of TIS [156].

PPTIS has been developed mainly to investigate processes with highly diffusive characters; including protein folding in water. These processes have to overcome free energy barriers that are relatively flat and wide, but are still of a rough nature. For such energy barriers the stochastic nature of the system causes shooting paths to diverge before the basins of attraction have the chance to guide the paths to the proper stable state [156].

PPTIS uses an illustrative example of a 1D system with a high energy barrier consisting of a series of metastable states. If the rate of reaching the energy barrier is low and the system loses memory of how it has got into each metastable state then it can be shown that the rate of folding only depends on the probability of going each way out of a metastable state and the rate of reaching the energy barrier [156].

This system represents the PPTIS assumptions, but with the metastable states substituted with the interfaces of TIS, and the relaxation into each metastable state represented by the complete loss of memory of where the system has been previously.
For further details see [156].

In contrast to the TIS technique, which requires the interfaces to be close enough to obtain good statistics, the interfaces should be sufficiently far apart in the PPTIS method to ensure complete loss of memory [156].

8.5. PROBABILISTIC ROADMAPS

Another method for investigating the dynamics of trajectories is to form a "probabilistic roadmap" across the energy surface [157][158]. This method assumes knowledge of the native state and creates nodes (or points on the surface) outwards from this state through a random Monte Carlo walk. Once this is done, each node is joined to its nearest neighbours with a straight path, or "road" (Figure 25(a)). Each road is assigned a probability using the same method of assigning probabilities to transition paths. [158]

If enough nodes have been generated this “roadmap” will connect the native state to every region of the energy surface (see Figure 25(b)). Once this roadmap is known, the native state can be joined to any unfolded configuration on the energy surface. If a desired unfolded state is not already a node then it is simply added [158]. Hence, finding the path of greatest probability throughout this road system (using Dijkstra’s algorithm) [159], finds a representative path from the unfolded configuration to the native state [158]. This is representative of the paths that occur in nature in the same way that paths of least action are representative of the fluctuations about them in path quenching, though the paths along a roadmap have no advantage over any other path (see Figure 25(c)).

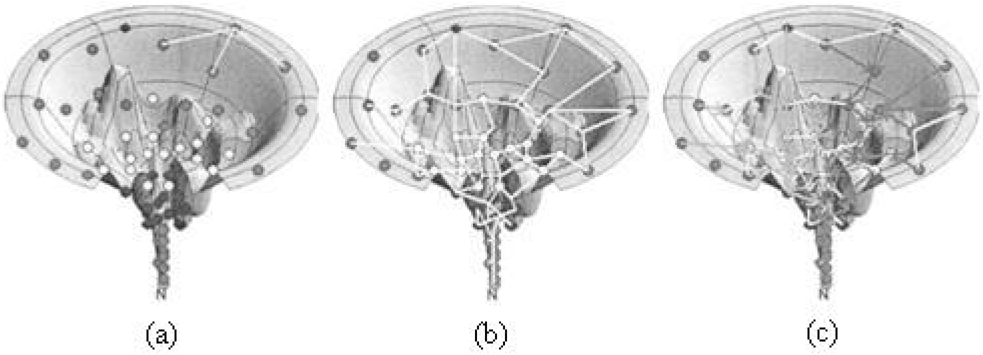


Figure 25: (a, b): Roadmap connection. (c): extraction of folding pathways shown imposed on a visualization of the potential energy landscape, where **N**, seen at the bottom of each graph, denotes the native structure [158].

While this method computes multiple folding pathways in one run and is hence very useful, the quality of the pathways, given by their probability, is poor. Furthermore, this method does not include the time taken for the transition to be made and hence rate constants cannot be found [158]. To counteract this, the method has been combined with TPS to form a road map which includes transition times and has more kinetically relevant paths, as they are formed using MD [142].

9. Conclusion

The models and methods mentioned in this report outline the most important aspects of the protein folding problem. While the general aim of computational modeling of protein folding is to obtain the function of proteins by structure determination, most of the abovementioned methods tackle this problem from different angles and starting points, so that the determination of the structure is rather a cooperation of these approaches, than a competition between them.

