Optimization of van der Waals Energy for Protein Side-Chain Placement and Design

Amr Fahmy and Gerhard Wagner*
Biological Chemistry and Molecular Pharmacology, Harvard Medical School, Boston, Massachusetts

ABSTRACT Computational determination of optimal side-chain conformations in protein structures has been a long-standing and challenging problem. Solving this problem is important for many applications including homology modeling, protein docking, and for placing small molecule ligands on protein-binding sites. Programs available as of this writing are very fast and reasonably accurate, as measured by deviations of side-chain dihedral angles; however, often due to multiple atomic clashes, they produce structures with high positive energies. This is problematic in applications where the energy values are important, for example when placing small molecules in docking applications; the relatively small binding energy of the small molecule is drowned by the large energy due to atomic clashes that hampers finding the lowest energy state of the docked ligand. To address this we have developed an algorithm for generating a set of side-chain conformations that is dense enough that at least one of its members would have a root mean-square deviation of no more than R Å from any possible side-chain conformation of the amino acid. We call such a set a side-chain cover set of order R for the amino acid. The size of the set is constrained by the energy of the interaction of the side chain to the backbone atoms. Then, side-chain cover sets are used to optimize the conformation of the side chains given the coordinates of the backbone of a protein. The method we use is based on a variety of dead-end elimination methods and the recently discovered dynamic programming algorithm for this problem. This was implemented in a computer program called Octopus where we use side-chain cover sets with very small values for R, such as 0.1 Å, which ensures that for each amino-acid side chain the set contains a conformation with a root mean-square deviation of, at most, R from the optimal conformation. The side-chain dihedral-angle accuracy of the program is comparable to other implementations; however, it has the important advantage that the structures produced by the program have negative energies that are very close to the energies of the crystal structure for all tested proteins.

INTRODUCTION

Side-chain conformation prediction, or protein side-chain placement, is an important problem in computational structural biology. Its accurate and efficient solution would have many applications, among them being ab initio protein structure prediction, protein design, and protein complex structure prediction—all of which are vital problems leading toward advances in understanding fundamental questions in structural biology.

The problem can be stated as the prediction of the conformation of the amino-acid side chains, given the coordinates of the protein backbone. The problem is easier than the full protein folding problem; however, it is still computationally hard in the sense that a polynomial time algorithm for side-chain placement can be transformed to solve other computationally hard problems in polynomial time, such as the "traveling-salesman problem" (1,2), for which, as of this writing, no polynomial time algorithm is known.

Despite this, and because of the importance of finding methods to solve the problem of protein side-chain placement, it has been addressed several times in the literature. Many different algorithmic techniques have been used to attempt solving the problem. Some methods are exact, based on the dead-end-elimination (DEE) algorithms (3–6) and on graph theory methods (7,8). Other methods are approximate, such as simulated annealing and Monte Carlo methods (9). Most of the above methods use rotamer libraries to model amino-acid side-chain conformations. The idea is to limit the number of side-chain conformations by gathering conformations that have already appeared in experimentally determined protein structures (see, for example, (10–12)).

Starting with the assignments of all possible rotamers to each residue of the protein from a rotamer library, the program TreePack (http://ttic.uchicago.edu/~jinbo/TreePack.htm) (8) first performs the DEE algorithms by removing rotamers that cannot be part of the optimized energy state. After this step, a small number of rotamers per residue will be left. To optimize the energy of the side-chain conformations, a dynamic programming (13)-based algorithm has been discovered whose runtime is short enough for handling many interesting proteins; in a few seconds of computer time, it is able to select from the remaining set a single rotamer for each residue that will minimize the protein energy. Based on the work of Xu and Berger (8), Krivov et al. (14) developed SCWRL4 (http://dunbrack.fccc.edu/scwrl4/), which also uses the dynamic programming method for side-chain placement and the rotamer library developed by Dunbrack and Karplus (11).

The tree decomposition algorithm used in these programs is certainly correct, which can be proven in that they indeed identify the assignment that minimizes the energy within the

Submitted June 8, 2011, and accepted for publication July 28, 2011.

*Correspondence: gerhard_wagner@hms.harvard.edu

Editor: Axel T. Brunger.

0006-3495/11/10/1690/9 $2.00

doi: 10.1016/j.bpj.2011.07.052

set of conformers used. Thus, we hypothesize that the large positive energies of the resulting structures (see Table 1) are due to the limited choices of rotameric states found in the rotamer libraries and, due to the use of an artificial energy function (15), a coarse approximation of the Lennard-Jones potential.

Here we test this hypothesis by using the dynamic programming algorithm while generating, on the fly and at any desired resolution, all possible side-chain conformations for all residues of a protein, and by using the exact van der Waals energy function. The side-chain conformations are generated by uniformly rotating all dihedral angles, starting from any initial side-chain state, by small-enough angular increments so as to increase the number of available side-chain conformations, thus avoiding atomic clashes, and guaranteeing that a conformation close to the optimal conformation is included in the generated set. The effect of increasing the resolution of rotamer libraries on the energy of the protein is discussed in Xiang and Honig (16). The side-chain states that are generated will be used in the dynamic programming algorithm to predict the conformation of the side chains in proteins. The approach presented here creates near ideally placed side chains without clashes. It minimizes the side-chain energies with a correct van der Waals energy and will allow flexible docking of small molecule ligands or other proteins.

