Heuristic Refinement Method for the Derivation of Protein Solution Structures: Validation on Cytochrome *b*562†

JAMES F. BRINKLEY,‡,|| RUSS B. ALTMAN,‡ BRUCE S. DUNCAN,§ BRUCE G. BUCHANAN,‡ and OLEG JARDETZKY*,§

Knowledge Systems Laboratory and Stanford Magnetic Resonance Laboratory, Stanford University, Stanford, California 94305

Received December 16, 1987

A method is described for determining the family of protein structures compatible with solution data obtained primarily from nuclear magnetic resonance (NMR) spectroscopy. Starting with all possible conformations, the method systematically excludes conformations until the remaining structures are only those compatible with the data. The apparent computational intractability of this approach is reduced by assembling the protein in pieces, by considering the protein at several levels of abstraction, by utilizing constraint satisfaction methods to consider only a few atoms at a time, and by utilizing artificial intelligence methods of heuristic control to decide which actions will exclude the most conformations. Example results are presented for simulated NMR data from the known crystal structure of cytochrome *b*562 (103 residues). For 10 sample backbones an average root-mean-square deviation from the crystal of 4.1 Å was found for all α-carbon atoms and 2.8 Å for helix α-carbons alone. The 10 backbones define the family of all structures compatible with the data and provide nearly correct starting structures for adjustment by any of the current structure determination methods.

The study of protein structure in solution differs from the study of protein structure in crystalline form in a number of significant ways. Protein molecules in a crystal are all in roughly the same conformation, so a single, rigid structure can be sought to explain the data.[1] The atoms within a crystal structure usually have a very sharp distribution around a mean position, and a temperature factor summarizes the uncertainty in this position.

In the interpretation of solution data, however, there may be much larger uncertainty in the position of some or all atoms. High-resolution solution data from nuclear magnetic resonance (NMR) may not adequately constrain all atoms.[2,3] In addition, the data are collected over a population of molecules that may or may not be in the same conformation: the data are an average over time as well as over many molecules. The attempt to converge on a single, rigid conformer could therefore be a misleading interpretation of the data.

A more accurate description for the solution structure of a molecule would be a $4N$-dimensional *probability density function* for the possible conformations of the protein, where N is the number of atoms, each of which is characterized by three x,y,z coordinates plus a fourth component of time. The value of the density function at each $4N$-dimensional point gives the probability that the protein is in a particular conformation at a particular time. The density function is of high dimension because the position of each atom is highly dependent on the positions of other atoms in the protein.

However, in solution studies the data are a time average of many molecules that may be moving or may simply be in different conformations. In this case it is generally impossible to determine from the data alone the behavior of the protein over time. Instead, the best that can be obtained from the data is an approximation to the $3N$-dimensional density function describing the probabilities that the time-averaged population of proteins are in various conformations. This function essentially gives the family of conformations that are compatible with the data.

In dealing with solution structures it is thus necessary to ascertain that the family obtained from the data is truly representative of the entire range of possibilities, i.e., that it is accurate. This can in principle be achieved only by a systematic sampling of the entire conformational space. Such systematic sampling, at atomic resolution, is possible for only relatively small structures and with larger peptides and proteins rapidly becomes computationally intractable. It then becomes necessary to choose between random sampling followed by optimization and the explicit introduction of simplifying assumptions to reduce the computational complexity of systematic sampling.

There are thus generally two paradigms for structure determination. Methods that are based on an *adjustment paradigm* are characterized by the iterative adjustment of a single randomly generated structure to the minimum of some evaluation function. Methods within the *exclusion paradigm*, such as ours, rely on systematic sampling of conformational space, with exclusion of those structures that are not compatible with a given set of constraints. The exclusion paradigm has appeal because it has the potential to define the limits of the allowed conformational space, which is necessary for any proof of validity or accuracy of a structure determination method. However, since the main objective of the exclusion paradigm is to define the scope and limits of conformational space implied by the data, the simplifying assumptions introduced to make the computation tractable must be explicit and testable.

Among the adjustment methods are distance geometry algorithms that use a global distance error function to evaluate the structure.[4-6] These algorithms utilize the gradient of the error function to iteratively adjust the structure until the errors are small. Restrained molecular dynamics algorithms model the free energy of the protein and seek the global minima of the energy function.[7-9]

There is evidence that structures determined within the adjustment paradigm may not always explain the original experimental data in the sense that they do not satisfy the NMR Bloch equations.[10,11] In addition, these methods are generally not capable of producing a representative sample of the $3N$-dimensional density function of all structures com-

† This project was supported in part by NSF Grant DMB8402348, NIH Grants RR02300, GM07365, and RR00785-14, DARPA Grant N0039-86-C-0033, NASA Grants NCC 2-274 and NCC 2-220, Boeing Grant W271799, an AT&T Bell Laboratories CRFP Fellowship, and a gift from Lockheed Corp.

* To whom correspondence should be addressed.
‡ Knowledge Systems Laboratory.
§ Stanford Magnetic Resonance Laboratory.
|| Present address: Department of Biological Structure, University of Washington, Seattle, WA.

0095-2338/88/1628-0194$01.50/0

patible with the data. Although multiple runs of adjustment methods can be made from different random starting structures, the large number of variables ($3N$) makes it unlikely that enough runs can be made to generate an adequate random sample.

Adjustment methods do well when the number of variables is small or when the starting structure is near the global minimum. Thus, these methods would be expected to work reasonably well for small proteins such as those reported in the literature or when the starting structure is already nearly correct.

Much of the work with current methods has therefore gone into obtaining a good starting structure. For example, in DISGEO,[4] an initial distance matrix is smoothed by using the triangle and inverse triangle inequality prior to embedding the structure in XYZ space. The tighter these bounds can be made, the closer the embedded structure will be to the final structure, the better the adjustment procedure will converge, and the better a representative set of starting structures can be obtained. Unfortunately, for realistic NMR data sets simple bounds smoothing does not usually reduce the bounds enough to allow a fully representative set of starting structures.

In the DISMAN program[5] the adjustment is done in dihedral angle space, but not with all dihedral angles at once since the procedure would not converge. Instead, starting with a random configuration, short-range constraints are first satisfied in order to obtain the local structure, after which longer range constraints are satisfied. Although this approach ensures better convergence, the difficulty is that there are too many random starting structures to guarantee a representative set of conformations compatible with the data.

In the exclusion paradigm, compatibility with experimental and theoretical data is checked by functions that can evaluate the ability of a structure to explain the data. The rating functions have three properties in contrast to the evaluation functions used by adjustment methods: (1) They need not be in a form that specifies an adjustment—they need only specify an "accept/reject" criterion or a figure of merit. (2) They need not simultaneously check all constraints, since they can be applied sequentially. (3) They can be applied in parallel, since they can be formulated as independent "exclusion tests".

An important characteristic of exclusion methods is that a family of legal conformations will be retained if more than one conformation is compatible with the data that have been introduced. Although not all these conformations will be able to predict the actual NMR data sets (as defined by the Bloch equations), it is likely that some of them will, since all structures consistent with the constraints have been retained. This set of conformations, if properly sampled, can be taken as an approximation of the $3N$-dimensional density function for the molecule.

The major difficulty for implementing exclusion methods is to find tractable ways to enumerate and test all possible conformations. The set of conformations compatible with the data could in theory be obtained by systematically exploring the $3N$-dimensional conformational space and enumerating those conformations that satisfy the data. This simple strategy is, however, computationally intractable. For example, if a protein has 1000 atoms and each atom is uniquely positioned, then there is only a single conformation. If each atom can be in any of 100 locations, then, in principle, there are 100^{1000} conformations, a number that is already too large to be tested on any computer.

However, the vast bulk of these conformations are not compatible with chemical bond constraints and the applied experimental constraints, which implies that the actual number of legal conformations may be of manageable size. The goal for an exclusion method is therefore to exclude large numbers of potential conformations without exhaustively enumerating them. In this paper we demonstrate that, by intelligent elimination of large numbers of conformations at once, the exclusion paradigm is in fact computationally feasible. We describe our current development of a method, called *heuristic refinement*, and its implementation in a computer program called PROTEAN. Earlier publications have described various aspects of the program in less detail.[12-19]

An important aspect of this method is that abstractions of both atoms and data are introduced in order to achieve computational tractability. Although these abstractions are quite reasonable, they nevertheless introduce a certain amount of bias, meaning that the final structures may not completely satisfy the constraints. Thus, the output of PROTEAN is a set of representative starting structures, each of which is nearly correct. Since adjustment methods do well when the starting structure is near the global minimum, any of the current distance geometry methods would be expected to work well to make the final adjustments. Thus, in its current implementation of an exclusion paradigm PROTEAN fills a major gap among current methods, namely, the production of a representative set of starting structures for adjustment by any of the current methods. PROTEAN does not replace these methods, but instead augments them by setting up the conditions under which they work best.

METHODS

The computational complexity of enumerating all possible protein conformations depends on the number of atoms and on the number of possible positions for each atom, where the number of possible positions defines the *accessible volume* of the atom. Therefore, PROTEAN uses four basic techniques to reduce these two numbers before exhaustively enumerating the remaining possibilities.

(1) Problem Decomposition. PROTEAN reduces the number of atoms by breaking the overall problem into subproblems, partially "solves" each subproblem, uses these partial solutions to constrain other subproblems, and then combines the partial solutions.

(2) Problem Abstraction. PROTEAN reduces the number of atoms by grouping locally constrained sets of atoms, such as those forming side chains or secondary structures, and considers the entire group as an abstract object before considering detailed locations.

(3) Local Satisfaction of Constraints. PROTEAN reduces the size of the accessible volume for each atom by sequentially applying constraints between pairs of objects rather than all objects at once.

(4) Heuristic Control. At each point in the problem solving, PROTEAN chooses that action which is likely to exclude the largest number of potential structures.

The heuristic refinement method therefore refines the protein along two main dimensions, that of structural detail and that of accessible volume. It uses heuristics to control the order of refinement operations to obtain the greatest efficiency, but if each refinement operation does not remove structures that are part of the solution, then the order of operations should not affect the corrections of the result.

Input. PROTEAN utilizes three kinds of input: experimental data, standard parameters of chemical structure, and method-specific parameters.

(1) Experimental Data. PROTEAN is designed to accept any experimental data that can be expressed as constraints on the relative positions of one or more atoms in the protein. The current implementation of PROTEAN uses the following data:

- **Primary structure** indicates the complete connectivity of all atoms in the protein, obtained from protein sequence analysis.

• **Secondary structure** indicates groups of locally constrained atoms such as helices or β-strands. This information can be obtained visually from the NMR spectrum[20] or with the aid of an expert system, developed in our laboratory, that uses patterns of NMR information to infer the locations of secondary structure.[21]

• **Short- and long-range NOE information** indicates that some protons are within 2–4 Å of each other. We assume that the peaks in the NMR spectra have been assigned and can be mapped to specific protons within the primary structure. This is the primary source of information about the three-dimensional relationships between atoms.

• **General distance constraints** indicate that two atoms are within some range of distances from each other. For example, these can be derived from the constraints on maximum distance that are implied by covalent connectivity of atoms or by fluorescence transfer experiments.

• **Volume** indicates that conformations exceeding certain dimensions are unlikely.

• **Surface information** indicates that some atoms or side chains are on the surface of the protein and others are buried within the protein.

(2) Standard Parameters of Chemical Structure. The system uses generally accepted chemical approximations for van der Waals hard sphere atomic radii, bond lengths, and peptide bond angles.

(3) Method-Specific Parameters. PROTEAN also requires an indication of the desired resolution of the final answer. Since we use an exclusion paradigm, we start with a full set of possible structures. The current implementation of PROTEAN discretely samples the conformational space, so it is necessary to specify the sampling grain size of the final set of conformations. For highly constrained proteins, we might select a sampling grid with 1.0-Å intervals. For underconstrained proteins a sampling of 4 Å or more provides sufficient granularity for observing the variation in the position of a subunit.

The selection of final grain size is an important factor in determining the run time of PROTEAN. PROTEAN dynamically varies grain size during problem solving for purposes of efficiency. Initial placement is done by coarse sampling. Once a general region of occupancy has been defined, PROTEAN samples the space more finely in order to achieve the desired grain size. Thus, we do not try to solve the entire problem in full detail from the start.

Output. The output of PROTEAN is a set of structures that are internally consistent with all the supplied constraints. In effect, it provides a discrete sample of the $3N$-dimensional distribution of possible conformations. If the variance in position for all atoms is small, then each sampled conformation will have small root mean square (RMS) differences with all the others. If the variances are large (as for the case of severely underdetermined problems), then the conformations may be quite different.