When the folding of a specific protein is modeled, the approach that is initially used depends on the one hand on the sequence length of the protein, i.e. the number of amino acids, and on the other hand on computational limitations, such as processing power and time. In the case of small polypeptide simulations where limitation due to processing power and time are negligible, brute force molecular dynamics with explicit solvation is used as it is the most exact method available. The aforementioned limitations are however very real, making all-atom simulations unfeasible especially for larger proteins.

It is here where accelerated molecular dynamics (MD) and global structure searching methods become relevant. For larger proteins, brute force molecular dynamics becomes excessively slow and computationally expensive, so that accelerated MD is used to map the hypersurface more quickly. This is done by methods that prevent MD simulations from getting stuck in minima, which however comes at the cost of folding path information.

When processing power is more sparse global structure searching (GSS) methods are applied to find the native state of a protein. These methods generally ignore path information to search for regions of low energy more effectively. Generally, GSS however cannot provide any assurance that a minimum is the global one due to general use of Monte Carlo methods, and more specifically the Metropolis Algorithm. The specific GSS method used depends on the model and the eye of the beholder, in that there is no general preference, due to the difficulty of comparing methods.

Finally, due to Levinthal’s paradox the folding kinetics must be investigated by looking at the transition dynamics. Thereby determining whether the global minimum found by the previous methods is correct, or the folding process into this state is kinetically hindered. These methods use minima obtained from path investigations such as hyperdynamics and brute force MD, thereby completing the circle of the interlocking methods investigating protein folding.

REFERENCES

[1] T. Creighton, *Protein Folding*, (W.H. Freeman, New York, 1992)

[2] M. Cox, G. Phillips Jr. *Handbook of proteins: structure, function and methods*, (Wiley, Chichester, 2007)

[3] M. Ridley, *Genome*, (Harper Perennial, New York, 2006).

[4] C. Opitz et al., *Proc. Natl. Acad. Sci. USA* **100** (22), 12688 (2003)

[5] E. Corder et al, Science **261**, 921 (1993)

[6] J. Drenth, *Principles of Protein X-Ray Crystallography*, (Springer, Berlin, 2006)

[7] K. Wüthrich, *Angew. Chem. Int. Ed. Engl.* **42**, 3340 (2003)

[8] Protein Data Bank, http://www.pdb.org (retrieved 8.04.2010)

[9] A. Šali, *Mol. Med. Today* **1** (6), 270 (1995)

[10] J. Ugaonkar, R. Baldwin, *Nature* **335,** 694 (1988)

[11] A. Lesk, C. Chothia, *Philos. Trans. R. Soc. London Ser. B* **317**, 345 (1986)

[12] X. Qu, R. Swanson, R. Day, J. Tsai, *Curr. Protein Pept. Sci.* **10** (3), 270 (2009)

[13] D.J. Wales, H.A. Scheraga, *Science* **285**, 1368 (1999)

[14] C. Floudas et al., *Chem. Eng. Sci.* **61**, 966 (2006)

[15] C. B. Anfinsen. *Science* **181**, 223 (1973)

[16] D. Fischer et al., *FASEB J.* **10**, 126 (1996)

[17] .N. Lewis, H.A. Scheraga, *Arch. Biochem. Biophys.* **144**, 576 (1971)

[18] D. Wales, *Energy Landscapes*, (Cambridge University Press, Cambridge, 2003)

[19] D. Wales, *Phil. Trans. R. Soc. A* **363**, 357 (2005)

[20] T. Schlick, *Optimization Methods in Computational Chemistry*, Reviews in Computational Chemistry **3**, 1 (VCH Publishers, New York, 1992)

[21] Z.Q. Li, H.A. Scheraga, *J. Mol. Struct. (Theochem)* **48**, 333 (1988)

[22] D.J. Wales, *Curr. Opin. in Struct. Biol.* **20**, 3 (2010)

[23] R. Unger, J. Moult, *Bull. Math. Biol.* **55** (6), 1522 (1993)

[24] C. Levinthal, *J. Chim. Phys.* **65**, 44 (1968)

[25] R. Zwanzig, A. Szabo , B. Bagchi,. *Proc. Natl. Acad. Sci. USA* **89** (1), 20 (1992)

