INFORMS Journal on Computing
Vol. 16, No. 4, Fall 2004, pp. 406–418
ISSN 0899-1499 | EISSN 1526-5528 | 04 | 1604 | 0406

DOI 10.1287/ijoc.1040.0102

Optimizing an Empirical Scoring Function for Transmembrane Protein Structure Determination

Genetha Anne Gray, Tamara G. Kolda
Computational Sciences and Mathematics Research Department, Sandia National Laboratories, PO Box 969 MS 9159, Livermore, California 94551-0969, USA {gagray@sandia.gov, tgkolda@sandia.gov}

Ken Sale, Malin M. Young
Biosystems Research Department, Sandia National Laboratories, PO Box 969 MS 9951, Livermore, California 94551-0969, USA {klsale@sandia.gov, mmyoung@sandia.gov}

We examine the problem of transmembrane protein structure determination. Like many questions that arise in biological research, this problem cannot be addressed generally by traditional laboratory experimentation alone. Instead, an approach that integrates experiment and computation is required. We formulate the transmembrane protein structure determination problem as a bound-constrained optimization problem using a special empirical scoring function, called Bundler, as the objective function. In this paper, we describe the optimization problem and its mathematical properties, and we examine results obtained using two different derivative-free optimization algorithms.

Key words: optimization; computational biology; nonlinear programming; parallel algorithm; protein structure; Bundler scoring function
History: Accepted by Harvey J. Greenberg, Guest Editor; received June 2003; revised February 2004; accepted May 2004.

1. Introduction

In this study, we formulate the transmembrane protein structure determination problem as a bound-constrained nonlinear optimization problem,

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & \mathcal{L} \le \mathbf{x} \le \mathcal{U}, \end{aligned} \tag{1}$$

where $f\colon \mathbb{R}^n \to \mathbb{R}$ is a nonlinear function; $\mathbf{x}, \mathcal{L}, \mathcal{U} \in \mathbb{R}^n$; and $\mathcal{L}$ and $\mathcal{U}$ are given lower and upper bounds on $\mathbf{x}$ respectively. In this application, the objective function f is an empirical scoring function designed to rate the validity of proposed transmembrane protein structures. The variable $\mathbf{x} \in \mathbb{R}^n$ represents the spatial positions of certain components of the transmembrane protein, and the bounds $\mathcal{L}$ and $\mathcal{U}$ are derived using observed properties of these components.

There is a wide variety of optimization methods available for finding a solution to (1). However, the effectiveness and efficiency of these algorithms can be application specific. Hence, answering the question of which to use is not easy. In this paper, we examine the transmembrane protein structure identification problem and its model formulation. We consider two different optimization algorithms that are appropriate for this application. We discuss why we chose these two methods and compare and contrast numerical results for a transmembrane protein of known structure.

This paper is organized as follows. In §2 we discuss the biological significance of transmembrane proteins and the importance of determining their structures. Then, in §3 we describe the mathematical formulation of the transmembrane protein structure determination problem and give some details of the scoring function. Section 4 reviews the basic characteristics of the optimization methods that we chose to apply to the problem, motivates their use, and gives the details of their implementations. The results of our numerical study are presented in §5. Finally, in §6 we summarize our work and draw conclusions.

2. Biological Background

Approximately one third of the proteins encoded for by a typical genome are transmembrane proteins (Buchang et al. 2002), and they participate in many important cellular processes. Some transmembrane proteins form a channel through which certain ions and molecules can enter or leave the cell. Others act as signal transduction receptors or play roles in cell recognition, senses mediation, or cell-to-cell communication. Many diseases are the result of transmembrane protein malfunction, absence, or mutation. Hence, these proteins are an important target of drug design. In fact, a large percentage of the current pharmaceuticals act on transmembrane proteins

(Wilson and Bergsma 2000). Additional information about the structure and function of transmembrane proteins can be found in texts such as Brandon and Tooze (1999), Banaszak (2000), Creighton (1992), and references therein.

Like all proteins, a transmembrane protein is a macromolecule consisting of a chain of amino acids. The defining characteristic of a transmembrane protein is that this chain traverses the cell membrane one or more times. For example, a G-protein-coupled receptor, one type of transmembrane protein involved in signal transduction, spans the cell membrane seven times. The portion of the transmembrane protein within the cell membrane consists primarily of hydrophobic amino acids, while the portion outside the cell membrane consists mainly of hydrophilic amino acids. These characteristics, in conjunction with the makeup of the cell membrane, dictate the overall structure of transmembrane proteins. In particular, due to the chemical environment of the membrane interior, the amino acids that are inside the cell membrane form stable secondary structures including α-helices and β-sheets. To date, two major structural classes of transmembrane domains have been observed: all α-helical and all β-stranded. We will limit the subsequent discussion to the all α-helical case and note that 20% to 30% of a genome's proteins are likely to have a transmembrane helical domain (Wallin and Heijne 1998, Krough et al. 2001). In this study, we consider a transmembrane protein to consist of a bundle of connected α-helices. Figure 1 contains an illustration of the transmembrane protein rhodopsin contained in a retina cell membrane. In this figure, the seven linked cylinders, labeled A through G, represent the seven α-helices that traverse the cell membrane. (Note that this figure was obtained from the G-protein-coupled receptor website (Hulsen and Lutje-Hulsik 2001).)

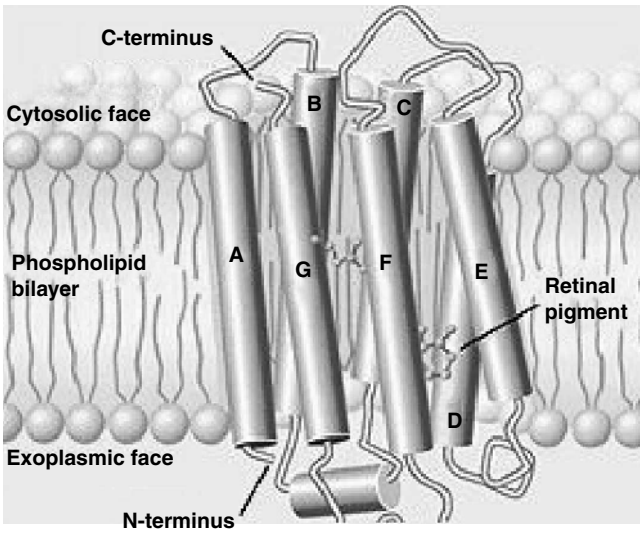


Figure 1 Illustration of the Transmembrane Protein Rhodopsin

Note. Copyright © 2001, T. Hulsen and D. Lutje Hulsik; reprinted with permission.

As of May 2004, the protein data bank (PDB) contains over 25,000 structures, and its size is increasing exponentially (Berman et al. 2000). However, the majority of the proteins found in the PDB are soluble proteins. In contrast, the structures of only about 80 transmembrane proteins have been determined (White 1998 and references therein). This is due to the fact that experimental structure determination methods such as X-ray crystallography and nuclear magnetic resonance (NMR) have been difficult to apply to transmembrane proteins. Furthermore, because so few transmembrane protein structures have been determined, few suitable templates exist for homology modeling (Herzyk and Hubbard 1998). Therefore, the development of an integrated computational/experimental model to address transmembrane protein structure and function questions is an important challenge in the field of structural biology.

The modeling of transmembrane proteins can be broken up into the separate tasks of defining the transmembrane helices and determining the relative orientation of these helices. A process known as *sliding-window hydrophobicity* is an accurate and well established method of predicting transmembrane helices given their amino acid sequences (Rose 1978; Jayasinghe et al. 2001a, b). No widely accepted method has yet emerged to subsequently ascertain the spatial locations of these helices. Because the cell membrane imposes certain structural constraints on the positions of the helices and thus limits the number of possible structures, several ab-initio computational approaches have been proposed (Bowie 1999, Nikiforovich et al. 2001, Vaidehi et al. 2002). One such procedure is based on the fact that the conformational space of membrane proteins can be effectively sampled and enumerates all the possible helical bundles (Bowie 1999). However, this method neglects the orientations of the individual helices around their respective axes. Several other promising methods have been specifically designed for G-protein coupled receptors but have yet to be validated for other transmembrane proteins (Nikiforovich et al. 2001, Vaidehi et al. 2002, Dobbs et al. 2002).