Geometric Transformations. The basic representation utilized by PROTEAN is a series of geometric objects, each described by a set of points given within a local coordinate system. The possible spatial locations of these objects are described by the following elementary geometric transformations.[22]

Location of an Oriented Object. The location of an oriented object in three dimensions relative to a fixed coordinate system can be described as a series of rotations of the object to define orientation, followed by a translation of the object to its final position. We use the three Euler angle representation of orientation (ϕ, θ, ω)[22] in which the first angle, ϕ, is a rotation around the global Z axis, the second angle, θ, is a rotation around the global Y axis, and the third angle, ω, is another rotation around the global Z axis. The translation component is represented by a three-dimensional vector, $\mathbf{x} = (x,y,z)$.

The possible locations of an object with respect to a fixed coordinate system can, therefore, be systematically sampled by enumerating all possible values of $(x,y,z,\phi,\theta,\omega)$. Given a local, movable coordinate system oriented around an object, then the global position, $\mathbf{a}'$, of any point, $\mathbf{a}$, in the movable coordinate system can be calculated by a matrix multiplication

$$\mathbf{a}' = \mathbf{T}\mathbf{a}$$

where $\mathbf{T}$ is a 4×4 matrix that is the matrix product of 4×4 matrices representing each of the elementary rotations and translations

$$\mathbf{T} = f(x,y,z,\phi,\theta,\omega)$$

These transformation matrices can be multiplied to compose spatial relationships. For example, let $\mathbf{T}_{ab}$ be the transformation describing the location of coordinate system B with respect to coordinate system A. Any point $\mathbf{b}$ in B can be converted to a point in A by multiplying $\mathbf{T}_{ab}\mathbf{b}$. If we have another coordinate system, C, whose locations relative to coordinate system B are known, $\mathbf{T}_{bc}$, then the locations of C relative to A are given by

$$\mathbf{T}_{ac} = \mathbf{T}_{ab}\mathbf{T}_{bc}$$

The location of each object in a protein structure, with respect to some global coordinate system, can thus be represented by a 6-tuple of $(x,y,z,\phi,\theta,\omega)$. PROTEAN makes use of this relation by defining the positions of related atoms in local coordinate systems, such as those of the secondary structures, and then introducing the local coordinate systems into a global coordinate system defined by declaring one object to be a fixed *anchor*. Given the positions in the global coordinate system of the atoms comprising two such objects, we are able to test the locations of the two objects for satisfaction of distance and other constraints.

Rotation around an Axis. A second elementary transformation is rotation about an axis, utilized to define the positions of atoms distal to rotatable bonds. For example, given an arbitrary axis defined by a unit vector $\mathbf{k}$ located at a point $\mathbf{x} = (x,y,z)$, a rotation of θ deg of point $\mathbf{p}$ to point $\mathbf{p}'$ is given by[22]

$$\mathbf{p}' = \mathbf{R}\mathbf{p}$$

where $\mathbf{R}$ is a 4×4 matrix representing the rotation of a point about an arbitrary axis

$$\mathbf{R} = f(\mathbf{k},\mathbf{x},\theta)$$

Thus, the space of possible positions for an atom after rotation of a bond (all other bonds fixed) can be calculated by sampling the rotation angle θ at some reasonable increment such as 20°.

The matrices $\mathbf{T}$ and $\mathbf{R}$ are examples of homogeneous transforms,[23] which in this case consist of a rotation matrix in the upper left 3×3 submatrix and a translation vector in the fourth column. An important advantage of this representation is that many different types of spatial relationships, such as translations and rotations, may be multiplied together to create a single transform representing an arbitrarily complex spatial relationship. PROTEAN makes use of this fact to relate different parts of the molecule to each other.

Geometric Operations. Each homogeneous transform describing the relationship of one object to another is equivalent (and can be converted to) a *location* $(x,y,z,\phi,\theta,\omega)$. Associated with any object in a coordinate system is a list of locations that constitutes its accessible volume in that coordinate system. For objects without orientation only the first three components are meaningful.

PROTEAN currently represents the spatial distribution of objects by sampling space discretely and by retaining locations that are compatible with the constraints implied by the data. There are obvious alternatives (for example, the description

of the distribution function as a continuous volume in space or as a parameterized distribution), and we are examining these. However, the advantage of the discrete representation is that the only assumption is that enough sampled locations can be retained to adequately model the distribution.

Conceptually, every object begins with an infinitely large accessible volume in a coordinate system and has its accessible volume reduced by testing locations and excluding those that are incompatible with the constraints. When all or a set of accessible volumes becomes small enough, the remaining possible conformations can be enumerated.

PROTEAN uses five basic operations to manipulate the geometric objects in order to generate the family of structures compatible with the constraints. These operations are coherent instance generation, anchor, yoke, append, and prune. Much of the strategy of PROTEAN can be seen as the intelligent selection of these operations on particular sets of geometric objects in order to exclude large classes of conformations without having to enumerate all possible structures. Each of the five operations is performed within the context of a subproblem called a *partial arrangement*, which is a set of geometric objects representing parts of the protein, an initially implicit list of locations defining the accessible volume of each object, and a set of constraints between the objects. The accessible volumes of the objects are defined with respect to one of the objects, called a fixed *anchor*. Since the complete set of atoms, together with all measured distance constraints, is a partial arrangement that defines the complete problem, the five operations could be performed directly on all the atoms. In particular, the coherent instance generation procedure could be applied to exhaustively enumerate all possible atom positions. However, because of the computational complexity of considering all atoms at once, groups of atoms are combined into separate partial arrangements, the five operations are applied to reduce their accessible volumes, and the partial arrangements are then combined.

Coherent Instance Generation. The set of discretely sampled accessible volumes of individual components of a molecule implicitly contains all conformations. We must sample explicitly from these distributions in order to generate single structures that are internally consistent. A *coherent instance* of the objects in a partial arrangement is a set of locations (one for each object) such that all the constraints among the objects are satisfied and constitutes a single conformation of the protein or part of the protein.

The complete set of conformations can be enumerated from the set of individual accessible volumes with the following procedure:

For location 1 in accessible-volume-for-object-1 do
  For location 2 in accessible-volume-for-object-2 do
    For location N in accessible-volume-for-object-N do
    1. Position each object i in location i
    2. Check all constraints between objects 1 and N
    3. Accept the set of N locations as a legal coherent instance if all constraints are satisfied.

In this simple generate-and-test procedure the number of possible coherent instances is related to the number of objects, N, and the average number of locations in each accessible volume, L, by L^N. In actual practice we use a procedure called *backtrack search*[24] to reduce the number of possible coherent instances, but the procedure is still very expensive. It is therefore critical to reduce the number of locations in each accessible volume before coherent instance generation is performed. The operations described in the following four sections allow small subsets of the objects to be considered befor all objects are considered at once. These operations, discussed in a more theoretical fashion elsewhere,[13,25] are examples of a general class of algorithms that have been developed in artificial intelligence for dealing with constraint satisfaction problems.[26,27] The operations are discussed with reference to Figure 1.

Anchor Operation. The ANCHOR operation is used to determine the initial set of locations for a movable object, called the *anchoree*, relative to the fixed anchor in a partial arrangement. It uses distance constraints between fixed points in the two coordinate systems to define the accessible volume. For the 6-dimensional location specification it can be written as

ANCHOR(constraints, object)
  For x from $x$min to $x$max by $x$inc do
    for y from $y$min to $y$max by $y$inc do
      ...
        for omega from omegamin to omegamax by omegainc do

1. Transform anchoree points by $\mathbf{T} = f(x\ y\ z$ phi theta omega) in order to position the anchoree at location $L = (x\ y\ z$ phi theta omega).
2. Save L if all constraints between the anchor and the anchoree are satisfied at this location.

$x$min and $x$max define the range of search through the x coordinate by increments of $x$inc (and similarly for the other parameters). The constraints are checked by transforming all points of the anchoree to location L, calculating the distances between anchor and anchoree points, and determining if the distances fall within the ranges specified by the constraints. An additional requirement is that the objects do not intersect, which is a crude van der Waals constraint. In practice, not all locations L need be generated since the distance constraints place limits on the x,y,z position components of any satisfiable location. The resulting discrete list of locations is a representation for the accessible volume of the anchoree with respect to the anchor. A display program can draw the anchoree at each of these locations, resulting in a *cloud* or *halo* that gives a visual indication of the accessible volume.

Yoke Operation. The yoke operation is used to reduce the accessible volumes of two movable anchorees by applying constraints between them. It removes a location L from one accessible volume if L is incompatible with *all* locations in the other accessible volume. This removal effectively excludes every conformation that contains L without requiring an exhaustive enumeration of those conformations.

YOKE(A,B constraints)
Clear all marks on locations in accessible volumes A and B.
For each location A_i in accessible volume A do
  For each location B_i in accessible volume B do
  1. position object A at location A_i
  2. position object B at location B_i
  3. If all constraints are satisfied between A and B then Mark(A) and Mark(B)
Finally,
Remove all unmarked locations from the two accessible volumes.

The yoke operation is called many times on a set of objects related by constraints. For example, in Figure 1, suppose object B has its accessible volume reduced by a yoke operation with object C. If B also has constraints with a third object D, then the yoke operation may be applied between objects B and D. If this operation further reduces the accessible volume of object B, then the original YOKE operation between B and C must be repeated. Thus, for a large number of objects related by constraints, the YOKE operation is performed very often. This procedure is the primary mechanism by means of which constraints between pairs of objects are able to propagate their influence to the overall structure.

Append Operation. An append operation is an alternative method to anchoring for defining an initial accessible volume.

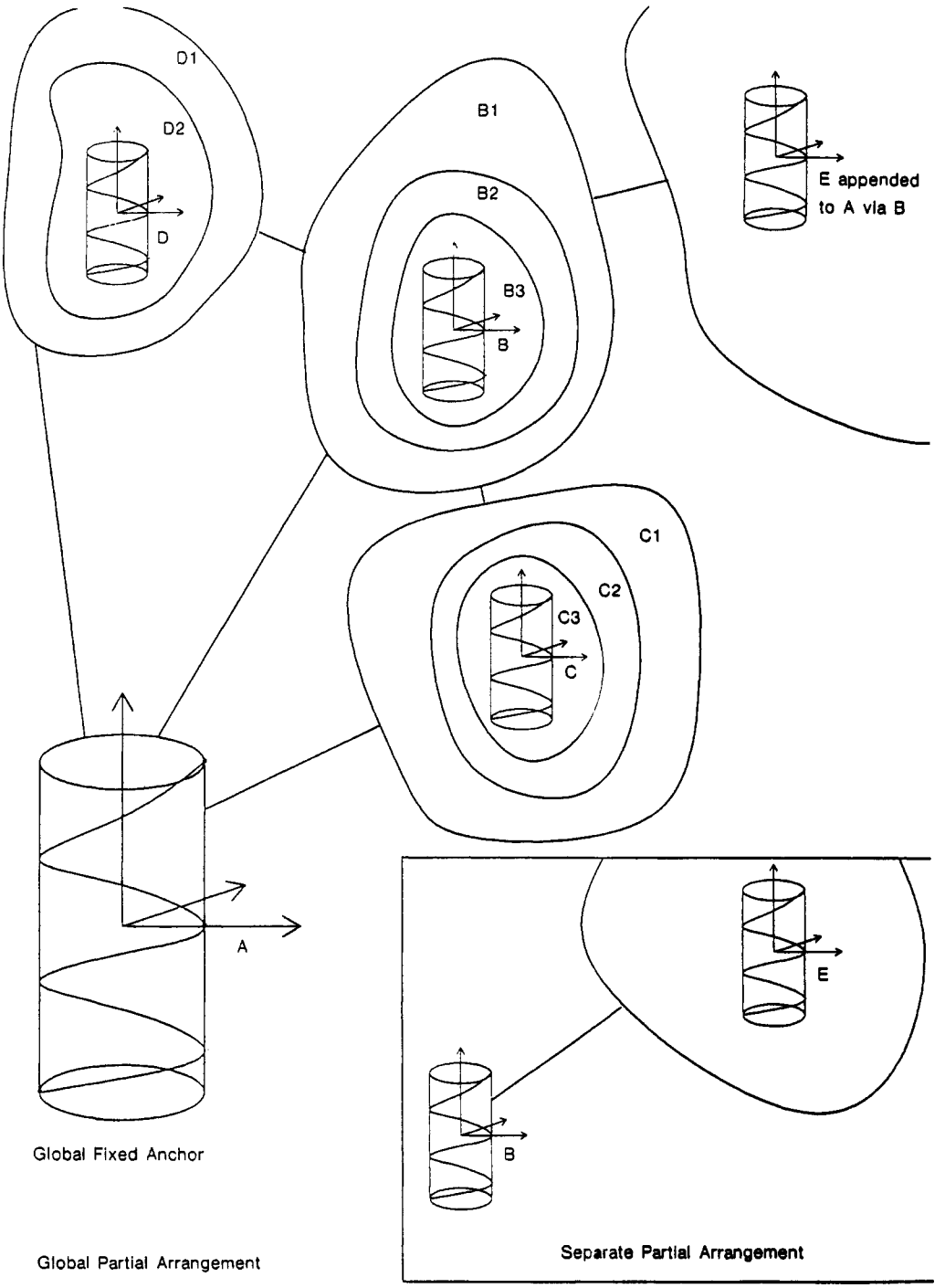

Figure 1. Anchor, yoke, and append operations. Object A is the global fixed anchor. Objects B, C, and D are first anchored to A, producing accessible volumes B_1, C_1, and D_1. B and C are yoked, producing accessible volumes B_2 and C_2. B is yoked with D, producing D_2 and B_3. B and C are then reyoked, producing B_3 (no change) and C_3. E is first anchored to B in a separate partial arrangement and then appended to A via B.