[26] K. Dill, H. Chan, *Nat. Struct. Biol.* **4**, 10 (1997)

[27] B. Roux, T. Simonson, *Biophys. Chem.* **78**, 1 (1999)

[28] W.L. Jorgensen et al., *J. Chem. Phys. **79***, 926 (1983)

[29] C. Clementi, *Curr. Opin. Struct. Biol.* **18**, 10 (2008)

[30] N. Gō, *J. Stat. Phys.* **30** (2), 413 (1983)

[31] F. Stillinger, T. Head-Gordon, C Hirshfield, *Phys. Rev. E* **48** (2), 1469 (1993)

[32] L. Huang, W. Huang, *J. Comput. Sci. Tech.* **22** (4), 59 (2007)

[33] G. A. Cox, R. L. Johnston, *J. Chem. Phys.* **124**, 204714 (2006)

[34] S. Brown, N. Fawzi, T. Head-Gordon, *Proc. Natl. Acad. Phys. Sci.* **100** (19), 10712 (2003)

[35] V. Pande, *Folding@Home – Science*, http://folding.stanford.edu/English/Science (retrieved 20.02.2010)

[36] T. Huang et al, *Proc. Natl. Acad. Phys. Sci.* **101** (21), 7960 (2004)

[37] S. Auer et al., *PLoS Comput. Biol.* **4** (11), 1 (2008)

[38] S. Auer, C. Dobson, M. Vendruscolo, *Hum. Front. Sci. Program J.* **1** (2), 137 (2007)

[39] K. Roternman, M. Lambert, K. Gibson, H.A. Scheraga, *J. Biomol. Struct. Dyn.* **7** (3), 421 (1989)

[40] B.R. Brooks et al, *J. of Comput. Chem.* **4** (2), 187 (1983)

[41] W.D. Cornell et al., *J. Am. Chem. Soc.* **117**, 5179 (1995)

[42] Y.A. Arnautova et al., *Proteins: Struct., Form. & Bioinf.* **77**, 38 (2009)

[43] H. A. Scheraga, *Chem. Rev.* **71**, 195 (1971)

[44] M. Allen, *Introduction to Molecular Dynamics Simulation,* NIC Series **23**, ed. N. Attig et al., 2 (Gustav-Stresemann-Institut, Bonn, 2004)

[45] T. Schlick, J.P. Mesirov, *Mathematical Approaches to Biomolecular Structure and Dynamics,* The IMA Volumes in Mathematics and its Applications **82** (Springer, Berlin, 1996)

[46] L. Verlet, *Phys. Rev.* **159**, 98 (1967)

[47] A, Liwo, M. Khalili, H.A. Scheraga, *Proc. Natl. Acad. Sci.* **102** (7), 2362 (2005)

[48] J McCammon, JB Gelin, M Karplus, *Nature* **267**, 585 (1977)

[49] V. Pande, *Folding@Home*, http://folding.stanford.edu/English/Main (retrieved 20.02.2010)

[50] V. Pande, *Folding@Home Awards*, http://folding.stanford.edu/Russian/Awards (retrieved 11.03.2010)

[51] G. Jayachandran, V. Vishal, V. Pande, *J Chem Phys* **124**, 164902 (2006)

[52] V. Pande, *Villin,* http://www.stanford.edu/group/pandegroup/folding/villin/ (retrieved 20.02.2010)

[53] R. Day, V. Daggett, *Adv. Protein Chem.* **66**, 373(2003)

[54] J. Liu, *Monte Carlo Strategies in Scientific Computing,* (Springer, New York, 2001)

[55] D. Landau, *A guide to Monte Carlo simulations in statistical physics 2nd Ed*, (Cambridge University Press, Cambridge, 2005)

[56] M.E.J. Newman, G.T. Barkema, *Monte Carlo Methods In statistical Physics*, 84 (Oxford University Press, New York, 2002)

[57] N. Metropolis et al., *J. Chem. Phys.* **21**, 1087 (1953)

[58] S. Cahill, M. Cahill and K. Cahill, *J. of Comput. Chem.* **24** (11), 1363 (2002)

[59] E. Darve and A. Pohorille, *J. Chem. Phys.* **115**, 9169 (2001)