3. Transmembrane Protein Structure Determination

In Faulon et al. (2003), Sale et al. (2004), a novel two-step approach for determining the spatial location of the transmembrane protein helices is proposed. The first step, described in detail in Faulon et al. (2003), involves searching the conformational

space of transmembrane proteins to find a set of candidate helical bundles matching some given experimental distance constraints. The second step refines and reduces this set of bundles via optimization techniques. Using the structures obtained in step one as starting points, solutions to problem (1) are sought, where the objective function f assigns a score to each candidate helical arrangement that indicates how similar it is to the true structure. The minimization problem of step two is the focus of this paper.

3.1. Mathematical Description of the Problem

In this study, determining the structure of the transmembrane protein focuses on describing the relative orientation of the helices, or how they bundle. Each helix is assumed to be a rigid body, so we describe its position in space using its center of mass and a line segment defined by the two points centered in the terminal turns of the helical ends. We define a three-dimensional reference space for each helix using its initial center of mass and initial helix axial line segment. In other words, the position of each helix is defined in terms of its original location. Then, the variables in (1) are merely the x, y, and z translations from the original centers of mass of each helix and the x, y, and z rotations about the initial helix axial line segment for each helix. This is illustrated in Figure 2. Hence, a transmembrane protein with m helices has $6m$ variables. At this time, we do not consider the loops that connect the helices as part of the structure determination but note that they can be added via existing techniques after the helical positions have been established (Vriend 1990, Xiang and Honig 2001).

Most of the $6m$ variables have simple bounds that derive from the fact that transmembrane proteins reside in the cell membrane. The restrictions on the x and y rotations of each helix are based on the survey of helix tilt angles given in Bowie (1997). The z rotational variables have no such limitations and are allowed to vary in the entire z-rotational space. Both the x and the y translations are confined to a space that is approximately one third of the total radius of the membrane protein. These constraints are based on the study of helix packing behavior presented in Bowie (1997). The z translation variables have the tightest bounds to restrict the helical portions of the transmembrane protein to the interior of the cell membrane.

We now need a way to compare possible structures and decide which one best approximates the transmembrane protein in question. If the structure were known, such comparisons could be made simply using root mean square deviation (RMSD), a way of comparing two protein structures by calculating the sum of the distances of comparable atoms; see, for example, Leach (2001). However, the overall goal of this work is to identify unknown transmembrane protein structures, so we must develop another technique. We use a penalty scoring function known as *Bundler* to rate each structure (Sale et al. 2004). Bundler measures how well a structure conforms to specific criteria based on experimental data and helix bundling features described in the literature, and it does not require any a priori knowledge of the location of the helices. The Bundler score is smallest for those structures that most closely meet the specified criteria. Thus, we define an objective function f for problem (1) using Bundler to give this structure a score. Therefore, minimizing f is the computational tool for determining the structure of a transmembrane protein.

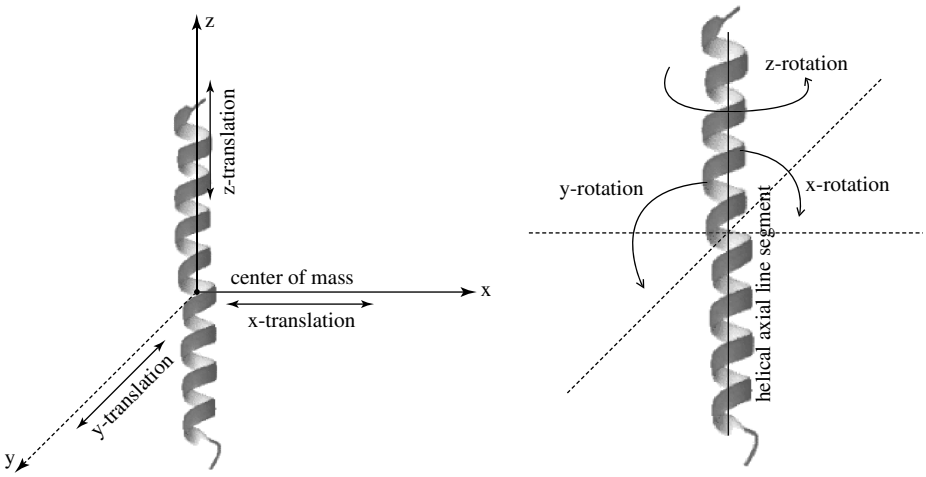


Figure 2 Depiction of the Six Positional Variables Associated with Each Helix
Note. Translations on the left and rotations on the right.

3.2. The Scoring Function: Bundler

As previously stated, the Bundler scoring function combines experimental data and topological models created from a survey of known transmembrane helix packing interactions. For each structure, the score is calculated as the sum

$$P = P_E + P_I, \tag{2}$$

where P_E quantifies the structure's violation of a set of experimental distance constraints and P_I quantifies how well the structure satisfies some helix packing parameters determined by analyzing a set of 16 nonredundant membrane proteins (Sale et al. 2004).

It has been shown that distance constraints are important in determining transmembrane protein structure. In fact, the number of possible structures decreases exponentially with the number of distance constraints available and increases exponentially with the error on the distance measures (Faulon et al. 2003). Bundler incorporates experimental distance constraints in the term

$$P_E = \sum_{(a,b)\in\Omega} K_E * \begin{cases} (d_{ab} - \ell_{ab})^2, & d_{ab} < \ell_{ab}, \\ 0, & \ell_{ab} \le d_{ab} \le u_{ab}, \\ (u_{ab} - d_{ab})^2, & d_{ab} > u_{ab}, \end{cases} \tag{3}$$

where ℓ_{ab} and u_{ab} are predetermined upper and lower bounds on the distance between atoms a and b, respectively; d_{ab} is the distance between atoms a and b in the current structure; Ω is a subset of atom pairs; and K_E is a force constant. The distance constraints ℓ_{ab} and u_{ab} are obtained from experimental methods such as chemical cross-linking, dipolar electron paramagnetic resonance (dipolar EPR) (Berliner et al. 2001), fluorescence resonance energy transfer (FRET), or NMR for assembling transmembrane helical proteins (Krishna and Berliner 1999). Note that these constraints are not procurable for every pair of atoms in the structure. Instead, experimental distance constraints are only available for a small subset, Ω, of all atom pairs.

Obtaining enough distance constraints to determine a structure uniquely is difficult, particularly for transmembrane proteins (Faulon et al. 2002, 2003). Furthermore, these distances are not error free. Therefore, to better identify desirable structures, Bundler also includes a term that measures correspondence to observed helix packing properties (determined from an analysis of known structures). This term, P_I, is actually a sum of six different terms:

$$P_I = P_\delta + P_\theta + P_\phi + P_{sc} + P_{vdw} + P_c. \tag{4}$$

Each term checks a different helical bundling property.

The packing distance score, P_δ, and packing angle score, P_θ, consider all the helical pairs in the bundle and penalize them if they are too far apart or too close together. More specifically, the packing distance score gauges how far apart two helices are in terms of their centers of mass, and the packing angle score examines the angle between two helices in terms of their axial line segments. Let Γ denote the set of $m(m-1)/2$ distinct helical pairs (i, j). Then, the packing distance score is defined as

$$P_\delta = \sum_{(i,j)\in\Gamma} K_\delta * \begin{cases} (\delta_{ij} - \delta_l)^2, & \delta_{ij} < \delta_l, \\ 0, & \delta_l \le \delta_{ij} \le \delta_u, \\ (\delta_u - \delta_{ij})^2, & \delta_{ij} > \delta_u. \end{cases} \tag{5}$$

Here, $\delta_l = \bar{\delta} - 1.5s_\delta$ and $\delta_u = \bar{\delta} + 1.5s_\delta$, where $\bar{\delta}$ and s_δ are the mean and standard deviation of the inter-helical distances, respectively, which are calculated using a set of 16 known structures; δ_{ij} is the distance between the centers of mass of helices i and j in the current structure; and K_δ is a given force constant. Similarly, the packing angle score is defined as

$$P_\theta = \sum_{(i,j)\in\Gamma} K_\theta * \begin{cases} (\theta_{ij} - \theta_l)^2, & \theta_{ij} < \theta_l, \\ 0, & \theta_l \le \theta_{ij} \le \theta_u, \\ (\theta_u - \theta_{ij})^2, & \theta_{ij} > \theta_u, \end{cases} \tag{6}$$

where $\theta_l = \bar{\theta} - 1.5s_\theta$ and $\theta_u = \bar{\theta} + 1.5s_\theta$, and $\bar{\theta}$ and s_θ are the mean and standard deviation of the inter-helical packing angles; θ_{ij} is the inter-helical packing angle between helices i and j in the current structure; and K_θ is a given force constant.