Given the accessible volume of an object B relative to an object A and the accessible volume of a third object E relative to B, then the accessible volume of E relative to A can be found by simply multiplying the transforms corresponding to the two given accessible volumes. This operation is used to combine separate partial arrangements of the protein into a larger arrangement.

APPEND (A,B,E)
For location AB_i in accessible volume of B relative to A do
  For location BE_j in accessible volume of E relative to B do
    Save location $AE_{ij} = AB_i \times BE_j$.

Prune Operation. If constraints to the fixed coordinate system are introduced after an object already has an accessible volume, then each of the locations in the accessible volume can be checked against the new constraints to test for compatibility.

PRUNE(A, Constraints)
For location A_i in accessible volume A do
1. Transform object to A_i
2. Save A_i if Constraints are satisfied

PROTEAN **Strategy.** The overall strategy of PROTEAN is to divide the problem of determining structure into subproblems, each of which constitutes a separate partial arrangement of subparts of the protein represented at various levels of abstraction. Combinations of the anchor, yoke, prune, and append operations are used to reduce the accessible volumes of the objects in the partial arrangement relative to the anchor. Partial arrangements are combined by the append operation,

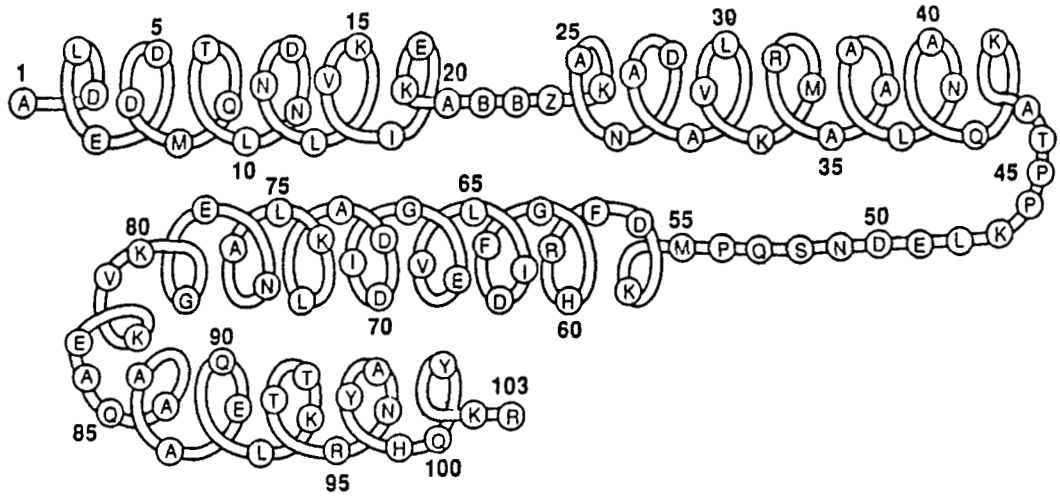

Figure 2. Primary and secondary structure of cytochrome *b*562.

and coherent instances are selected from the reduced accessible volumes.

The overall strategy used in this paper may be divided into the following steps. There are many other possible strategies since each operation simply excludes possible conformations. We are developing more elaborate control mechanisms for experimenting with different strategies in the belief that the optimal strategy will vary with each protein. We are also developing methods to extend these strategies in order to more precisely constrain the structure.

(1) Represent each secondary structure as a solid object that most closely approximates the expected shape of the secondary structure. If no single solid matches the expected shape, then divide the secondary structure into two or more solid subunits.

(2) Determine the initial accessible volume for each amino acid side chain with respect to a coordinate system associated with the peptide of the amino acid. The accessible volumes for each atom within the 20 standard side chains are precalculated by an anchor operation and stored in a side-chain library.

(3) Refine the side-chain accessible volumes in the coordinate system of each secondary structure or secondary structure subunit. The initial side-chain accessible volumes are transformed from the peptide coordinate system to that of the secondary structure by an append operation. Constraints between side chains and the backbone, all within the same secondary structure, are used to reduce the side-chain accessible volumes with the yoke operation.

(4) Abstract the constraints between side chains in different secondary structures with approximate accessible volumes for the side chains.

(5) Determine accessible volumes for each secondary structure, within a global coordinate system defined by one fixed secondary structure, by using the abstract constraints and multiple applications of the anchor and yoke procedures.

(6) Refine the accessible volumes of the secondary structures, within the global coordinate system, by using more detailed accessible volumes for the side chains.

(7) Select coherent sets of secondary structures to define the general topology of the molecule.

(8) Generate consistent backbone traces for each coherent conformation within the global coordinate system by using the locations of the secondary structures as a guide for placing the backbone.

RESULTS: VALIDATION ON CYTOCHROME *b*562

In this section we give the details of our current implementation of the strategy outlined in the previous section. To make the operations more concrete, we also discuss their use

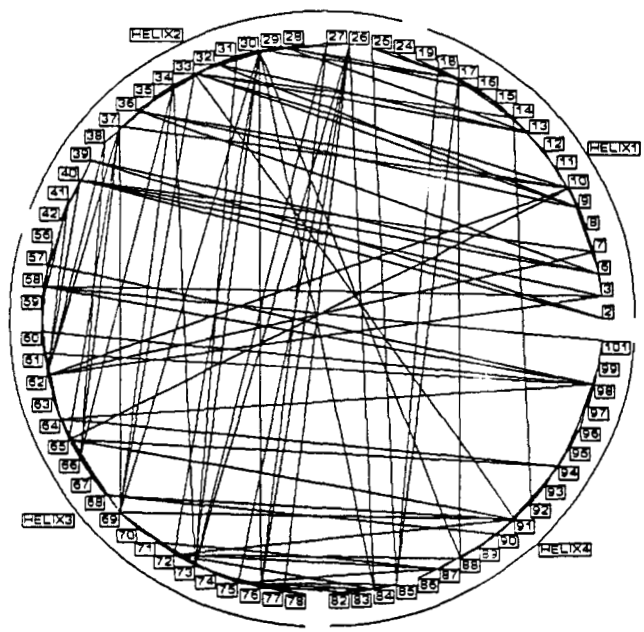

Figure 3. Constraint network for cytochrome *b*562. Lines between amino acids represent constraints inferred from the data or from knowledge of protein structure. Constraints are only shown for helices.

in the context of cytochrome *b*562, a protein of 103 amino acids the structure of which is known from X-ray crystallographic studies.[28] Cytochrome *b*562 was chosen as an example between it is fairly large and may not be handled well by current distance geometry methods. It contains well-defined secondary structures and several coil elements. It therefore exercises the basic capabilities of PROTEAN without introducing additional complications that would make a description of the results more difficult. In this example, we do not demonstrate use of β-strands or the application of volume and surface constraints. These are discussed elsewhere.[12]

We started with the known crystal structure from the Brookhaven Protein Data Bank (PDB).[28] Both the primary and secondary structure were obtained from this file. Hydrogen atoms were added to the structure by a modified version of the MIDAS *addhydrogen* program.[29] Simulated NOEs were generated by examining all inter-hydrogen distances in the crystal structure and retaining those stereochemically distinct distances of less than 4 Å. A total of 729 distance constraints were generated in this manner. Figure 2 shows the primary and secondary structure as obtained from the PDB file. Figure 3 schematically shows the simulated constraints between amino acids.

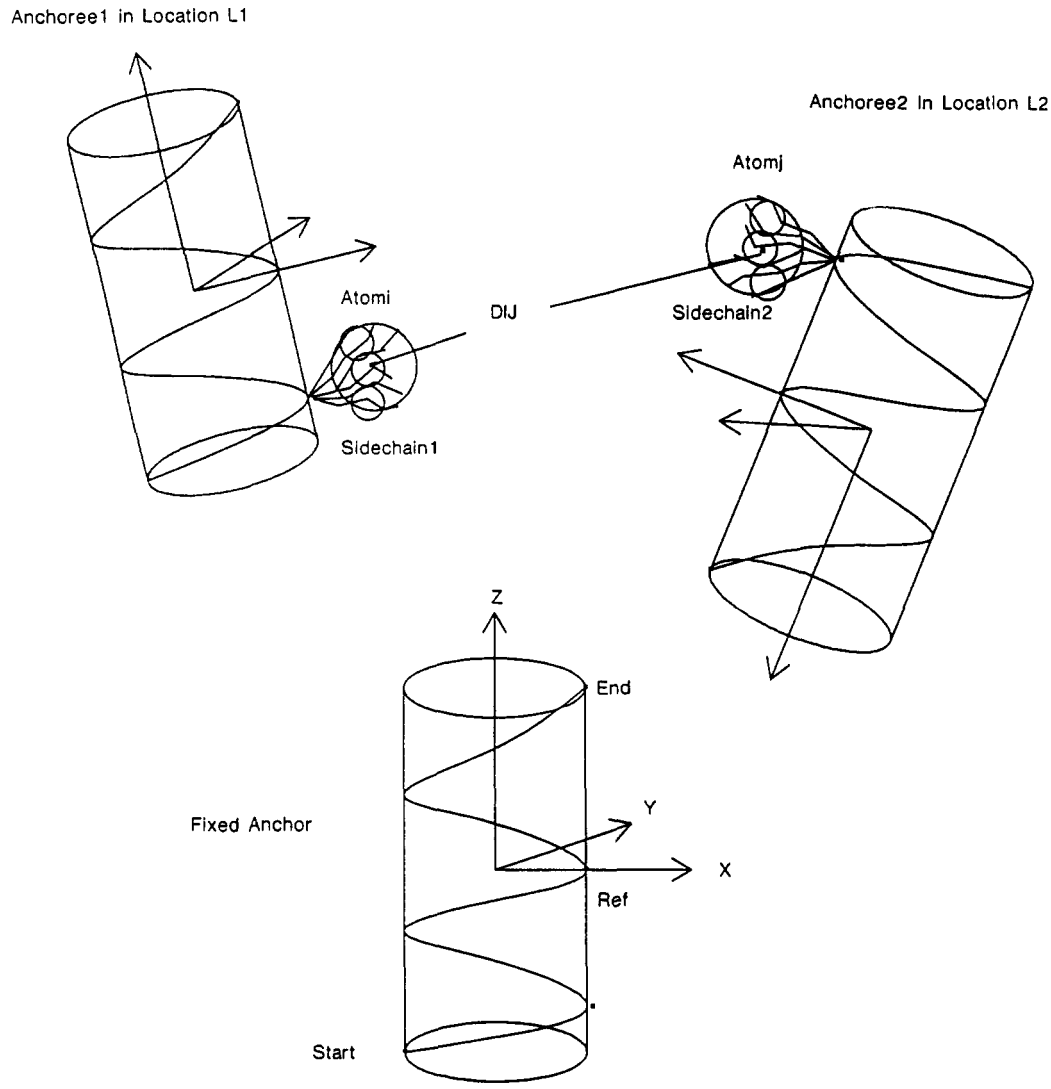

Figure 4. Several abstractions used in PROTEAN. Each helix is represented by a regular cylinder in a local coordinate system defined by three amino acids Start, End, and Ref. The positions of an atom i in a sidechain 1, with respect to a secondary structure coordinate system, can be abstracted first as a large sphere for coarse distance checks and then as a set of smaller spheres for more refined checks. Distance checks are made by transforming points of anchoree 1 and anchoree 2 into the global coordinate system of the fixed anchor and then determining the distance D_{ij} between the transformed points.

Represent Secondary Structures as Solid Objects. The subsequences of amino acids that form a secondary structure can be grouped together to form relatively rigid abstract objects, since a secondary structure is by definition a well-defined local arrangement of backbone peptides. In the current implementation we assume that the backbone peptides are all fixed in their ideal configurations.