[60] T. Huber, A. Torda, W. van Gunsteren, *Comput. Aided Mol. Des.* **8**, 695 (1994)

[61] S. Marsili and et al., *J. Chem. Phys. B,* **110**, 14011 (2006)

[62] A. Laio, M. Parrinello, *Proc. Nat. Acad. Sci. USA* **99** (20), 12562 (2002)

[63] S. Piana, A. Laio, *J. Phys. Chem. B* **111**, 4553 (2007)

[64] A. Laio, et al., *J. Phys. Chem. B* **109** (14), 6714 (2005)

[65] A. Laio, F. Gervasio, *Rep. Prog. Phys.* **71**, 126601 (2008)

[66] F. Pietrucci, A. Laio, *J. Chem. Theory Comput.* **5** (9), 2197 (2009)

[67] A. Barducci et al., *J. Am. Chem. Soc.* **128,** 2705 (2006)

[68] F. Marini et al., *Gene* **422,** 37 (2008)

[69] C. Camilloni et al., *Proteins: Struct., Funct., Bioinf.* **71**, 1649 (2007)

[70] Y. Sugita, A. Kitao, Y. Okamoto, *J. Chem. Phys.* **113** (18), 6042 (2000)

[71] S. Piana, *J. Mol. Biol.* **375**, 460 (2008)

[72] G. Bussi et al., *J. Am. Chem. Soc.* **128**, 13435 (2006)

[73] A. Voter, *Phys. Rev. Lett.* **78** (20), 3908 (1997)

[74] A. Voter, *J. Chem. Phys.* **106** (11), 4665 (1997)

[75] A. Rahman, C. Tully, *J. Chem. Phys.* **116** (20), 8750 (2002)

[76] D. Hamelberg, J. Mongan, J.A. McCammon, *J. Chem. Phys.* **120** (24), 11919 (2004)

[77] A. Liwo et al., *Proc. Natl. Acad. Sci. USA* **96**, 5482 (1999)

[78] C.A. Floudas, J.L. Klepeis, P.M. Pardalos, *Global Optimization Approaches in Protein Folding and Peptide Docking,* DIMACS Series in Discrete Mathematics and Theoretical Computer Science **47**, ed. F. Roberts, p.141 (American Mathematical Society, Providence, 1999)

[79] K.D. Gibson, H.A. Scheraga, *J. Comput. Chem.* **8**, 826 (1987)

[80] C.S. Adjiman, I.P. Androulakis, C.A. Floudas, *Comput. Chem. Eng.* **21**, S445 (1997)

[81] I.P. Androulakis, C.D. Maranas, C.A. Floudas, *J. Global Optim.* **11**, 1 (1997)

[82] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning,* 1 (Addison-Wesley, Sydney, 1989)

[83] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, *Science* **220**, 671 (1983)

[84] M. Levitt, *J. Mol. Biol.* **170**, 723 (1983)

[85] V. Granville, M. Krivanek, J.P. Rasson, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **16**, 652 (1994)

[86] H. Kawai, T. Kikuchi, Y. Okamoto, *Protein Eng.* **3**, 85 (1989)

[87] Y. Okamoto, T, Kikuchi, H. Kawai, *Chem. Lett.* **7**, 1275 (1992)

[88] Y. Okamoto, *Recent Res. Dev. Pure Appl. Chem.* **2**, 1 (1998)

[89] Z. Li, H.A. Scheraga, *Proc. Natl. Acad. Sci. USA* **84**, 6611 (1987)

[90] D.M. Gay, *ACM Trans. Math. Software* **9**, 503 (1983)

[91] M. Vasquez, E. Meirovitch, H. Meirovitch, *J. Phys. Chem.* **98**, 9380 (1994)

[92] D.J. Wales, J.P.K. Doye, *J. Phys. Chem. A* **101**, 5111 (1997)

[93] L. Piela, J. Kostrowicki, H.A. Scheraga, *J. Phys. Chem.* **93**, 3339 (1989)

[94] U.H.E. Hansmann, Y. Okamoto, *Curr. Opin. Struct. Biol.* **9**, 177 (1999)

[95] M. Rosenbluth, A. Rosenbluth, *J. Chem. Phys.* **23**, 356 (1955)

[96] H. Frauenkron et al., *Phys. Rev. Lett.* **80** (14), 3149 (1998)

[97] M. Bachmann, W. Janke, *Phys. Rev. Lett.* **91**, 208105 (2003).