The packing density is defined as the ratio of atomic volume to solvent accessible volume (Richards 1974). It gauges how efficiently a protein folds together or, equivalently, how much interior space is left unused. The packing density score, P_ϕ, is defined as

$$P_\phi = K_\phi * \begin{cases} (\phi - \phi_l)^2, & \phi < \phi_l, \\ 0, & \phi_l \le \phi \le \phi_u, \\ (\phi_u - \phi)^2, & \phi > \phi_u, \end{cases} \tag{7}$$

where $\phi_l = \bar{\phi} - 1.5s_\phi$ and $\phi_u = \bar{\phi} + 1.5s_\phi$, and $\bar{\delta}$ and s_δ are the mean and standard deviation of the observed packing density; ϕ is the packing density of the current structure; and K_ϕ is a given force constant. It penalizes structures that are packed too tightly or too loosely.

In transmembrane proteins, it has been observed that amino acids have a preference for which amino acids they interact with on neighboring helices (Adiman and Liang 2001, Nikiforovich et al. 2001, Adamian et al. 2003). The side-chain interaction propensity score, P_{sc}, incorporates this into Bundler. It is based on the membrane helical inter-facial pairwise (MHIP) amino acid interaction propensity table in Adimand and Liang (2001), and it penalizes structures containing amino acid pairs that are in contact contrary to their normal observed behavior. Let Λ_i be the

set of Cβ atoms (Brandon and Tooze 1999) in helix i and Υ be the set of m consecutive helical pairs. (Note that two helices are a consecutive pair if they are directly connected by an outer loop.) Then, the side-chain propensity score is defined as

$$P_{sc} = \sum_{(i,j)\in\Upsilon} \left[\sum_{a\in\Lambda_i,\, b\in\Lambda_j} K_{sc} * (p - p_{ab}) \right], \tag{8}$$

where p is the maximum propensity score in the MHIP table, p_{ab} is the MHIP propensity value of atoms a and b, and K_{sc} is a constant.

To prevent inter-helical clashes, Bundler includes the van der Waals repulsive function (Brünger 1992)

$$P_{vdw} = \sum_{(a,b)\in\Lambda} K_{vdw} * \begin{cases} 0, & r_{ab} \geq sR_{ab}, \\ (s^2R_{ab}^2 - r_{ab}^2)^2, & r_{ab} < sR_{ab}. \end{cases} \tag{9}$$

Here, Λ is the set of all distinct pairs of Cβ atoms, r_{ab} is the distance between Cβ atoms a and b in the current structure, R_{ab} is the observed distance at which atoms a and b interact or repulse, s is a predetermined van der Waals scaling factor, and K_{vdw} is a given constant.

Finally, to ensure that each helix has at least two neighboring helices, Bundler includes a contact score. This piece of the scoring function guarantees that the helices are packed tightly and prevents any one helix from being excluded from the bundle. It is defined as

$$P_c = \sum_{i\in\Delta} K_c * \begin{cases} 0, & c_i \geq 2, \\ (2 - c_i), & c_i < 2, \end{cases} \tag{10}$$

where Δ is the set of helices; c_i is the number of helices that helix i is in contact with; and K_c is a given constant. Two helices are defined to be in contact if their centers of mass are within a given distance of one another. This distance bound is calculated using the analysis of the 16 known structures.

Observe that both the side-chain interaction propensity score, P_{sc}, and the contact score, P_c, introduce discontinuities in the Bundler scoring function. Moreover, P_E, P_δ, P_θ, and P_ϕ contain points at which the derivative is undefined. These properties of Bundler are worth noting as they affect our choice of optimization method. We also note that all the pieces of the Bundler scoring function contain at a least one constant as well as some predetermined bounds. Setting these parameters is an important component of the transmembrane protein structure determination problem but does not effect the optimization of Bundler and is thus not addressed in this paper.

3.3. Optimizing Bundler

In this paper, we are interested in the details of optimizing the Bundler scoring function, and so we have included only a basic description of Bundler. Further details and more specific explanations of the function's development and validation are not critical to our numerical study and can be found in Sale et al. (2004). However, we wish to make some comments and observations about Bundler in terms of our optimization goals and expectations.

First, we reiterate the fact that the Bundler scoring function incorporates real data obtained via laboratory experimentation. Hence, there is a certain amount of noise in our objective function. At present, there is no regularization term in the Bundler scoring function to prevent fitting this noise, and thus it is not productive to demand that an optimization algorithm yield a structure with a Bundler score of zero. Moreover, we have observed that small variations in Bundler scores result in only noise-level differences in the structures (Sale et al. 2004), and so we do not require a high level of accuracy from the optimization method.

Second, we remind the reader that optimizing the Bundler scoring function is the second step of a method for determining the spatial locations of the helices of a transmembrane protein. In the first step, the discrete conformational space is reduced to hundreds or even thousands of candidate helical bundles to be used as the starting points in the second step, minimizing (1). In order to attain a small number of final candidates for further study, we require a fast and efficient optimization method capable of further refining the results of step one.

Finally, it should be noted that the Bundler scoring function incorporates helix packing parameters defined using a very small sample (16 nonredundant proteins) of transmembrane helical bundles. Until this set can be dramatically increased, we do not necessarily expect Bundler to identify the true (or native) structure as the structure with the absolute lowest score. Instead, we have designed the Bundler scoring function to serve as an empirical measure for differentiating between groups of bundles that are far from the native structure from those that are near. It is still unclear to us how low the Bundler score of a structure must be in order for that structure to be of use in our process of protein structure determination. We believe that the threshold of useful scores will vary from protein to protein and thus must be determined empirically.

4. Optimization Methods

Because the Bundler scoring function is nonsmooth and contains discontinuities, we have chosen to apply derivative-free methods to obtain a solution to (1). Although we focus on two particular methods here, there are many other derivative-free methods; see for example, Powell (1998), Kolda et al. (2003)

and references therein. Moreover, finite differencing could be used to approximate the gradient so that we could use derivative-based methods. However, because Bundler is discontinuous and directly incorporates noisy experimental distance constraints, such approximations may contain too much error to be useful (Hough and Meza 2002). In this paper, we present results using simulated annealing and parallel pattern search, described below.

4.1. Simulated Annealing

Simulated annealing (SA) is arguably the most widely used optimization method for molecular conformation problems. For just a few of the many examples of the use of simulated annealing in computational biology, see Ghosh et al. (2002), Perkins and Dean (1993), Campbell et al. (1998), Goodsell and Olson (1990), Brünger et al. (1997). The SA algorithm is a computational analogue to the industrial annealing process in which metal alloys are slowly cooled to obtain an optimal molecular configuration. This controlled cooling process is very important because a less stable configuration is obtained when the alloy is cooled too quickly. Computationally, annealing is implemented by allowing optimization steps that do not necessarily reduce the objective function. The idea is that a few bad steps can be accepted in order to get on the best path to the solution.

The SA algorithm is based on the Metropolis method (Metropolis et al. 1958) of obtaining the equilibrium configuration of a group of atoms at a given temperature. A connection between the Metropolis method and Monte Carlo simulation was first described in Pincus (1970). The simulated annealing optimization technique that is used today was proposed in Kirkpatrick et al. (1983). It begins with a Metropolis Monte Carlo simulation at a high temperature. After a sufficient number of Monte Carlo steps have been taken, the temperature is reduced and the Metropolis Monte Carlo is continued. This process is repeated until a specified final temperature is reached. At high temperatures, a relatively large number of random steps will be accepted, and, as the temperature decreases, fewer steps are accepted.

The main advantage of SA over other optimization methods is that it is global. In theory, the algorithm can avoid becoming trapped in bad local minima regardless of its starting point. Furthermore, SA is easy to implement. Unfortunately, SA also has many well-documented disadvantages. It requires extensive computational work (van Laarhoven and Aarts 1987, Moret and Shapiro 1991, Aarts et al. 1997, Elmohamed et al. 1998), and it is sensitive to the choices of its many parameters (Elmohamed et al. 1998, Piccioni 1987, Stiles 1994, Aarts et al. 1997, Randelman and Grest 1986, van Laarhoven and Aarts 1987). For example, there are at least a dozen different temperature cooling schedules from which to choose (Kirkpatrick et al. 1983, Geman and Geman 1984, van Laarhoven and Aarts 1987, Salamon et al. 2002). Finally, because the steps in SA are taken randomly, the algorithm does not employ any knowledge gained in previous iterations (Ali and Storey 1997).