Each secondary structure is defined within a local Cartesian coordinate system centered at a fixed REFerence amino acid near the midpoint of the sequence (Figure 4). The REFerence is calculated from the STARTing and ENDing amino acids of the secondary structure so that it is as close as possible to the middle of the sequence. The secondary structure coordinate system is defined such that the REFerence is a point on the positive x axis, and the z axis is the axis of the secondary structure.

If there is a great deal of regularity in the sequence (as is the case for α-helices), then many amino acids can be included in the solid, since the variation from ideality is relatively small. If there is less regularity in the sequence (as is the case for β-strands), then the structure may have to be broken into separate subunits, each consisting of only a few amino acids. In the limiting case of no regular structure (random coils), then only one amino acid may be included in each solid.

Table I. Best Fit of Ideal Helix α-Carbons to Crystal Helix α-Carbons (Angstroms): RMS, Minimum Distance, Maximum Distance

	RMS	min	max
helix 1	1.16	0.74	1.71
helix 2	1.78	0.94	3.41
helix 3	1.04	0.43	1.92
helix 4	1.43	0.42	1.90

For cytochrome $b562$ there are four helices and five intervening random coils, the longest of which is 13 residues (Figure 2). Each of the four helices was instantiated as a cylinder with the ideal parameters for an α-helix: radius 2.3 Å, rise rate (distance between α-carbons along the axis) 1.5 Å, and rotation between α-carbons 100°.

To test the match between the ideal helices and the actual atomic coordinates, we calculated the RMS error between the ideal helices and the actual helices in the crystal structure. We did this by superimposing the centroids of each set of α-carbons, determining the best rotation about the centroid to minimize the RMS error using the singular value decomposition method of McLachlan,[30] and then adjusting the fit by a local grid search optimization.

Table I shows the RMS deviation between the α-carbons of the ideal helices and the corresponding crystal helices. The

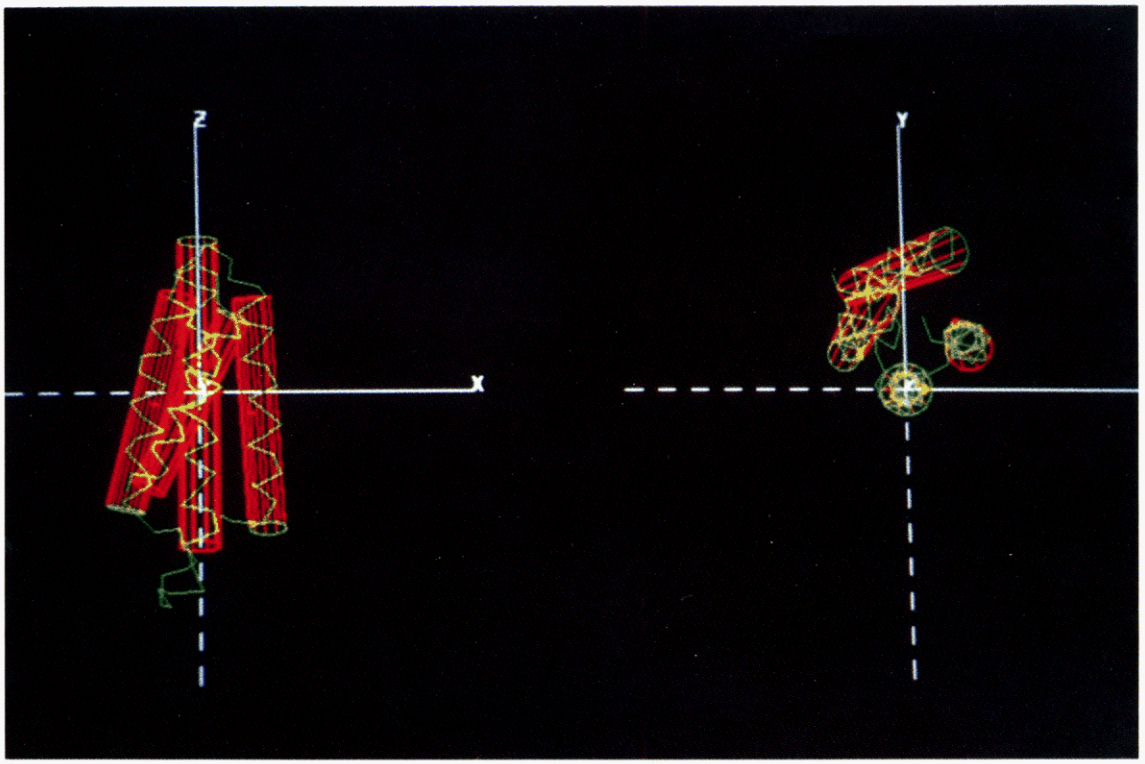

Figure 5. Ideal solid abstraction of helices superimposed on crystal structure of cytochrome *b*562, two views, α-carbons only.

average error was 1.3 Å. The actual distances between corresponding α-carbons ranged from 0.43 to 3.41, with the largest distances found between carbons at the end of the helices (where the assumption of ideality is least defensible). These discrepancies between ideal and actual α-carbons could, in the worst case, lead to violation of distance constraints by as much as 6 Å. For this study, constraints that were not satisfied by the ideal solid positions were deleted.

We are currently developing methods for relaxing the ideality assumption by using intrasecondary constraints and anchor and yoke operations to define nonfixed accessible volumes for the secondary structure peptides. We could also simply break up the helices into smaller subunits as we do for β-strands. However, for underconstrained data obtained from NMR there may not be enough information to allow the exact topology of the secondary structures to be obtained. In this case the assumption of ideal secondary structure is adequate.

Figure 5 shows the crystal backbone α-carbons and the superimposed ideal solid positions in the coordinate system of helix 3, demonstrating that the ideal assumption is reasonable as a first approximation.

Determine Side-Chain Accessible Volumes in Peptide Coordinate Systems. Coordinates for atoms comprising each of the 20 amino acids were obtained from the standard amino acid coordinates of the Protein Data Bank. The axis rotation transform was then used to discretely rotate all free bonds of the side chain, generating a set of side-chain configurations with respect to a peptide coordinate system with origin at the α-carbon of the amino acid and z axis along the α–β bond (Figure 6). These configurations defined the initial side-chain accessible volumes as generated by an anchor operation in dihedral space rather than Cartesian space.

Each side-chain χ angle was sampled through its Ramachandran range at varying intervals depending on the distance of the rotatable bond from the α-carbon. The sampling intervals ranged from 10 to 120° and were chosen so as to generate those angles corresponding to energy minima of the side chains. All side-chain conformations that did not violate intra-side-chain van der Waals constraints (using a hard-sphere

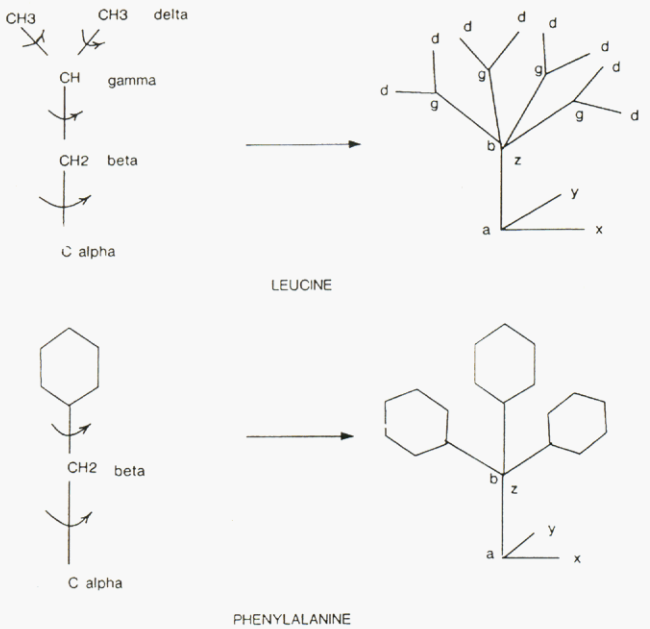

Figure 6. Generation of initial side-chain accessible volumes for leucine and phenylalanine. Rotatable bonds are systematically sampled, and the resulting set of discrete configurations is stored as a reusable side-chain library. Each configuration defines a location of the side chain in its own peptide coordinate system, with z axis along the α–β bond.

approximation) were retained in the list of legal conformations. The column labeled "initial" in Table II shows the number of configurations for the helix side chains that were generated in this manner. These configurations were stored in a side-chain library for reuse on other proteins.

Refine Side-Chain Accessible Volumes in Secondary Structure Coordinate Systems. The side-chain accessible volumes were positioned within the coordinate system of each secondary structure by a simple transformation of coordinate systems (an append operation). The side-chain conformations were

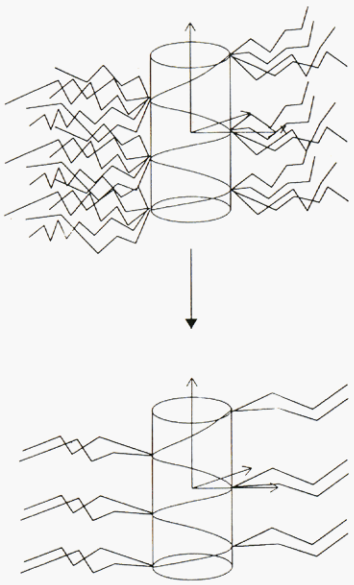

Figure 7. Reduction of side-chain accessible volumes utilizing only intrahelix constraints. (Top) The initial side-chain accessible volumes are transformed to the helix coordinate system by an append operation, thereby defining a partial arrangement in which the side chains are the objects, the individual side-chain configurations are the locations, and the NOE constraints between side chains on the same helix are the constraints. (Bottom) Application of the yoke procedure removed some side-chain configurations that are not compatible with any other side-chain configuration.

further reduced (with the yoke operation) by application of van der Waals and NOE constraints to other side chains and to the backbone of the secondary structure (Figure 7).

Table II shows the initial and refined number of side-chain configurations for each helix. The percent reduction in configurations ranged from 0 for alanine to 50 for Arg34, with an average 17% reduction. The green cloud in Figure 8 shows the initial accessible volume for one amino acid, Leu30 (corresponding to 144 configurations of the side chain), after it has been transformed to the coordinate system of helix 2. The orange cloud shows the reduced accessible volume of Leu30 (corresponding to 120 side-chain configurations) after yoking it with the helix 2 backbone and with other side chains of helix 2 (a 17% reduction).

These results show that intrasecondary structure constraints can be a powerful means of reducing the number of possible locations for side chains. They also show that some side chains are not reduced much at all. Since the computational expense of anchor and yoke operations depends on the number of objects and the sizes of their accessible volumes, it may be more efficient to not reduce the accessible volume of all side chains at once. This tradeoff is an example of the potential use of heuristic control in increasing the efficiency of the program.

Determine Abstract Constraints between Secondary Structures. It would be prohibitively expensive to check all possible locations of every side chain on one helix with respect to all possible locations of every side chain on another helix. It is possible, however, to summarize the side-chain accessible volumes for initial processing. Given a set of atom positions defined by a side-chain accessible volume with respect to a secondary structure coordinate system, we can find the average position of the atom and a maximum distance from this average position. For example, in Figure 4, suppose side chain 1 of anchoree 1 has the accessible volume shown. Then the accessible volume of an atom i in side chain 1 can be represented by a single large sphere with center at the mean of the atom positions and with radius such that all atom positions are enclosed by the sphere. The accessible volume of an atom j of side chain 2 of anchoree 2 can be represented in a similar manner. This process of representing a complex set of objects by a simpler one we call abstraction.