[98] A. Shmygelska, H.H. Hoos: *BMC Bioinformatics* **6,** 30 (2005)

[99] R.H. Swendsen, J.S. Wang, *Phys. Rev. Lett.* **57**, 2607 (1986)

[100] Y. Sugita, Y. Okamoto, *Chem. Phys. Lett.* **314**, 141 (1999)

[101] C. Thachuk, A. Shmygelska, H.H. Hoos, *BMC Bioinf.* **8**, 342 (2007)

[102] F. Glover, *Interfaces* **20**, 74 (1990)

[103] U.H.E. Hansmann, L.T. Wille, *Phys. Rev. Lett.* **88**, 4 (2002)

[104] X.L. Zhang, W. Cheng, *An Improved Tabu Search Algorithm for 3D Protein Folding Problem*, Lecture Notes in Computer Science **5351**, ed. T.B. Ho, Z.H. Zhou, 1 (Springer, Berlin, 2008)

[105] F. Liang, W.H. Wong, *Stat. Sin.* **10**, 317 (2000)

[106] F. Liang, W.H. Wong, *J. Chem. Phys.* **115**, 3374 (2001)

[107] K.A. Dill, *Biochemistry* **24**, 1501 (1985)

[108] J. Lee, H.A. Scheraga, S. Rackovsky, *J. Comput. Chem.* **18,** 1222 (1997)

[109] X.L. Zhang, X.L. Lin, *Protein folding prediction using an improved genetic-annealing algorithm*, Lecture Notes in Computer Science **4304**, ed. A. Sattar, B.H. Kang, 1196 (Springer, Berlin, 2006)

[110] X.L. Zhang et al., *Genetic-annealing algorithm for 3D off-lattice protein folding model*, Lecture Notes in Computer Science **4819**, ed. T. Washio, 186 (Springer, Berlin, 2007)

[111] M. Dorigo, V. Maniezzo, A. Colorni, *IEEE Trans. Syst., Man, Cybern. B* **26** (1), 29 (1996)

[112] A. Shmygelska, H.H. Hoos, *An improved ant colony optimization algorithm for the 2D HP protein folding problem*, Lecture Notes in Artificial Intelligence **2671**, ed. Y. Xiang, B. ChaibDraa, 400 (Springer, Heidelberg, 2003)

[113] S.M. Legrand, K.M. Merz, *J. Global Optim.* **3**, 49 (1993)

[114] L.B. Morales, R. Guardunojuarez, D.Romero, *J. Biomol. Struct. Dyn.* **8**, 721 (1991)

[115] L.B. Morales, R. Guardunojuarez, D.Romero, *J. Biomol. Struct. Dyn.* **9**, 951 (1992)

[116] J. Kostrowicki, H.A. Scheraga, *J. Phys. Chem.* **96**, 7442 (1992)

[117] Protein Structure Prediction Center, http://predictionscenter.gc.ucdavis.edu (retrieved: 10.03.10)

[118] D. Gront et al., *Template-free Predictions of Three dimensional Protein Structures: From First Principles to Knowledge-based Potentials* in *Prediction of Protein Structures, Functions and Interactions*, ed. J. Bujnicki, 117 (Wiley & Sons, Chichester, 2009)

[119] Y. Iba, *Int. J. Mod. Phys. C* **12** (5), 623 (2001)

[120] U.H.E. Hansmann, Y.Okamoto, J. Comput. Chem. **14**, 1333 (1999)

[121] G. Torrie, J.Valleau, *J. Comput. Phys.* **23**, 187 (1977)

[122] N. Nakajima, H. Nakamura, A. Kidera, *J. Comput. Chem.* **101**, 817 (1997)

[123] C. Tsallis, *J. Stat. Phys.* **52**, 479 (1988)

[124] U.H.E. Hansmann, *Phys. A* **242**, 250 (1997)

[125] W. Wenzel, K. Hamacher, *Phys. Rev. Lett.* **82** (15), 3003 (1999)