Because SA is the method of choice in the computational biology community and because it is also easy to implement, it was the first optimizer that we tried. In our implementation of SA, we use the geometric annealing schedule,

$$T_{\text{new}} = \alpha T_{\text{old}}, \tag{11}$$

where $\alpha = 0.95$. We selected this schedule on the basis of numerical experiments, and our selection is supported by Johnson et al. (1989, 1991). The initial temperature and number of temperature cycles were chosen independently for each of the numerical tests presented in this paper. Each temperature cycle is terminated after either 1,000 structures are generated or 100 structures are accepted. New structures are generated as follows: First, one of the helices is randomly selected. Then, starting from the arrangement of the last accepted structure, the position of the selected helix is randomized either by translating it or by rotating it around the x and y or the z axis. The type and amount of randomization are randomly chosen within a user defined maximum.

Our SA algorithm is implemented in C and uses the PDB Record I/O Libraries to read and write Brookhaven PDB formatted files (Couch et al. 1995). Our implementation of SA is serial. Although some parallelized versions exist (Kliewer and Tschöke 1998, Stiles et al. 1989, Lee 1995), none are compatible with MPI libraries such as MPICH-1.2.4 (Gropp et al. 1996, Gropp and Lusk 1996). We chose to use a basic implementation of SA but note that there are many sophisticated variations. For example, reannealing has been shown to be effective by adapting to changes in parameter sensitivities when the search becomes trapped (Ingber 1989, 1993). Other adaptive and multistart modifications of SA have also been shown to be successful (Piccioni 1987, Stiles 1994, Ali and Storey 1997, Salamon et al. 2002).

4.2. Asynchronous Parallel Pattern Search

To contrast SA, we opted to apply an algorithm from a completely different class of optimization methods—pattern searches. Because this class of methods is generally overlooked in computational biology, we were particularly interested in examining its applicability and performance.

Pattern search methods are practical for solving problems such as (1) when the derivative of the objective function is unavailable and approximations are unreliable. They use a predetermined pattern of

points to sample the given function domain. When certain requirements on the form of the points in the pattern are followed, it can be shown that if the objective function is smooth, global convergence to a stationary point is guaranteed (Dolan et al. 2000, Lewis and Torczon 1996, Torczon 1997). Bundler, our objective function, is not smooth, but further analysis reveals that pattern search may still find a minimum even for nonsmooth functions (Kolda et al. 2003). We also note that pattern search methods are typically used for optimization problems with fewer than 100 variables (Kolda et al. 2003). Most transmembrane proteins have fewer than 13 helices, and we are interested in proteins that have 12 or fewer. Hence, the transmembrane protein structure determination problem that we consider contains at most 72 variables, and pattern search is a reasonable choice.

The majority of the computational cost of pattern search methods is in the function evaluations, so parallel pattern search (PPS) techniques have been derived to reduce the overall computational time. Specifically, PPS exploits the fact that, once the points in the search pattern have been defined, the function values at these points can be computed simultaneously (Dennis and Torczon 1991, Torczon 1992). The particular implementation of PPS that we use is asynchronous. Asynchronous parallel pattern search (APPS) (Kolda and Gray 2004) retains the positive features of PPS, but it does not assume that the amount of time required for an objective function evaluation is constant or that the processors are homogeneous. It does not have any required synchronizations and thus requires less total time than PPS to return results (Hough et al. 2001). Furthermore, it has been shown that APPS is globally convergent under the standard assumptions for PPS (Kolda and Torczon 2004). Finally, there is an existing open source version of APPS, called APPSPACK, which is easy to install and use (Kolda and Gray 2004).

APPSPACK is available in MPI, PVM, and serial modes. To make use of the parallel machines at our disposal, we opted to use the MPI mode of APPSPACK version 3.0. This mode and version requires a minimum of three processors: one master agent to coordinate the search, one cache agent to save and look up points at which the function has already been evaluated, and at least one worker to perform function evaluations. For our problem, the use of cache is particularly advantageous as it should reduce the required number of new function evaluations and increase algorithm efficiency. The default MPI version of APPSPACK requires that the function evaluations be run as separate executables and communicates with the worker tasks via file input and output. In our case, we customized APPSPACK to avoid the overhead of system calls and file I/O and improve overall efficiency.

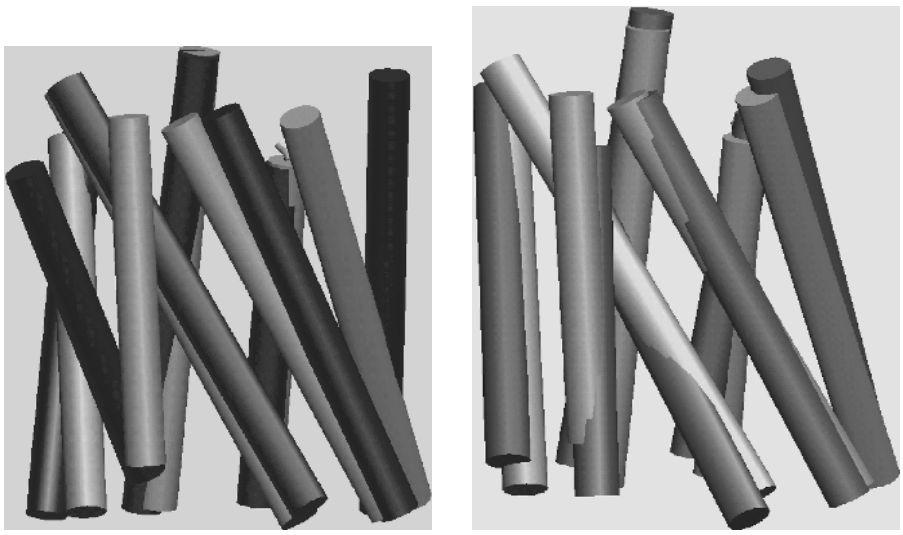
We found that APPS required almost no tuning. We used the default values for all the parameters except the convergence tolerance, which was set to be 0.01. The default tolerance of 10^{-4} is small with respect to the variable sensitivity in our application, and thus we increased it in order to reduce the number of function evaluations required to obtain convergence. We also note that our implementation uses the coordinate direction search pattern.

5. Numerical Results

In this section, we present numerical results obtained using experimental distance constraints for rhodopsin. Rhodopsin is a transmembrane protein that is located in the retinal rods of the eye, and it plays a role in vision. It is a G-protein-coupled receptor made up of 7 transmembrane helices and thus has 42 variables in its structure determination problem. The 3-D structure of the dark-adapted form of rhodopsin is known, having been determined using x-ray crystallography (Palczewski et al. 2000). Moreover, a set of experimental distance constraints for dark-adapted rhodopsin has been compiled in Yeagle et al. (2001) making it an appropriate test case for our numerical experiments. In this paper, we use the structure of rhodopsin determined in Palczewski et al. (2000), PDB entry 1F88, as the true structure. Because we are using a known structure, we can compute the difference between the true structure and any other structure using RMSD computed across the Cα atoms. Although we cannot use RMSD when trying to ascertain structures that have not yet been determined, we use it in our study to add clarity to the comparisons.

5.1. Motivation

We begin by presenting a simple example that motivated this study. Here, we use only one starting point, which was obtained by randomizing the true structure of rhodopsin. The subsequent starting structure has an initial Bundler score of 11,342 and an RMSD of 15.0. We first tried optimizing this structure using our SA algorithm. After extensive tuning, the best structure we were able to produce resulted from using a starting temperature of 500 and 290 temperature cycles. This structure has a final Bundler score of 377 and an RMSD of 4.5. Next, we applied the APPS algorithm. On our first try, we were able to produce a structure with a score of 122 and an RMSD 3.4. Figure 3 contains two pictures that illustrate the spatial positions of the helices relative to the known structure. In both pictures, the light gray cylinders represent the α helices of dark-adapted rhodopsin. In picture (a), on the left, the dark cylinders depict the



(a) Simulated Annealing (b) APPS

Figure 3 Comparison of the True and Calculated Locations of the Helices of Dark-Adapted Rhodopsin

locations of the helices found using simulated annealing. Picture (b), on the right, shows the helices' locations determined by APPS as the dark cylinders. Note that APPS determines the orientation of all seven helices relatively well. In contrast, two of the helices determined by SA are a poor match.