These positions and uncertainties can be used to rule out large numbers of conformations before more detailed checking of the remaining ones. For example, in Figure 4 a constraint stating that two atoms must be less than 4 Å apart is satisfied

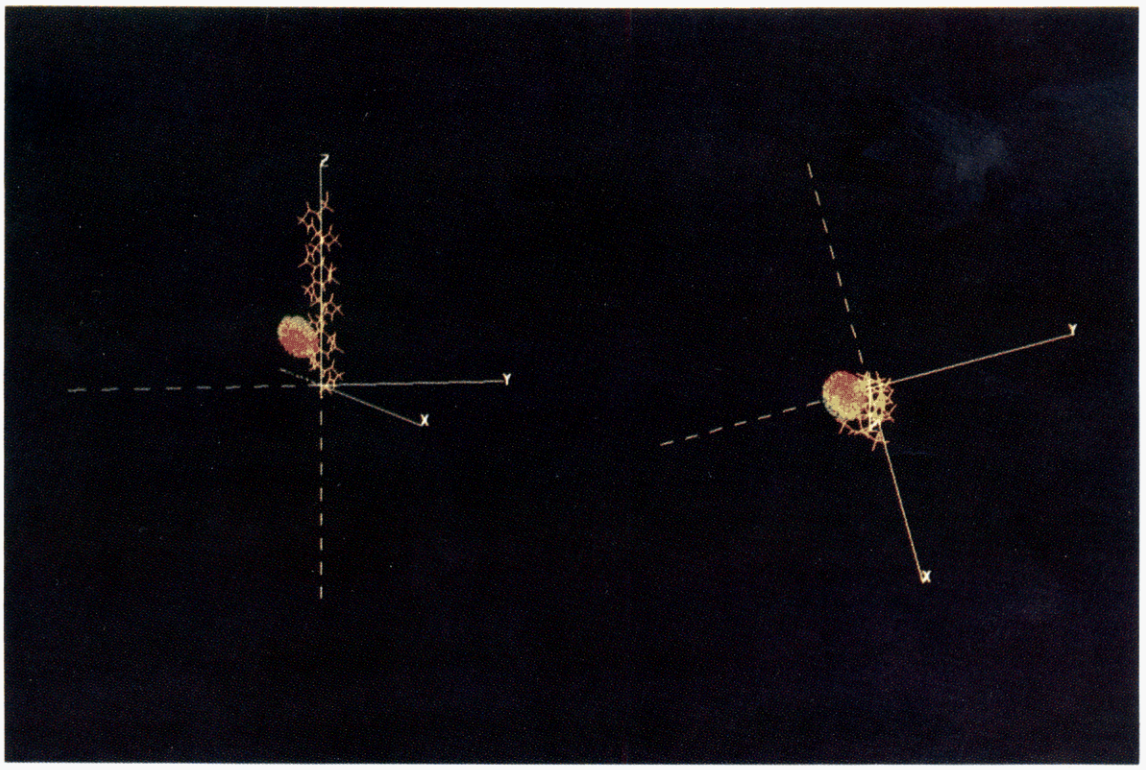

Figure 8. Initial accessible volume of Leu30 appended to coordinate system of helix 2 (green cloud). Refined accessible volume after taking into account intrahelix constraints (orange cloud). Two views.

Table II. Refinement of Side-Chain Accessible Volumes in Secondary Structure Coordinate Systems[a]

residue	initial	refined	%	residue	initial	refined	%
Asp6	144	126	13	Asp63	144	114	21
Met7	216	142	34	Ile64	144	126	13
Gln8	216	147	32	Leu65	144	125	13
Thr9	24	24	0	Val66	24	23	4
Leu10	144	115	20	Glu67	216	149	31
Asn11	144	117	19	Gly68	1	1	0
Asp12	144	119	17	Ile69	144	127	12
Asn13	144	113	22	Asp70	144	117	19
Leu14	144	110	24	Asp71	144	113	22
Lys15	162	96	41	Ala72	1	1	0
Val16	24	24	0	Leu73	144	117	19
Ile17	144	121	16	Lys74	162	101	38
Ala25	1	1	0	Leu75	144	109	24
Asn26	144	138	4	Ala76	1	1	0
Asp27	144	116	19	Asn77	144	114	21
Ala28	1	1	0	Glu78	216	148	31
Ala29	1	1	0	Gly79	1	1	0
Leu30	144	120	17	Ala84	1	1	0
Val31	24	24	0	Gln85	216	152	30
Lys32	162	105	35	Ala86	1	1	0
Met33	216	141	35	Ala87	1	1	0
Arg34	243	121	50	Ala88	1	1	0
Ala35	1	1	0	Glu89	216	147	32
Ala36	1	1	0	Gln90	216	150	31
Leu38	144	115	20	Leu91	144	125	13
Asn39	144	110	24	Lys92	162	97	40
Ala40	1	1	0	Thr93	24	24	0
Lys56	162	123	24	Thr94	24	24	0
Asp57	144	141	2	Arg95	243	133	45
Phe58	144	136	6	Asn96	144	112	22
Arg59	243	136	44	Ala97	1	1	0
His60	144	81	44	Tyr98	144	90	38
Gly61	1	1	0	His99	144	83	42

[a] Initial is number of configurations in side-chain library. Refined is number remaining after intrasecondary structure constraints and anchor, yoke operations are utilized. Percent is percent change. Not all side chains were placed. Helix 1: Asp2–Lys19. Helix 2: Lys24–Lys42. Helix 3: Lys56–Gly79. Helix 4: Lys82–Lys101.

at this abstract level if, for two helix locations, the distance d_{ij} between the average atom positions is less than $4 + r_i + r_j$, where r_i and r_j are the radii of the two enclosing spheres. The two helix locations may in fact not be compatible when more refined constraints are tested, but any locations that are not compatible at this coarse level of testing will certainly not be compatible at the more refined level.

For cytochrome *b*562 each side-chain hydrogen involved in long-range NOE constraints was summarized with a mean position and a radius based on maximum excursion from the mean. The abstracted positions had an average uncertainty of 4 Å. As a check on the adequacy of the side-chain and ideal helix approximations, the abstracted constraints were tested with the locations determined by the best fit of the ideal helices to the crystal structure. Of the 300 constraints that were so tested, 260 were satisfied by the ideal helix locations. The majority of the unsatisfiable constraints occurred at the ends of the helices. As mentioned previously, the remaining constraints were not included. These unsatisfiable constraints mean that we cannot reconstruct the crystal exactly using a single ideal cylinder for each helix. However, most of these constraints would be satisfied if we were to break each helix into two or more smaller cylinders, and all of them would be satisfied if we were to treat each peptide individually. The number of peptide units that can be grouped together as an ideal secondary structure depends therefore on the degree of precision implied by the data. For very highly constrained data the current results show we may have to break up secondary structures into several subunits, while for less constrained data, as might be expected from NMR, the single ideal approximation should be adequate.

Table III. Refinement of Secondary Structure Accessible Volumes in Global Coordinate System. Three Stages of Processing[a]

stage	helix 1	helix 2	helix 4
initial	3115	119	265
intermediate	1874	117	153
final	254	3	3

[a] Initial is after anchoring 2 and 4, and appending 1 via 2. Intermediate is after yoking with abstract constraints. Final is after yoking with refined constraints.

Determine Secondary Structure Accessible Volumes in Global Coordinate System by Using Abstract Constraints. To determine the accessible volumes for the helical elements, a single secondary structure is chosen as a fixed anchor. The anchor coordinate system becomes a global coordinate system within which all atom positions are eventually described.

The anchor is chosen to have many constraints to the other objects in order to minimize the size of the initial accessible volumes. After introduction of all objects by anchoring (or appending if there are no direct constraints to the fixed object), we can iteratively yoke accessible volumes to reduce them. The order of yoking can affect the efficiency of this process. The most valuable yokes are those that are between objects which have a large number of constraints between them and between objects which have had their accessible volumes significantly reduced since they were last yoked.

In some contexts, the number of objects is so large that separate sets of objects are positioned relative to one another in separate partial arrangements. Only after the accessible volumes have been fully reduced are they combined by using anchor, append, and yoke operations.

In the context of cytochrome *b*562, helix 3 was selected as the center of the fixed coordinate system. Helices 2 and 4 were anchored to helix 3. The Cartesian portion of the location (x,y,z), was sampled in a 64-Å cube at an interval of 1 Å. The orientation component $(\theta\ \phi\ \omega)$ was sampled at 10°. Helix 1 was not well constrained with respect to helix 3 but was very strongly constrained by helix 2. Therefore, helix 1 was first anchored to helix 2 in a separate partial arrangement and then appended to the partial arrangement anchored by helix 3, utilizing the accessible volume of helix 2 with respect to helix 3 and that of helix 1 with respect to helix 2. The remaining constraints between helix 1 and helix 3 were then used to prune the result.

The first row of Table III shows the initial anchor results. In a 64-Å cube, there are 6.11×10^9 possible locations. The search of this space was made more efficient because we were able to use the triangle inequality[4] to exclude certain Cartesian locations without reference to orientation (because they were simply too far from the fixed object). Figure 9 (top) shows the initial anchored accessible volumes for helices 2 and 4 and the appended and pruned accessible volume for helix 1.

Once all three helices had their initial accessible volumes determined, they could be further reduced by repeated application of the yoke procedure until there was no change in the number of locations in each accessible volume. At this point, every location within the accessible volume of a given object was compatible with at least one location for each other object. The second row of Table III shows the size of the accessible volumes after yoking with the single-sphere abstraction of the atomic positions.

Refine Secondary Structure Accessible Volumes in Global Coordinate System by Using Refined Constraints. As the number of secondary structure positions decreases, we can make more detailed checks. For instance, we can approximate the accessible volume of a side-chain atom by a set of three overlapping spheres that together cover the accessible volume (Figure 4). If we are given two helix locations that satisfy the

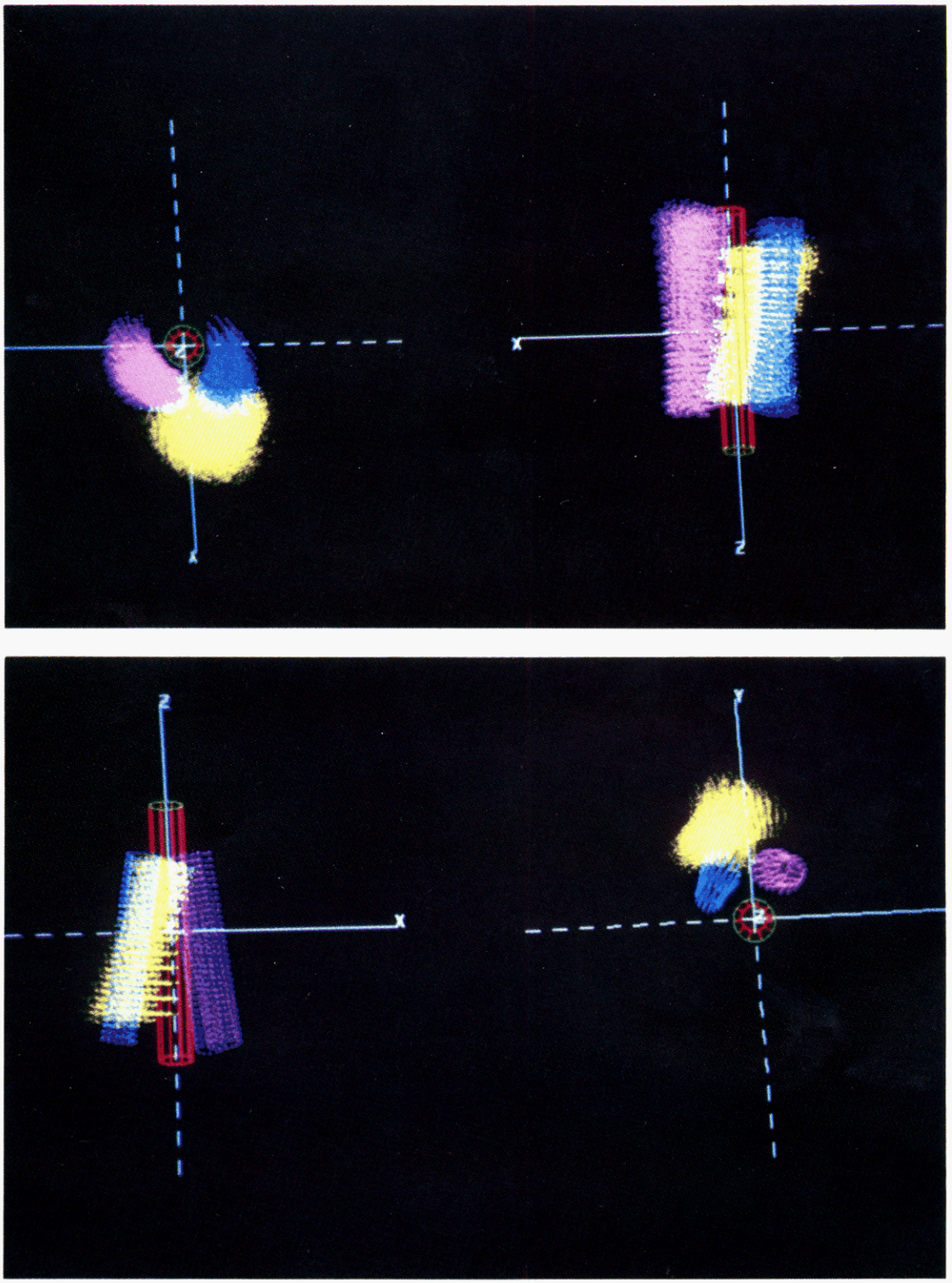

Figure 9. (Top) Accessible volumes for helices 1, 2, and 4 in coordinate system of helix 3 after initial anchor and append operations. (Bottom) Accessible volumes after final yokes with refined constraints.

single-sphere approximation of the constraints, we can further test these locations using the refined description. Given the same requirement that a pair of atoms must be within 4 Å, the constraint is considered satisfied if at least one pair of sphere centers satisfies the condition that d_{ij} is less than $4 + s_i + s_j$, where s_i and s_j are the radii of each of the smaller spheres. This type of constraint is called *disjunctive* since the constraint is satisfied if any of its components are satisfied. Note that the smaller radii of the disjunctive spheres means that each distance must satisfy a tighter constraint, but the cost is that more distances must be checked.