[126] A. Ferrenberg, R. Swendsen, *Phys. Rev. Lett.* **61** (23), 2635 (1998)

[127] U.H.E. Hansmann, *New Algorithms and the Physics of Protein Folding* in *New Directions in Statistical Physics*, ed. L. Wille 176 (Springer, Berlin, 2004)

[128] F. Wang, D. Landau, Physical Review Letters 86, 2050 (2001)

[129] T. Terada, Y. Matsuo, A. Kidera, *J. Chem. Phys.* **118**, 4306 (2003)

[130] B. Hesselbo, R. Stinchcombe, *Phys. Rev. Lett.* **74** (12), 2152, (1995)

[131] M. Mezei, *J. Comp. Phys.* **68** 237 (1987)

[132] R.W.W. Hooft, B.P. van Eijck, J. Kroon, J. Chem. Phys. **97**, 6690 (1992)

[133] E. Marinari, G. Parisi, *Europhys. Lett.* **19**, 451 (1992)

[134] A. Irback, F. Potthast, *J. Chem. Phys.* **103** (23), 10298 (1995)

[135] A.Irback, E. Sandelin, *J. Chem. Phys.* **110**, 12256 (1999)

[136] Y. Okamoto, *J. Mol. Graphics Modell.* **22**, 425 (2004)

[137] O.M. Becker, M. Karplus, *J. Chem. Phys.* **106**, 1495 (1997)

[138] D.J. Wales, M.A. Miller, T.R. Walsh, *Nature* **294**, 758 (1998)

[139] P.E. Leopold, M. Montal, J.N. Onuchic, *Proc. Natl. Acad. Sci. USA* **89**, 8721 (1992)

[140] J.P.K. Doye, D.J. Wales, *Phys. Rev. Lett.* **80**, 1357 (1998)

[141] C. Dellago, P.G. Bolhuis, D. Chandler, *J. Chem. Phys.* **108**, 9236 (1998)

[142] N. Singhal, C.D. Snow, and V.S. Pande, *J. Chem. Phys.* **121** (1), 415 (2004)

[143] T.S. van Erp, D. Moroni, P.G. Bolhuis, *J. Chem. Phys.* **118** (17), 7762 (2003)

[144] P.G. Bolhuis et al., *Annu. Rev. Phys. Chem.* **53**, 291 (2002).

[145] P. Pechukas, *Annu. Rev. Phys. Chem.* **32**, 159 (1981)

[146] D.G. Truhlar, R.E. Wyatt, A*nnu. Rev. Phys. Chem.* **27**, 1 (1976)

[147] C. Dellago et al., *J. Chem. Phys,* **108**, 1964 (1998)

[148] B.M. Dickinson, *J. Chem. Phys.* **131**, 74108 (2009)

[149] R. Radhakrishnan, T. Schlick, *Proc. Natl. Acad. Sci. USA* **101**, 5970 (2004)

[150] R. Du et al., *J. Chem. Phys.* **108**, 334 (1998)

[151] S.L. Quaytman, S.D. Schwartz, *Proc. Natl. Acad. Sci. USA* **104** (30), 12253 (2007)

[152] P.G. Bolhuis, *Proc. Natl. Acad. Sci. USA* **100** (21), 12129 (2003)

[153] D. Zahn, *J. Chem. Phys.* **123**, 44104 (2005)

[154] D. Moroni, T.S. van Erp, P.G. Bolhuis, *Phys. Stat. Mech. Appl.* **340**, 395 (2004)

[155] J. Rogal, P.G. Bolhuis, *J. Chem. Phys* **129**, 224107 (2008)

[156] D. Moroni, P. G. Bolhuis, T.S. van Erp, *J. Chem. Phys,* **120**, 4055 (2004)

[157] N.M. Amato, G.J. Song, *Comput. Biol.* **9**, 149 (2002)

[158] N.M. Amato, K.A. Dill, G. Song, *J. Comp. Biol.* **10**, 239 (2003)

[159] T.H. Cormen, C.E. Leiserson, R.L. Rivest, *Introduction to Algorithms 6th ed.*, (MIT Press, Cambridge, MA., 1992)