We also examined the computational efficiency of each method. As previously discussed, SA often requires extensive computational work. This example was no exception. The SA algorithm required 81,800 function evaluations and 61 hours of run time on a single processor. In comparison, the APPS algorithm required only 32,458 function evaluations and 17 minutes of run time on 86 processors. It is difficult to compare the two algorithms directly because SA is serial and APPS is parallel. However, we can note that APPS required fewer total function evaluations than SA. Moreover, if SA were parallelized in the most efficient manner possible and run on 86 parallel processors, it would still take almost 45 minutes to obtain a solution.

This result led us to pursue a more thorough evaluation. We now present this study and its results.

5.2. Numerical Study

For our numerical study, 87 starting structures were selected from 7.0×10^{13} possible candidates using the procedure described in detail in Faulon et al. (2003) and a set of 27 distance constraints, $\mathcal{D}_1$, obtained from Yeagle et al. (2001). This procedure resulted in structures that have no experimental distance penalty, i.e., $P_E = 0$, where P_E is as defined in (3), for each of the 87 structures with respect to $\mathcal{D}_1$. To fully test the capabilities of the optimization methods, we use a different set of distance constraints, $\mathcal{D}_2$. The set $\mathcal{D}_2$ contains upper and lower bounds for the same 27 pairs of atoms as $\mathcal{D}_1$, but the range of these bounds is tighter as detailed in Yeagle et al. (2001), Sale et al. (2004). The average Bundler score of the starting structures is 26,555 with a maximum of 76,080 and a minimum of 8,608. The distribution of these scores is shown in Figure 4.

To optimize the 87 structures, we first applied our SA algorithm with a starting temperature of 300 and 125 temperature cycles. Next, we applied APPS to the same set of 87 starting structures. The results of this test set are displayed in Figure 5 and summarized in Figure 6. Note that APPS produces a much wider range of final scores than SA, and it appears that with

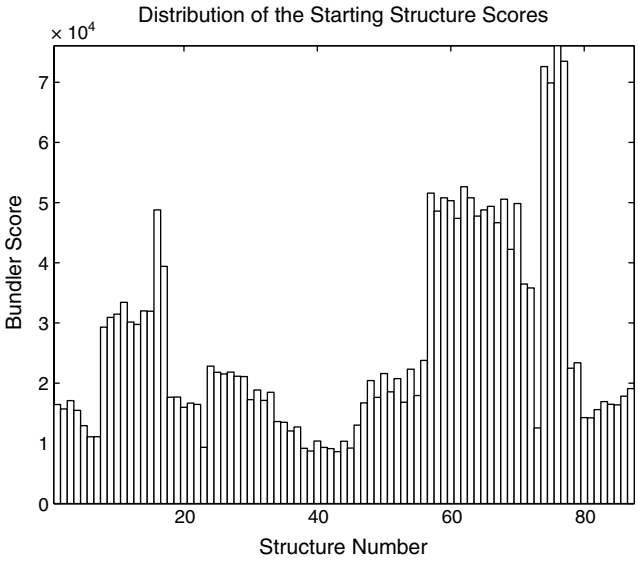


Figure 4 Distribution of the Initial Bundler Scores

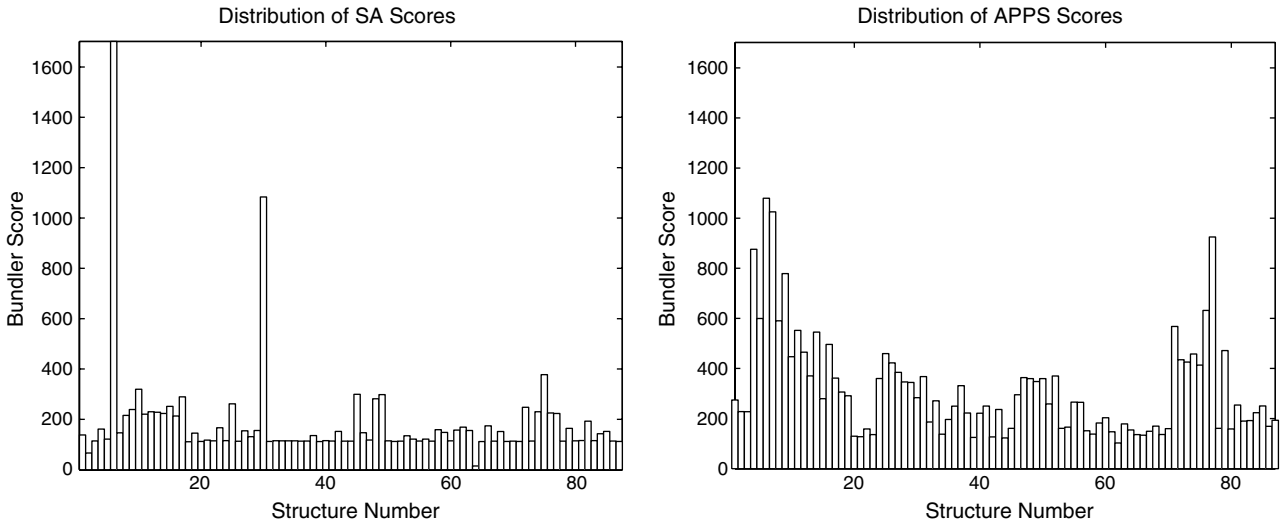


Figure 5 Final Bundler Scoring Distribution for SA (Left) and APPS (Right)

APPS, some of starting structures get stuck in bad local minima. In contrast, the majority of the scores achieved by SA are below 200 and in fact, 40 of the 87 structures differ by less than six Bundler points.

We can conclude that overall, this implementation of SA more *effectively* reduces the Bundler score. However, recall the aim of this particular project: to produce at least one structure with an empirically low score as efficiently as possible. In considering this goal, the fact that SA produced more structures with low scores does not necessarily give it an edge over APPS. The APPS algorithm does yield some structures with low scores. Further study showed that for each SA structure with a score of less than 200, there is at least one APPS structure with a score of less than 200 such that its RMSD with respect to the SA structure is less than three. Given the errors in the distance constraints, we can therefore conclude that SA and APPS are finding the same minima. Furthermore, as Figure 6 shows, the computational cost of the success of SA is quite high. Each structure requires a minimum of 49,500 function evaluations to produce a solution. In comparison, the maximum number of function evaluations needed by APPS is 48,812, and 24 of the runs required fewer than 20,000 evaluations. Therefore, we conclude that APPS more *efficiently* reduces the Bundler scoring function. In fact, APPS is better suited than this implementation of SA for our transmembrane protein structure determination problem.

Next, we decided to more closely examine our implementation of SA to see if there was a simple way of improving the efficiency of SA without sacrificing too much of its effectiveness. One way of reducing the number of SA function evaluations is reducing the number of temperature cycles. We use

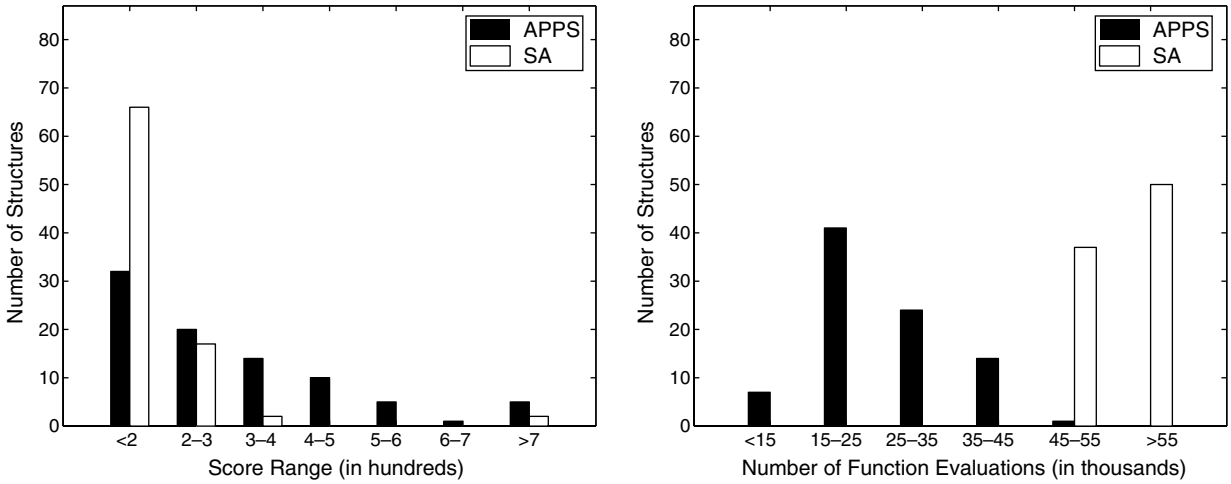


Figure 6 Summary of the Final APPS and SA Bundler Values (Left) and Required Number of Function Evaluations (Right)