This refinement of constraints can be carried further by approximating the atomic accessible volume by progressively larger numbers of spheres, with smaller and smaller radii. In the limit, each individual sampled point for the atom can be checked individually. This becomes feasible, however, only in later stages of processing when the number of possible conformations is greatly reduced.

In the case of cytochrome *b*562, the use of the more detailed constraints resulted in a dramatic decrease in the size of the

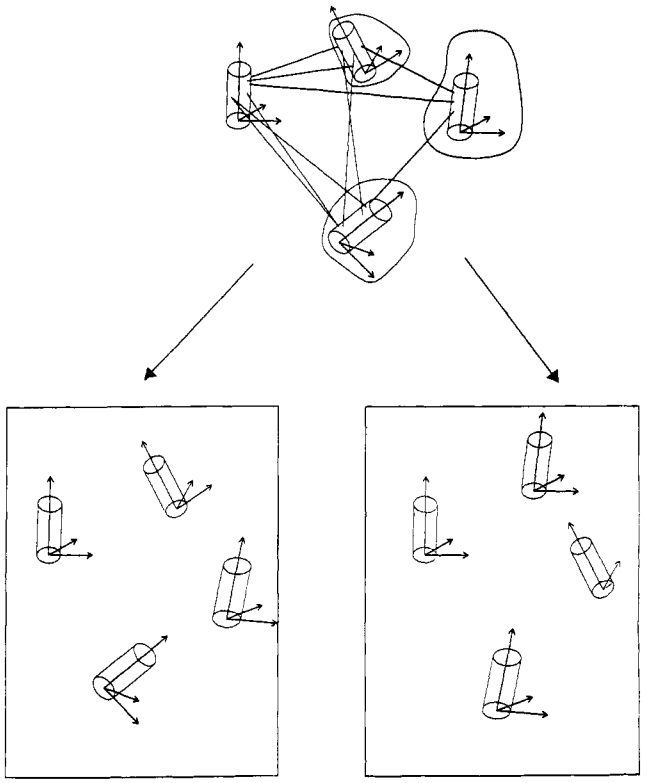

Figure 10. Secondary structure coherent instance generation. All possible combinations of locations defining the three reduced helix accessible volumes are examined for simultaneous compatibility of constraints. Each satisfying combination is a coherent instance. A representative set of these instances defines the overall topology of the molecule, and each instance constitutes a separate subproblem that may be refined further to the atomic level.

accessible volumes for the helices. They are summarized in the bottom row of Table III and in Figure 9 (bottom).

Generate and Select Secondary Structure Coherent Instances. At any time in processing it is theoretically possible to generate coherent instances of all objects in the protein. However, this procedure is generally not feasible for all objects at once, even after the application of anchor and yoke operations to initially reduce the sizes of the accessible volumes, since the accessible volumes may still be relatively large. Therefore, we selectively employ the coherent instance routine to generate coherent instances for the secondary structures alone, using the reduced accessible volumes. A representative set of these coherent instances is a description of the general topology of the molecule, without taking into account the detailed atomic positions. Each of these coherent instances then defines a separate subproblem that is refined into a detailed atomic description (Figure 10). By employing the coherent instance routine in this selective manner, we are able to use knowledge of the hierarchy of protein structure to obtain a representative set of structures in a computationally feasible manner.

For cytochrome *b*562 the total number of remaining possible helix coherent instances was $254 \times 3 \times 3 = 2286$. Since the helix accessible volumes were sampled within a 64-Å cube at 1-Å position resolution and 10° orientation resolution, the potential number of possible helix coherent instances was $[64 \times 64 \times 64 \times 36 \times 36 \times 36]^3 = 10^{29}$, a number far too large to have been tested by straightforward enumeration. By the methods outlined in the preceding steps we were able to reduce the number of possibilities to a computationally manageable 2286 without incorrectly eliminating structures. The coherent instance generator was run on these 2286 possible structures, producing 964 instances that satisfied all the constraints. Of these 964 instances 10 were selected for further processing by sampling every hundredth instance.

Generate Consistent Backbone Traces. The generation of coherent instances for the secondary structures results in a representative set of conformations for the majority of backbone peptides.

In the case of cytochrome *b*562, each conformation consisted of one six-dimensional location per helix, each of which implied a location for the helix peptides given the ideality assumption for the helices. The locations were all defined with respect to the coordinate system of helix 3.

Due to ideality assumptions and the sampling resolution (we typically sample at 1 Å), there are often small discontinuities in the positions of the backbone atoms implied by the coherent instance. It is therefore necessary to employ a local adjustment method to bring the backbone atoms into covalently continuous, chemically plausible positions and to place the intervening coils. In general, this involves small changes in the positions of atoms to accommodate these constraints. We have developed a process, called *atomic threading*, to find the continuous trace of the backbone that most closely approximates the coherent instance. Threading produces as output a peptide backbone that passes through the volume defined by the coherent instance but eliminates errors in bond length and bond angle. It has four steps: (1) Create a continuous peptide backbone of length equal to the length of the protein. (2) Set ϕ/ψ angles of regular secondary structure (helices and β-structures) to standard values. Set ϕ/ψ angles of the intervening coils to random values within the allowed Ramachandran range of an extended β-strand. (3) Superimpose the segment of peptide backbone onto the corresponding atoms within the anchor of the global coordinate system. (4) Conduct local searches of the ϕ/ψ angles of the intervening coils to minimize the squared error between the position of peptide backbone atoms and atoms within the coherent instance. This search proceeds in two directions from the fixed anchor: toward the N-terminus and toward the C-terminus (Figure 11).

Step 4 is repeated for each intervening coil segment between fixed solids of the coherent instance. For example, in Figure 11 there are four fixed helices defining the secondary structure coherent instance, of which helix 3 is taken to be the fixed anchor. Therefore, three local optimizations are performed between the fixed components of the coherent instance, in the order given by the numbers 1–3 in the figure. For each of these segments the variables of the optimization are the ϕ/ψ angles of the intervening coil segment. All other angles are held fixed. For example, for optimization 1 between the fixed location of helix 3 and that of helix 4 the variables are $\{\phi_1,\psi_1,\phi_2,\psi_2,\phi_3,\psi_3,\phi_4,\psi_4,\phi_5,\psi_5\}$, whereas for optimization 2 between the fixed location of helix 3 and that of helix 2 the variables are $\{\phi_6,\psi_6,\phi_7,\psi_7,\phi_8,\psi_8\}$.

The variables of the optimization are initially set to random values within the legal Ramachandran range of an extended β-strand. For each local optimization the target function has the form

$$\mathbf{T} = \sum_{i=k,l} d_{m_i f_i}^2$$

where m_i is an α-carbon on the moveable backbone, f_i is the corresponding fixed α-carbon of the coherent instance, $d_{m_i f_i}^2$ is the squared distance between m_i and f_i, k and l are the lower and upper sequence numbers of the α-carbons involved in the target function. For example, for optimization 1 in Figure 11 $k = 1$, $l = 6$, and for optimization 2 $k = 7$, $l = 12$.

It is important to note that this is not a global optimization using a global gradient search method, as, for example, that used by Braun and Go.[5] At each stage of threading the goal is to adjust the local dihedral angles of the amino acids between (or at the ends) of regular secondary structures such that the squared deviation between the backbone and the coherent instance α-carbons is minimized. If the precise locations of

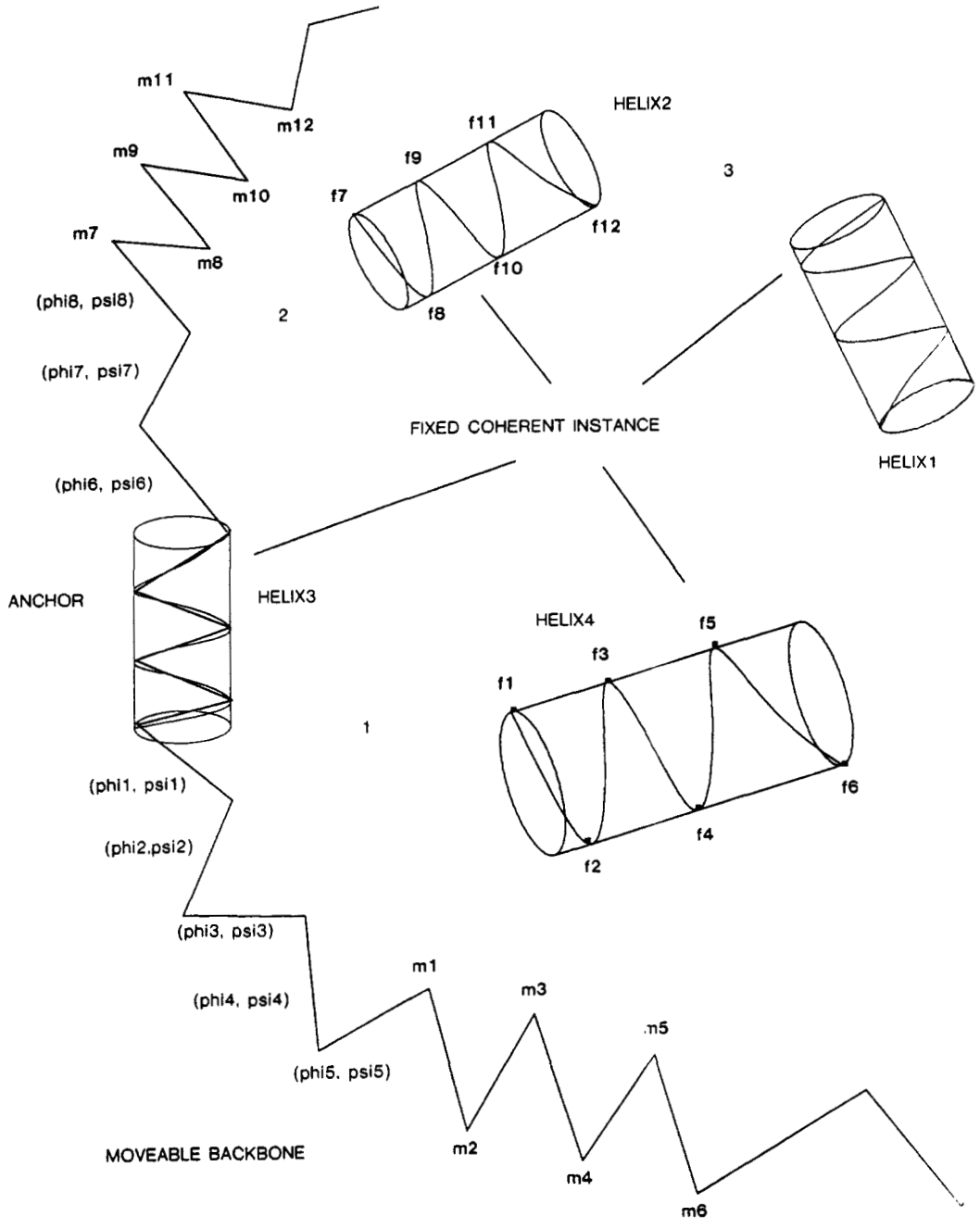

Figure 11. Threading. A movable backbone is fitted to a fixed coherent instance by a series of local optimizations. For each optimization a small number of dihedral angles are varied in order to obtain the minimum squared deviation between a set of fixed α-carbons f_i on the coherent instance and a corresponding set of movable α-carbons m_i on the backbone.

some backbone peptides are not known, we can still generate plausible (Ramachandran compatible) structures that pass through those backbone peptides that are known. Since there are always only a small number of backbone peptides that are not known (the intervening coils in the case of cytochrome *b*562), then there are always only a small number of variables in the optimization. Thus, multiple runs of the thread procedure with different random starting values for the coil dihedral angles will be able to generate a representative set of structures defining the coil accessible volumes.

The use of the coherent instance provides a strong constraint on the possible ϕ/ψ angles. In essence, the coherent instance reduces the global adjustment problem (as employed by most current methods) to a set of smaller adjustment problems, each of which involves only a small number of variables. In this situation most optimization methods, even the simple grid search utilized in our current implementation, would be expected to perform quite well.