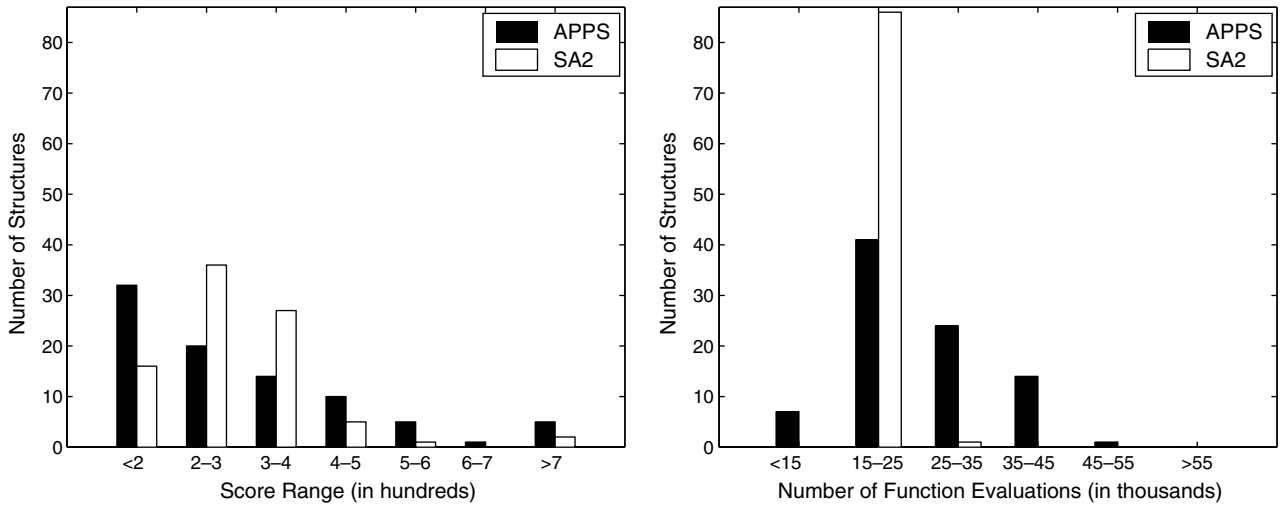


Figure 7 Summary of the Final APPS and SA2 Bundler Values (Left) and Required Number of Function Evaluations (Right)

SA2 to denote the SA algorithm terminated after only 60 temperature cycles, or approximately one third of the number of function evaluations of the previous implementation. The results of this comparison are shown in Figure 7. The APPS and SA2 algorithms obtain solutions to problem (1) for the 87 different starting points in a similar number of function evaluations. An RMSD comparison of the structures with scores less than 200 showed that SA2 and APPS find the same minima. Hence, for our problem, both SA2 and APPS achieve our goals. The SA2 algorithm is less effective than SA at reducing the scoring function, but it still produces structures with a low Bundler score, and it is more efficient than SA. The explicit distribution of the final SA2 Bundler scores is shown in Figure 8. The average final Bundler score is 306 with a maximum of 1,883 and a minimum of 132. The results of this test allowed us to conclude that APPS is indeed a practical choice for our problem and that it is competitive with simulated annealing, the method of choice in the computational biology community.

For our final test, we tried to reduce the number of SA temperature cycles by using a lower starting temperature. Here, we use an initial temperature of 30 and do 75 temperature cycles. By beginning with a lower temperature, we will not accept as many randomized steps and thus we are, in effect, doing a more localized search. We use SA3 to denote this algorithm and summarize its results in Figure 9. Although SA3 still requires fewer function evaluations than SA, it does not successfully produce structures with low Bundler scores. None of the SA3 final scores are below 275, and, in fact, only two structures have scores below 300. The final Bundler score distribution for SA3 is given in Figure 10, and the average final Bundler score is 561 with a maximum of 2,386 and a minimum of 274. In addition, SA3 is overall less efficient than both the SA2 and the APPS algorithms. Therefore, we can conclude that the simulated annealing algorithm using these particular parameters, a low initial temperature and a small number of temperature cycles, is not a viable alternative for solving our problem.

6. Conclusions

In this paper, we discuss the transmembrane protein structure identification problem. In particular, we focus on the second step of an innovative two-step method that combines laboratory and computational techniques (Faulon et al. 2003, Sale et al. 2004) to determine the spatial locations of the transmembrane helices. This second step refines a large set of possible helical bundles, generated in step one, by optimizing an empirical scoring function known as Bundler.

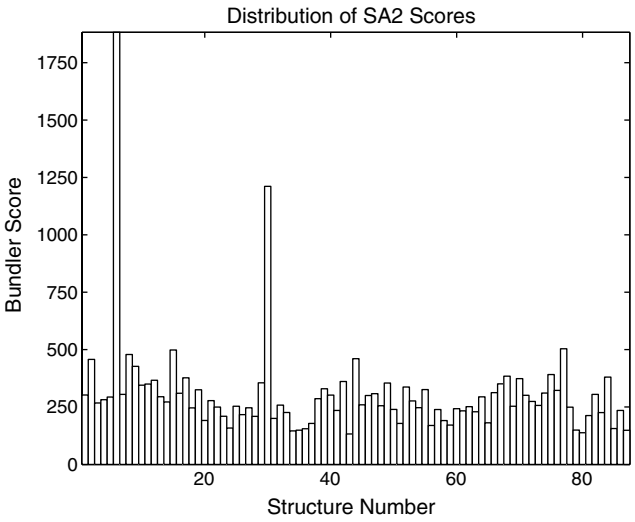


Figure 8 Distribution of the Final SA2 Bundler Scores

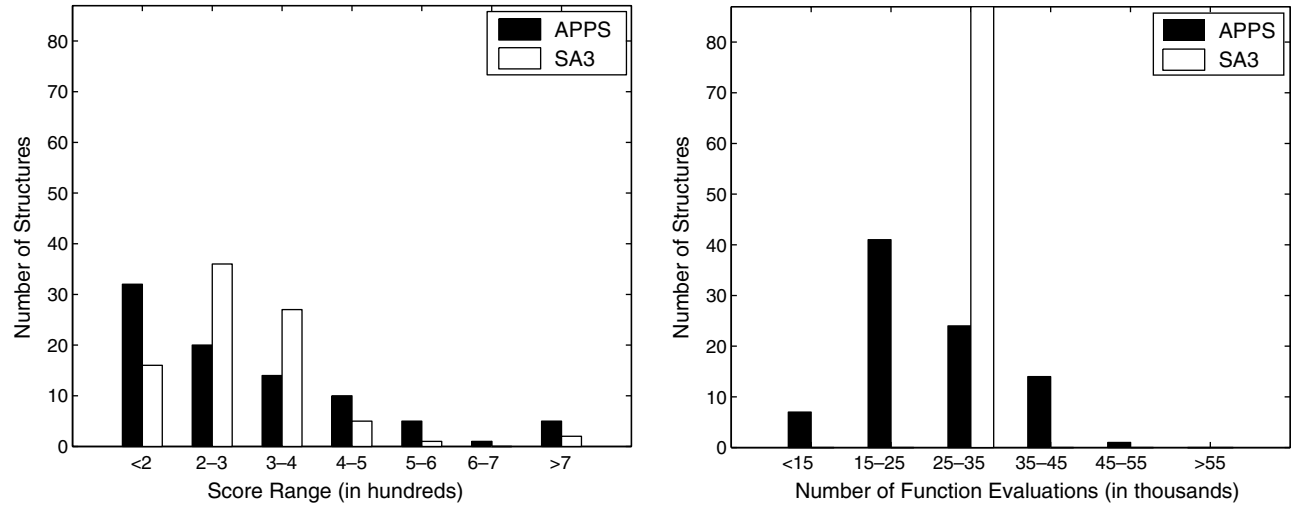


Figure 9 **Summary of the Final APPS and SA3 Bundler Values (Left) and Required Number of Function Evaluations (Right)**

The optimization of Bundler raises the question of finding an appropriate minimization algorithm.

Because Bundler is a discontinuous function that incorporates noisy experimental data, we opted to use a derivative-free method. We considered two very different algorithms: SA and APPS. The SA algorithm imitates an industrial cooling process and uses Metropolis Monte Carlo to generate new points. In contrast, APPS is a pattern search method that uses a predetermined pattern of points to sample a given function domain. In testing these methods, we had to consider the goal of our project: identifying at least one structure with a Bundler score low enough to warrant further study. Because Bundler was designed using only a small set of transmembrane proteins and inexact laboratory data, a high level of accuracy is neither expected nor desired from the optimization.

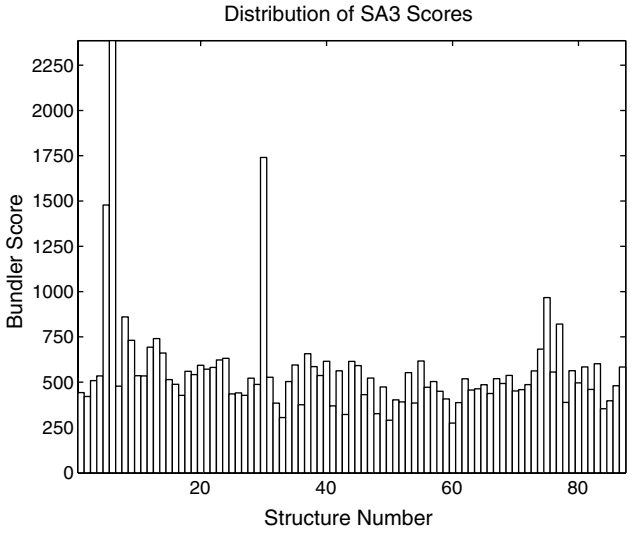


Figure 10 **Distribution of the Final SA3 Bundler Scores**

However, efficiency is important as hundreds or even thousands of structures must be optimized. Therefore, we were looking for an optimization method that is both efficient and sufficiently accurate.

Given the numerous variations of the SA algorithm and the number of documented successes using SA, we are confident that we could eventually find a suitable version of the SA algorithm to solve our transmembrane protein structure determination problem. In fact, we have demonstrated that the SA2 implementation is sufficient for identifying the helical placement of rhodopsin. However, it is unclear whether or not this algorithm would be sufficient for a general transmembrane protein or if its parameters would be biased for certain proteins. We must also consider the fact that the Bundler scoring function will likely undergo a series of minor improvements, and we do not want any of these small changes to require that the optimization algorithm be retuned.

To our knowledge, APPS has not been previously applied to a problem of protein structure determination. We did not encounter any difficulties in applying it to our transmembrane protein problem. In fact, on our first try, we were able to produce desirable results. Moreover, in light of our efficiency and accuracy specifications, APPS was superior to both our SA and the SA3 algorithms and was comparable to our SA2 algorithm.

Despite their similar performance, APPS still has two implicit advantages over our SA2. First, APPS is easy to fine tune. Note that SA2 is exactly the same as our original SA algorithm with the exception of one small change to one parameter. However, the results obtained from these two algorithms are significantly different. Choosing the total number of temperature cycles is difficult. We must complete enough cycles to sufficiently reduce the Bundler score but not so many

as to ignore our efficiency requirements. This is of concern to us because we will be using different sets of distance constraints and making minor changes to the Bundler scoring function. Second, because it is parallel, APPS will require less wall-clock time for problems with a large number of starting structures. Therefore, we have chosen to use APPS as the optimizer for this problem.

Acknowledgments

The authors gratefully acknowledge the Mathematical, Information, and Computational Sciences Program of the United States Department of Energy and the Laboratory Directed Research and Development program at Sandia National Laboratories for their support of this research. Sandia National Laboratories is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000.

References

Aarts, E. H., J. Korst, P. J. van Laarhoven, 1997. *Local Search in Combinatorial Optimization*, Chapter 4. *Simulated Annealing*. John Wiley and Sons, New York.

Adamian, L., R. Jackups, Jr., T. A. Binkowski, J. Liang. 2003. Higher-order interhelical spatial interactions in membrane proteins. *J. Molecular Biol.* **327** 251–272.

Adiman, L., J. Liang. 2001. Helix-helix packing and interfacial pairwise interactions of residues in membrane proteins. *J. Molecular Biol.* **311** 891–907.

Ali, M. M., C. Storey. 1997. Aspiration based simulated annealing algorithm. *J. Global Optim.* **11** 181–191.

Banaszak, L. 2000. *Foundations of Structural Biology*. Academic Press, San Diego, CA.

Berliner, L. J., S. S. Eaton, G. R. Eaton. 2001. *Distance Measurements in Biological Systems by EPR*, Vol. 19. *Biological Magnetic Resonance*. Kluwer Academic Publishers/Plenum Publishing Corp., New York.

Berman, H. M., J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I. N. Shindyalov, P. E. Bourne. 2000. The Protein Data Bank. *Nucleic Acids Res.* **28** 235–242. http://www.rcsb.org/pdb.

Bowie, J. U. 1997. Helix packing in membrane proteins. *J. Molecular Biol.* **272** 780–789.

Bowie, J. U. 1999. Helix-bundle membrane protein fold templates. *Protein Sci.* **8** 2711–2719.

Brandon, C., J. Tooze. 1999. *Introduction to Protein Structure*, 2nd ed. Garland Publishing Inc., New York.

Brünger, A. T. 1992. *X-PLOR: A System for X-ray Crystallography and NMR*, Version 3.1. Department of Molecular Biophysics and Biochemistry, Yale University, New Haven, CT. http://www.ocms.ox.ac.uk/mirrored/xplor/manual/htmlman/htmlman.html.

Brünger, A. T., P. D. Adams, L. M. Rice. 1997. New applications of simulated annealing in X-ray crystallography and solution NMR. *Structure* **5** 325–336.

Buchan, D., D. Shepherd, D. Lee, F. Pearl, S. Rison, J. Thorton, C. Orengo. 2002. Gene3D: Structural assignment for whole genes and genomes using the CATH domain structure database. *Genome Res.* **12** 503–514.

Campbell, B. J., G. Bellussi, L. Carluccio, G. Perego, A. K. Cheetham, D. E. Cox, R. Millin. 1998. The synthesis of the new zeolite, ERS-7, and the determination of its structure by simulated annealing and synchrotron X-ray powder diffraction. *J. Chemical Soc., Chemical Comm.* **16** 1725–1726.

Couch, G. S., E. F. Pettersen, C. C. Huang, T. E. Ferrin. 1995. Annotating PDB files with scene information. *J. Molecular Graphics* **13** 153–158.

Creighton, T. E. 1992. *Proteins: Structures and Molecular Properties*, 2nd ed. W. H. Freeman & Co., New York.

Dennis, Jr., J. E., V. Torczon. 1991. Direct search methods on parallel machines. *SIAM J. Optim.* **1** 448–474.

Dobbs, H., E. Orlandini, R. Bonaccini, F. Seno. 2002. Optimal potentials for predicting inter-helical packing in transmembrane proteins. *Proteins* **49** 342–349.

Dolan, E. D., R. M. Lewis, V. Torczon. 2000. On the local convergence properties of parallel pattern search. Technical report 2000-36, NASA Langley Research Center, Institute for Computer Applications in Science and Engineering, Hampton, VA.

Elmohamed, S., G. Fox, P. Coddington. 1998. A comparison of annealing techniques for academic course scheduling. *2nd Internat. Conf. Practice Theory Automated Timetabling*. Syracuse, NY, 146–166.

Faulon, J.-L., M. D. Rintoul, M. M. Young. 2002. Constrained walks and self-avoiding walks: Implications for protein structure determination. *J. Phys. A.: Math. Gen.* **35** 1–19.

Faulon, J.-L., K. Sale, M. M. Young. 2003. Exploring the conformational space of membrane protein folds matching distance constraints. *Protein Sci.* **12** 1750–1761.

Geman, S., D. Geman. 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intelligence* **6** 721–741.

Ghosh, A., R. Elber, H. A. Scheraga. 2002. An atomically detailed study of the folding pathways of protein A with the stochastic difference equation. *Proc. National Acad. Sci. USA* **99** 10394–10398.

Goodsell, D. S., A. J. Olson. 1990. Automated docking of substrates to proteins by simulated annealing. *Proteins: Structure Function and Genetics* **8** 195–202.

Gropp, W. D., E. Lusk. 1996. User's guide for mpich, a portable implementation of MPI. Technical report ANL-96/6, Mathematics and Computer Science Division, Argonne National Laboratory.

Gropp, W., E. Lusk, N. Doss, A. Skjellum. 1996. A highperformance, portable implementation of the MPI message passing interface standard. *Parallel Comput.* **22** 789–828.