In the case of cytochrome *b*562 a set of approximately 30 threads were performed on each of the 10 coherent instances. In this case only the helix locations were utilized and no van der Waals or coil constraints were added to the target function. Each of the threads was examined, and those that had no gross van der Waals violations were retained (this was usually only 1 or 2 of the 30 threads). If there were more than one acceptable thread, then that thread which had the smallest RMS errors between backbone and coherent instance α-carbons was taken as the best thread. Table IV shows the RMS error between backbone and coherent instance α-carbons of the best threads to the 10 coherent instances of cytochrome *b*562. These numbers are the minimum values reached by the target function for the three optimizations performed for each thread. Note that it would be relatively straightforward to augment the target function with additional terms reflecting the amount by which the coil α-carbons violated distance constraints implied by the NMR data or reflecting the amount of van der

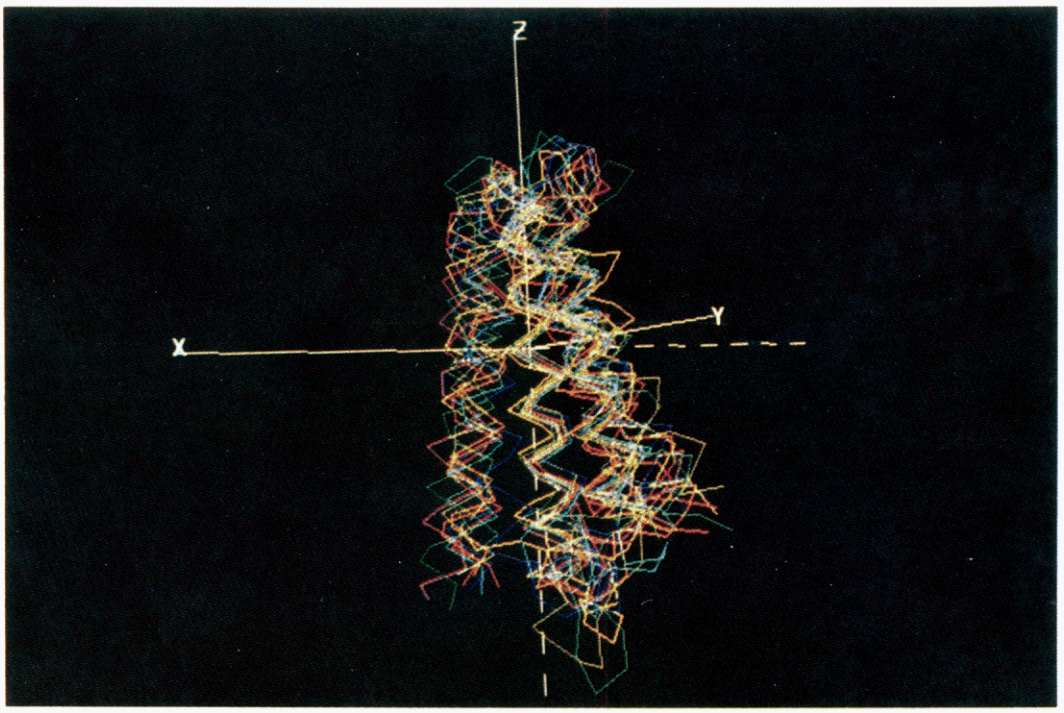

Figure 12. Best threaded backbones for 10 coherent instances, superimposed on crystal structure.

Table IV. RMS (Angstroms) between Threaded Backbone α-Carbons and Corresponding α-Carbons of Helices in Each Coherent Instance

instance	helix 1	helix 2	helix 4
1	2.29	2.68	3.87
101	1.76	1.12	1.89
201	2.88	1.18	4.86
301	3.19	1.72	1.89
401	3.25	1.07	3.49
501	2.13	1.19	1.74
601	1.22	1.69	1.97
701	1.84	1.42	1.82
801	2.44	1.86	2.62
901	2.17	0.95	1.55

Waals violation.[5] These additions are currently being implemented and should result in the complete automatization of the thread procedure, along with better placement of the coils.

Each of the 10 threaded backbones was compared with the known crystal backbone and with the other threaded backbones by the centroid method described by McLachlan.[30] The final RMS between crystal α-carbons and threaded backbone α-carbons is shown in the first column of Table V for all coherent instances. The other columns in the lower triangular part of Table V show the RMS deviations between individual coherent instances. The RMS between crystal and threaded backbone α-carbons ranged from 3.25 to 5.42. The overall range of RMS was 2.63–5.73. The upper triangular part of Table V shows the RMS deviations when only helix α-carbons were included. In this case the RMS between the threaded backbones and the crystal ranged from 2.42 to 3.01, and the overall RMS ranged from 1.37 to 3.86.

Figure 12 shows the 10 threads superimposed on the crystal structure. Note that, since constraints to the coils were not utilized, the coil accessible volumes are larger than those of the helices. However, since starting values for the small number of coil dihedral angles were randomly chosen, the superimposed threads are a reasonable depiction of the accessible volumes of the coil. The figure shows that, even without the coil NOE constraints, the coils cannot in fact be anywhere. This is because the local optimization employed by the thread procedure implicitly uses covalent constraints implied by the connectivity of the backbone and the fixed ends of the helices, and the manual selection of the best threads adds van der Waals constraints.

SYSTEM ARCHITECTURE AND CONTROL

The current software and hardware components of PROTEAN are shown in Figure 13. The hardware consists of a Silicon Graphics Iris 3020 graphics workstation, a Digital Equipment Corp. Vax 11/750 for saving information (a fileserver), and several Xerox 1100 series Lisp workstations. The hardware

Table V. RMS Deviations between Threaded Backbones (Angstroms)[a]

	xtal	1	101	201	301	401	501	601	701	801	901
xtal	0.0	2.70	2.42	3.74	3.01	2.95	2.54	2.54	2.70	2.80	2.58
1	3.25	0.00	2.53	3.78	2.45	2.62	2.75	2.93	2.83	2.61	3.21
101	4.14	4.22	0.00	3.01	2.05	2.64	1.37	1.69	2.05	2.16	1.86
201	5.42	5.16	4.45	0.00	3.09	3.86	2.77	3.54	3.06	3.52	2.94
301	4.25	3.97	3.02	4.03	0.00	2.51	1.55	2.27	1.97	1.81	2.50
401	3.74	3.45	4.07	5.33	4.34	0.00	2.84	2.54	2.21	2.85	3.11
501	3.51	4.24	2.95	4.22	2.80	3.94	0.00	1.57	1.88	1.79	1.87
601	3.34	3.92	2.92	5.31	3.83	3.56	3.04	0.00	1.77	2.18	2.32
701	4.10	4.15	2.85	3.94	2.63	3.89	3.09	3.03	0.00	2.45	2.13
801	4.42	4.58	3.14	4.66	3.06	4.63	3.22	3.69	3.45	0.00	2.21
901	5.11	5.75	3.60	4.66	3.53	5.73	3.75	4.64	3.83	3.07	0.00

[a] Lower triangular part is all α-carbon atoms. Upper triangular part is helix α-carbons only. Xtal is crystal backbone.

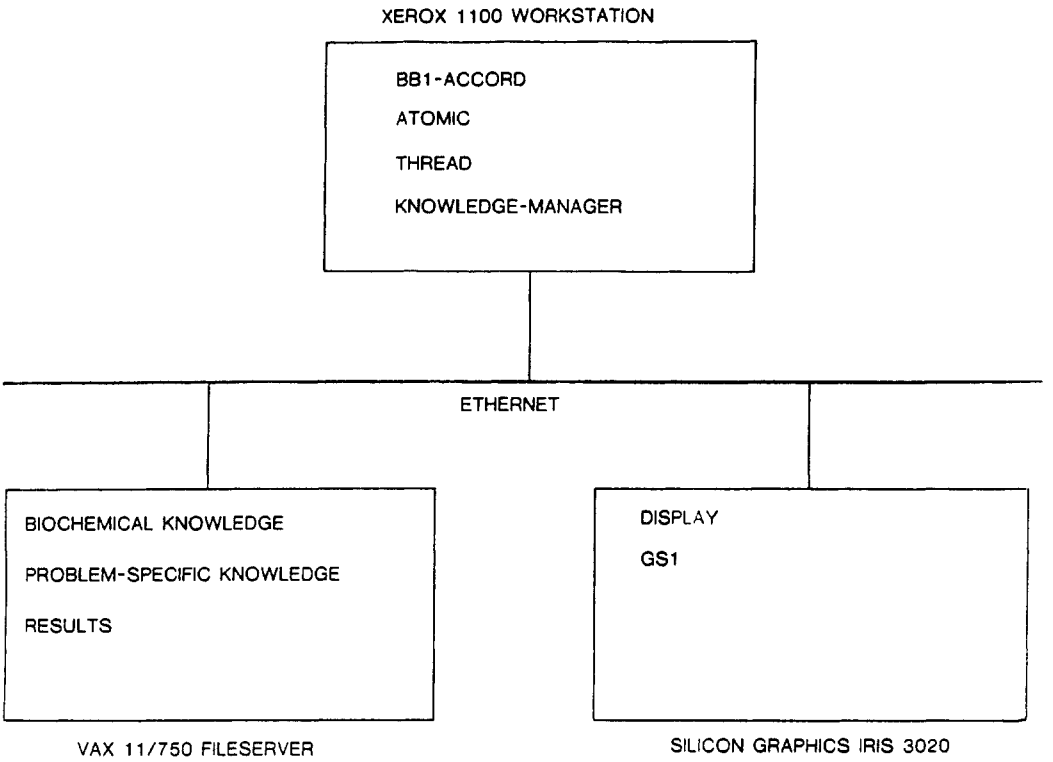

Figure 13. System architecture. PROTEAN currently comprises a set of relatively independent software modules that communicate via files stored on a Vax 11/750 fileserver. The files are grouped together into static biochemical knowledge, problem-specific knowledge for a particular protein, and results. The program modules run mainly on one or more LISP-based Xerox artificial intelligence workstations that are connected to other machines by an Ethernet local area network. Two server modules, a geometry system (GS1) and a display program, perform computationally intensive numerical calculations and display the results. The LISP modules include a knowledge manager for managing the file knowledge base, modules for performing atomic and thread procedures, and a control module called BB1-ACCORD.

is interconnected via an Ethernet local area network. Other machines on the extensive Stanford network are also occasionally used.

The software modules communicate their results via ASCII files, stored on the fileserver, the local Iris disk, and other machines on the network. Each file is an attribute–value list describing some object known by the system, and the name of the file is the name of the object. The value of an attribute in a particular file may be the name of another file, thereby creating symbolic links and allowing the formation of a distributed frame-based semantic network.[24] Related file objects are stored in the same directory on a given machine, thereby forming a particular knowledge base. For PROTEAN the major knowledge bases include static biochemical facts about proteins, problem-specific file objects representing physical parts of the protein and sets of constraints, and results file objects containing locations defining the accessible volumes of the physical objects. A software module called the "knowledge manager" accesses the file objects and contains routines for finding, editing, and accessing the attribute values of file objects. The manager maintains an internal cache: if the file object is not found in the cache, it is searched for in a user-changeable machine-pathname list.

The advantage of this file-based approach is that it allows the construction of large, distributed knowledge bases, each component of which may be maintained by one or two people. By means of the knowledge manager any particular software module running on any particular machine has access to the entire set of files but only needs to load the files it needs directly, and the loading is done in a manner that is transparent to the user.

The software modules that use these file objects were developed more or less independently by several people. At the current time they are not integrated into an overall system that allows the problem solving to be controlled automatically. Instead, each module is run manually, and the results are communicated via the file objects. However, this distributed approach to software development has proven to be a very useful mechanism for developing alternative approaches to solving this complex problem. The results in this paper were generated over a period of months as the modules were being developed (although none of the actual computations took more than one or two days on a Xerox 1108, which is about 10 times slower than a Vax 11/780).

The Geometry System GS1 is written in C and carries out all the geometric operations. It is a server that is called over the network by the other modules. The DISPLAY program accepts files produced by GS1 and displays them on the Iris. The atomic and thread modules (written in Interlisp) perform side-chain manipulations and backbone threading, respectively.

The module called BB1-ACCORD is used to control some of the problem solving at the solid level (step 5 of the strategy). This module runs in an artificial intelligence framework called BB1, which is an example of a *blackboard system*.[31] Previous reports of PROTEAN have described several components of this module.[16] The main features include a global database for storing intermediate stages in the problem solving, relatively independent subprograms for accomplishing various tasks, and a mechanism for heuristically choosing which action to perform next. Experiments have been performed with this control module to determine the cost of computing the best action to take next.[32,33] We are currently investigating the possibility of extending the ideas in BB1 in order to more fully automate the system.