Herzyk, P., R. E. Hubbard. 1998. Using experimental information to produce a model of the transmembrane domain of the ion channel phospholamban. *Biophysical J.* **74** 1203–1214.

Hough, P. D., J. C. Meza. 2002. A class of trust-region methods for parallel optimization. *SIAM J. Optim.* **13** 264–282.

Hough, P. D., T. G. Kolda, V. Torczon. 2001. Asynchrounous parallel pattern search for nonlinear optimization. *SIAM J. Sci. Comput.* **23** 134–156.

Hulsen, T., D. Lutje-Hulsik. 2001. GPCR Website: Pictures about GPCRs (G protein-coupled receptors). University of Nijmegen, The Netherlands, http://www.cmbi.kun.nl/~dlutjehu/pictures.html.

Ingber, L. 1989. Very fast simulated re-annealing. *Math. Comput. Model.* **12** 967–973.

Ingber, L. 1993. Simulated annealing: Practice versus theory. *Math. Comput. Model.* **18** 29–57.

Jayasinghe, S., K. Hristova, S. H. White. 2001a. Energetics, stability, and prediction of transmembrane helices. *J. Molecular Biol.* **312** 927–934.

Jayasinghe, S., K. Hristova, S. H. White. 2001b. MPtopo: A database of membrane protein topology. *Protein Sci.* **10** 455–458.

Johnson, D. S., C. R. Aragon, L. A. M. McGeoch, C. Schevon. 1989. Optimization by simulated annealing: An experimental evaluation, part I, graph partitioning. *Oper. Res.* **37** 865–892.

Johnson, D. S., C. R. Aragon, L. A. M. McGeoch, C. Schevon. 1991. Optimization by simulated annealing: An experimental evaluation; part II, graph coloring and number partitioning. *Oper. Res.* **39** 378–406.

Kirkpatrick, S., C. D. Gerlatt, Jr., M. P. Vecchi. 1983. Optimization by simulated annealing. *Science* **220** 671–680.

Kliewer, G., S. Tschöke. 1998. A general parallel simulated annealing library (parSA) and its applications in industry. PAREO 1998: First meeting of the PAREO working group on Parallel Processing in Operations Research, Versailles, France.

Kolda, T. G., G. A. Gray. 2004. APPSPACK 4.0: Asynchronous parallel pattern search for derivative-free optimization. ACM Trans. Math. Software. Forthcoming. Software and documentation available at http://software.sandia.gov/appspack/.

Kolda, T. G., V. Torczon. 2004. On the convergence of asynchronous parallel pattern search. *SIAM J. Optim.* **14** 939–964.

Kolda, T. G., R. M. Lewis, V. Torczon. 2003. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM Rev.* **45** 385–482.

Krishna, N. R., L. J. Berliner. 1999. *Structural Computation and Dynamics in Protein*, Vol. 17. *Biological Magnetic Resonance.* Kluwer Academic Publishers/Plenum Publishing Corp., New York.

Krough, A., B. Larsson, G. von Heijne, E. Sonnhammer. 2001. Predicting transmembrane protein topology with a hidden Markov model: Application to complete genomes. *J. Molecular Biol.* **305** 567–580.

Leach, A. R. 2001. *Molecular Modelling. Principles Applications*, 2nd ed. Prentice Hall, New York.

Lee, F. H. A. 1995. *Parallel simulated annealing on a message-passing multicomputer.* Ph.D. thesis, Utah State University, Department of Electrical Engineering, Logan, UT.

Lewis, R. M., V. Torczon. 1996. Rank ordering and positive basis in pattern search algorithms. Technical report 96-71, NASA Langley Research Center, Institute for Computer Applications in Science and Engineering, Hampton, VA.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, E. Teller. 1958. Equations of state calculations by fast computing machines. *J. Chemical Phys.* **21** 1087–1092.

Moret, B. M. E., H. D. Shapiro. 1991. *Algorithms from P to NP*, Vol. I. Benjamin/Cummings Publishing Company, Redwood City, CA.

Nikiforovich, G. V., S. Galaktionov, J. Balodis, G. R. Marshall. 2001. Novel approach to computer modeling of seven-helical transmembrane proteins: Current progress in the test case of bacteriorhodopsin. *Acta Biochimica Polonica* **48** 53–64.

Palczewski, K., T. Kumasaka, T. Hori, C. A. Behnke, H. Motoshima, B. A. Fox, I. L. Trong, D. C. Teller, T. Okada, R. E. Stenkamp, M. Yamamoto, M. Miyano. 2000. Crystal structure of rhodopsin: A G protein-coupled receptor. *Science* **289** 739–745.

Perkins, T. D. J., P. M. Dean. 1993. An exploration of a novel strategy for superposing several flexible molecules. *J. Comp. Aided Molecular Design* **7** 155–172.

Piccioni, M. 1987. A combined multistart-annealing algorithm for continuous global optimization. Technical report 87-45, The University of Maryland, Systems and Research Center, College Park, MD.

Pincus, M. 1970. Monte Carlo method for the approximate solution of certain types of constrained optimization problems. *Oper. Res.* **18** 1225–1228.

Powell, M. J. D. 1998. Direct search algorithms for optimization calculations. *Acta Numer.* **7** 287–336.

Randelman, R. E., G. S. Grest. 1986. *N*-city traveling salesman problem—optimization by simulated annealings. *J. Statist. Phys.* **45** 885–890.

Richards, F. M. 1974. The interpretation of protein structures: Total volume, group volume, distributions and packing density. *J. Molecular Biol.* **82** 1–14.

Rose, G. D. 1978. Prediction of chain turns in globular proteins on a hydrophobic basis. *Nature* **272** 586–590.

Salamon, P., P. Sibani, R. Frost. 2002. *Facts, Conjectures, and Improvements for Simulated Annealing*. Monographs on Mathematical Modeling and Computation 7. SIAM, Philadelphia, PA.

Sale, K. L., G. A. Gray, J.-L. Faulon, J. Schoeniger, M. M. Young. 2004. Optimal bundling of transmembrane helices of integral membrane proteins using sparse distance constraints. Technical report 2004-2647J, Sandia National Laboratories. *Protein Sci.* Forthcoming.

Stiles, G. S. 1994. The effect of numerical precision upon simulated annealing. *Phys. Lett. A* **185** 253–261.

Stiles, G. S., K. W. Bosworth, T. W. Morgan, F. H. Lee, R. J. Pennington. 1989. Parallel optimization of distributed database networks. *Proc. First Internat. Conf. Appl. Transputers*. IOS Press, Amsterdam, The Netherlands.

Torczon, V. 1992. PDS: Direct search methods for unconstrained optimization on either sequential or parallel machines. Technical report TR92-09, Rice University, Department of Computational and Applied Math, Houston, TX.

Torczon, V. 1997. On the convergence of pattern search algorithms. *SIAM J. Optimization* **7** 1–25.

Vaidehi, N., W. B. Floriano, R. Trabanino, S. E. Hall, P. Freddolino, E. J. Choi, G. Zamanakos, W. A. Goddard, III. 2002. Prediction of structure and function of G protein-couple receptor. *Proc. National Acad. Sci.* **99** 12622–12627.

van Laarhoven, P. M. J., E. H. L. Aarts. 1987. *Simulated Annealing: Theory and Applications*. Dordrecht Reidel Publishing Company, Dordrecht, The Netherlands. Republished in 1989 by Kluwer Academic, Boston, MA.

Vriend, G. 1990. WHAT IF: A molecular modeling and drug design program. *J. Molecular Graphics* **8** 52–56.

Wallin, E., G. von Heijne. 1998. Genome-wide analysis of integral membrane proteins from eubacterial, archaen, and eukaryotic organisms. *Protein Sci.* **283** 489–506.

White, S. 1998. Membrane proteins of known 3D structure. Stephen White Laboratory, University of California, Irvine. http://blanco.biomol.uci.edu/Membrane_Proteins_xtal.html.

Wilson, S., D. Bergsma. 2000. Orphan G-protein coupled receptors: Novel drug targets for the pharmaceutical industry. *Drug Design Discovery* **17** 105–114.

Xiang, Z., B. Honig. 2001. Extending the accuracy limits of prediction for side chain conformations. *J. Molecular Biol.* **311** 421–430.

Yeagle, P. L., G. Choi, A. D. Albert. 2001. Studies on the structure of the G-protein-coupled receptor rhodopsin including the putative G-protein binding site in unactivated and activated forms. *Biochemistry* **40** 11932–11937.