DISCUSSION

The RMS deviations between the crystal structure and the threaded backbones shown in Table V are on the order of those found with other methods and reflect the current stage of

development of PROTEAN. The fit of helix α-carbons shows that we are able to place secondary structures very well, while the fit of all α-carbons shows that the major contribution to any imprecision is due to coils. Further improvements in placing coils will occur when additional van der Waals and coil distance constraints are added to the thread target function in step 8. Since most proteins, especially larger ones, have a great deal of secondary structure and relatively short coils,[34] our current ability to place secondary structure will be very useful as a constraint on the placement of coils. As in the thread procedure, any of the current adjustment methods can be used to more precisely place the coils, given the locations of the secondary structures, since the number of variables will always be relatively small. Similarly, since the final threaded structures are already nearly correct, adjustment methods can also be used as a postprocessing step to eliminate any errors caused by sampling and by assumptions such as that of ideal secondary structure.

Because of the systematic nature of the sampling method, the backbones provide an accurate (if not yet completely precise) representation of the entire set of solutions that are consistent with the input constraints, and there are no other structures implied by the input constraints. Thus, there may be fewer structures, but there are definitely not more. The data were generated from the crystal structure, but the crystal structure is not the only structure compatible with the data. Our data were all distances between protons that were less than 4 Å—the full set of NOE measurements that could be expected under ideal circumstances. Therefore, only 729 interatomic distances were used of a total of 23 111 possible interatomic distances—in the process of selecting a subset of distances, we lost the information required to completely reconstruct the crystal structure. In cases for which the NMR data are even less constraining, the ability to represent a complete set of possible structures is valuable. We have characterized the behavior of PROTEAN with data sets of different quality for the protein myoglobin.[19] In all cases, the system is accurate (accessible volumes contain the crystal structure) and the precision (size of accessible volume) varies with the size of the data set.

The result of PROTEAN is a representative family of structures that are compatible with the data. This set is an approximation to the $3N$-dimensional probability density function of the molecule assuming uniform likelihood for all constraints. Similar families can be generated by adjustment methods, but only by changing the starting structure of the optimization procedure. Since the method of adjustment may be biased and since there are too many possible starting structures, it is unlikely (except for very small proteins) that these samples will be representative.

Adjustment methods work well when the number of variables is small or when the starting structure is near the final structure.[6] Thus, in atomic threading adjustment can be employed to determine the dihedral angles of the intervening coils since there are only a small number of variables. When the complete structure has been obtained in step 8 of the strategy, an adjustment method can be employed with all the dihedral angles as variables since the starting structure is near the final structure (as evidenced by the current RMS errors). Thus, by utilizing the exclusion paradigm to systematically generate a representative set of conformations, we generate nearly correct starting structures for further refinement by an adjustment method.

The exclusion paradigm has not been considered seriously in protein structure determination primarily because straightforward enumeration techniques are not feasible for all but the smallest molecules. However, the approach of searching a space of possible answers to a given problem is the fundamental paradigm of artificial intelligence.[35] With large, combinatorial problems such as structure determination, it is well understood that the space of possible answers is impossibly large for the simplistic method of generating each possible hypothesis and testing to see if it solves the problem. Thus, the main task is to control the search to avoid exhaustive generation. Information about plausible structures that is inferred from the data and from knowledge of the problem is the primary source of power in constraining the search.

Lederberg[36] has stressed the scientific desirability of having a generator that can produce an exhaustive list of possible answers and coupling it with information that excludes classes of answers for justifiable reasons before all the instances are generated. This strategy was used successfully in the DENDRAL project[37] and more specifically in the program for chemical structure generation, CONGEN. Our approach follows the same strategy of excluding classes of conformations only when there are good reasons and retaining the remaining family of structures as plausible alternatives to be further refined with additional experimental data or explicit assumptions.

In the case of PROTEAN the generator is the coherent instance generator, which could in theory generate and test all possible structures. By first refining the protein along the two dimensions of structure and accessible volume, we are able to eliminate entire classes of conformations without having to generate them. As in most artificial intelligence problem solving the information that allows us to do this comes from the local nature of the data, from knowledge of protein structure, and from heuristics we have learned about which possible actions are likely to exclude the most structures.

ACKNOWLEDGMENT

The PROTEAN project is an interdisciplinary collaboration between the Stanford Computer Science Department and the Stanford Medical School. We thank the following people who, in addition to the authors, have contributed to the development of PROTEAN: John Brugge, Craig Cornelius, Felix Frayman, Alan Garvey, Jeff Harvey, Barabara Hayes-Roth, Michael Hewett, Robin Holbrook, and Olivier Lichtarge.

Registry No. Cytochrome b_{562}, 9064-79-3.

REFERENCES

(1) Blundell, T. L.; Johnson, L. N. *Protein Crystallography*; Academic: New York, 1976.
(2) Jardetzky, O.; Roberts, G. C. K. *NMR in Molecular Biology*; Academic: New York, 1981.
(3) Wüthrich, K. *NMR of Proteins and Nucleic Acids*; Wiley: New York, 1986.
(4) Havel, T.; Wüthrich, K. "A distance geometry program for determining the structures of small proteins and other macromolecules from nuclear magnetic resonance measurements of intramolecular H-H proximities in solution". *Bull. Math. Biol.* **1984**, *46*, 673–698.
(5) Braun, W.; Gō, N. "Calculation of protein conformations by proton–proton distance constraints: A new efficient algorithm". *J. Mol. Biol.* **1985**, *186*, 611–626.
(6) Frayman, F. "PROTO: An approach for determining protein structures from nuclear magnetic resonance data: An exercise in large scale interdependent constraint satisfaction". Ph.D. Dissertation, Northwestern University, August 1985.
(7) Karplus, M.; McCammon, J. A. "Dynamics of proteins: elements and function". *Annu. Rev. Biochem.* **1983**, *52*, 263–300.
(8) Levitt, M. "Molecular dynamics of native protein I. Computer simulation of trajectories". *J. Mol. Biol.* **1983**, *168*, 595–620.
(9) Clore, G.; Brünger, A.; Karplus, M.; Gronenborn, A. "Application of molecular dynamics with interproton distance restraints to three-dimensional protein structure determination: A model study of crambin". *J. Mol. Biol.* **1986**, *191*, 523–551.
(10) Gariepy, J.; Lane, A.; Frayman, F.; Wilbur, D.; Robien, W.; Schoolnik, G.; Jardetzky, O. "Structure of the toxic domain of the *Escherichia coli* heat-stable enterotoxin STI". *Biochemistry* **1986**, *25*, 7854–7886.
(11) Lefèvre, J.-F.; Lane, A. N.; Jardetzky, O. "Solution structure of the Trp operator of *Escherichia coli* determined by NMR". *Biochemistry* **1987**, *26*, 5076–5090.
(12) Altman, R.; Jardetzky, O. "New strategies for the determination of macromolecular structure in solution". *J. Biochem. (Tokyo)* **1986**, *100*, 1403–1423.

(13) Brinkley, J.; Cornelius, C.; Altman, R.; Hayes-Roth, B.; Lichtarge, O.; Duncan, B.; Buchanan, B.; Jardetzky, O. "Application of constraint satisfaction techniques to the determination of protein tertiary structure". Technical Report KSL-86-28, Stanford University, March 1986.
(14) Buchanan, B.; Hayes-Roth, B.; Lichtarge, O.; Altman, R.; Brinkley, J.; Hewett, M.; Cornelius, C.; Duncan, B.; Jardetzky, O. "The heuristic refinement method for deriving solution structures of proteins". Technical Report KSL-85-41, Stanford University, 1985.
(15) Duncan, B.; Buchanan, B.; Lichtarge, O.; Altman, R.; Brinkley, J.; Hewett, M.; Cornelius, C.; Jardetzky, O. "PROTEAN: A new method for deriving solution structures of proteins". *Bull. Magn. Res.* **1987**, *8*, 111–119.
(16) Hayes-Roth, B.; Buchanan, B.; Lichtarge, O.; Hewett, M.; Altman, R.; Brinkley, J.; Cornelius, C.; Duncan, B.; Jardetzky, O. "PROTEAN: Deriving protein structure from constraints". *Proceedings of the Fifth National Conference on Artificial Intelligence*; American Association for Artificial Intelligence; Morgan Kaufman Publ.: Los Altos, CA, 1986; pp 904–909. Also published as Stanford University Technical Report KSL 86-38.
(17) Jardetzky, O.; Lane, A.; Lefèvre, J.; Lichtarge, O.; Hayes-Roth, B.; Buchanan, B. "Determination of macromolecular structure and dynamics by NMR". *NMR in the Life Sciences*, Plenum, Nato Advanced Study Institute, Erice, Italy, 1985; pp 49–72.
(18) Lichtarge, O. "Structure determination of proteins in solution by NMR". Ph.D. Dissertation, Stanford, November 1986.
(19) Lichtarge, O.; Cornelius, C. W.; Buchanan, B. G.; Jardetzky, O. "Validation of the First Step of the Heuristic Refinement Method for the Derivation of Solution Structures of Proteins from NMR Data". *Proteins: Struct., Funct., Genet.* **1987**, *2*, 340–358.
(20) Wüthrich, K.; Billeter, M.; Braun, W. "Polypeptide secondary structure determination by nuclear magnetic resonance observation of short proton–proton distances". *J. Mol. Biol.* **1984**, *180*, 715–740.
(21) Brugge, J. A.; Buchanan, B. G.; Jardetzky, O. "Toward automating the process of determining polypeptide secondary structure from ^{1}H NMR data". *J. Comput. Chem.* **1988**, *9*, 662–673.
(22) Paul, R. P. *Robot Manipulators: Mathematics, Programming and Control*; Massachusetts Institute Technology: Cambridge, MA, 1981.
(23) Newman, W. M.; Sproull, R. F. *Principles of Interactive Computer Graphics*, 2nd ed.; McGraw-Hill: New York, 1979.
(24) Winston, Patrick Henry *Artificial Intelligence*, 2nd ed.; Addison-Wesley, Menlo Park, CA, 1984.
(25) Brinkley, J. F.; Buchanan, B. G.; Altman, R. B.; Duncan, B. S.; Cornelius, C. W. "A heuristic refinement method for spatial constraint satisfaction problems". Technical Report STAN-CS-87-1142, KSL-87-05, Stanford University, January 1987.
(26) Mackworth, A. K. "Consistency in Networks of Relations". *Artif. Intelligence* **1977**, *8*, 99–118.
(27) Waltz, D. "Understanding line drawings of scenes with shadows". In *The Psychology of Computer Vision*; Winston, P. H., Ed.; McGraw-Hill: New York, 1975.
(28) Lederer, F.; Glatigny, A.; Bethge, P.; Bellamy, H.; Mathews, F. "Improvement of the 2.5 angstroms resolution model of cytochromeb562 by redetermining the primary structure and using molecular graphics". *J. Mol. Biol.* **1981**, *148*, 427.
(29) Jarvis, L.; Huang, C.; Ferrin, T.; Langridge, R. *UCSF MIDAS Molecular Interactive Display and Simulation: User's Manual*, 1986.
(30) McLachlan, A. D. "Gene duplications in the structural evolution of chymotrypsin". *J. Mol. Biol.* **1979**, *128*, 49–79.
(31) Hayes-Roth, B. "A blackboard architecture for control". *Artif. Intelligence* **1985**, *26*, 251–321.
(32) Garvey, A.; Cornelius, C.; Hayes-Roth, B. "Computational costs versus benefits of control reasoning". *Proceedings, Sixth National Conference on Artificial Intelligence*, American Association for Artificial Intelligence; Morgan Kaufman Publ.: Los Altos, CA, 1987; pp 110–115.
(33) Altman, R. B.; Buchanan, B. G. "Partial compilation of strategic knowledge". *Proceedings, Sixth National Conference on Artificial Intelligence*, American Association for Artificial Intelligence; Morgan Kaufman Publ.: Los Altos, CA, 1987; pp 399–404.
(34) Richardson, J. S. "The anatomy and taxonomy of protein structure". *Adv. Protein Chem.* **1981**, *34*, 167–339.
(35) Simon, H. A. *The Sciences of the Artificial*, Massachusetts Institute of Technology: Cambridge, MA, 1969.
(36) Lederberg, J. "Applications of Artificial Intelligence for Chemical Inference I: The Number of Possible Organic Compounds: Acyclic Structures Containing C, H, O, and N". *J. Am. Chem. Soc.* **1969**, *91*, 2973.
(37) Lindsay, R. K.; Buchanan, B. G.; Feigenbaum, E. A.; Lederberg, J. *Applications of Artificial Intelligence for Organic Chemistry: The DENDRAL Project*; McGraw-Hill: New York, 1980.