THEORY AND METHODS

We first provide an overview of the method followed by an explicit description. The basis of the method is to generate a set of side-chain conformations that we term side-chain cover sets (SCCS), each set of which is dense enough so that at least one of its members would be close to the conformation in the native state of the protein. The generation method is exhaustive up to a resolution R, which controls the dihedral angle increment in a manner that will be discussed shortly. The smaller the value of R, the more the number of valid side-chain conformations is increased, thus making available many more choices to the optimization algorithm. Next, we discuss the energy-resolution step and the DEE methods used in our implementation. The dynamic programming energy optimization algorithm is summarized after that.

TABLE 1 vdW side-chain energy of several proteins from the PDB compared to the energy of the predicted structures of TreePack and SCWRL4

PDB structure			TreePack	SWRL4
		vdW energy	vdW energy	vdW energy
PDB ID	Residues	Kcal/mol	Kcal/mol	Kcal/mol
16pk.pdb	415	−658.3	$>10^6$	109,087.6
1a8d.pdb	452	−890.2	$>10^8$	45,003.1
1a8i.pdb	812	−1650.6	$>10^5$	485,521.7
1b6a.pdb	355	−656.9	$>10^3$	10,680.5
1bfg.pdb	126	−236.8	$>10^{12}$	−103.9
1bg6.pdb	349	−506.4	$>10^8$	−364.3

Value of the vdW energy is often very large due to atomic clashes as compared to the energy of the structure found in the PDB.

Side-chain cover sets

Rotamer libraries store amino-acid side-chain conformations based on experimentally determined protein structures. There is small variation in bond-length and angle-bending values of the side chains—the major variables that determine the conformation of an amino-acid side chain are the dihedral angles.

In this section, we explain how to generate a side-chain conformation set that is sufficiently dense that the root mean-square deviation (RMSD) to an arbitrary side-chain conformation is, at most, R Å for any given R. The generated set will be called a side-chain cover set of order R (SCCS-R). For example, we will show that for R=1 Å—given an arbitrary side-chain conformation t (not necessarily a member of the SCCS)—the set will contain a member s such that the RMSD between s and t is, at most, 1 Å.

Definition

Let t be an arbitrary side-chain conformation of some amino acid and let S be a set of side-chain conformations of the amino acid. S is called a side-chain cover set of order R for the amino acid if there exists at least one member, $s \in S$, such that the RMSD between s and t is at most R.

The SCCS-R provides a better alternative than using rotamer libraries alone because of several reasons. First, the SCCS-R is complete because it uniformly covers the entire 2π range of each of the dihedral angles of the side chain. This is contrasted with rotamer libraries where some valid conformations may not have appeared in any experimentally determined protein structure. A second reason is that the resolution, R, can always be made smaller for any one amino acid or individual residue, to obtain more conformations that will allow for more accommodation among the atoms of the side chains of the protein so as to minimize its energy. Rotamer libraries are fixed and it is inevitable that side-chain placement programs perform a search near the stored rotamers to minimize energy (16). Finally, the resolution R is known a-priori, which allows more research into other factors that contribute to the accuracy of the solutions to the problem of side-chain placement.

For reasonable values of R, the size of the SCSS will be large for use in the optimization algorithm. After the generation of a SCCS, conformations with energies possessing too high a backbone are deleted from further consideration.

Generation of side-chain cover sets

In the problem we are addressing, the backbone will be held fixed and the dihedral angles of the side chain will be rotated. Within this picture, dihedral angles are nested, meaning that rotating about the first dihedral bond rotates the entire side chain and fixes only the first group. Rotating about the second dihedral bond leaves the first group unchanged, then rotates the rest of the side chain and fixes the second group, and so on.

To generate a SCCS-R, for each increment of each dihedral, the 2π range of all of the remaining dihedral bonds will be exhausted. The initial state of the side chain used in the generation algorithm is not important because we are going to rotate all dihedral angles over the 2π range. We can arbitrarily choose the initial state to be the state with all dihedral angles set at 0°. As an example of the generation algorithm presented here, for a side chain with two dihedrals, the first dihedral will be rotated once by its increment and the second dihedral will be rotated by its increment over its entire 2π range, then the first dihedral will be rotated by another increment, etc. This process continues until the 2π range of the first dihedral has been exhausted.

In Appendix A, we give the details of deriving the angular increments for side chains of up to four dihedral angles that are necessary to meet the required RMSD R. Table 2 shows sizes of SCCS-R, for various values of R for all amino acids with rotatable bonds. These values were computed when the amino acid is isolated and are smaller in the presence of a protein backbone. The table also gives the average smallest RMSD of 1000 random conformations of each amino acid to the SCCS-R for various values of R.

TABLE 2 Size of unrestricted side-chain cover set of order *R* for each of the amino acids with rotatable bonds

	$R = 4.0$		$R = 2.0$		$R = 1.0$		$R = 0.5$	
AA	Set size	RMSD	Set size	RMSD	Set size	RMSD	Set size	RMSD
Cys	2	1.490	3	0.852	6	0.479	11	0.238
Thr	2	1.599	3	0.896	5	0.486	9	0.259
Ser	2	1.572	3	0.891	5	0.477	9	0.250
Val	2	1.522	3	0.837	5	0.466	9	0.255
His	7	1.758	18	0.969	61	0.505	208	0.250
Phe	7	1.883	19	1.033	61	0.537	212	0.277
Tyr	7	1.818	22	1.053	71	0.528	256	0.268
Trp	7	2.141	26	1.179	95	0.615	382	0.315
Asn	6	1.765	10	0.923	18	0.322	68	0.157
Asp	6	1.885	10	0.965	14	0.381	48	0.181
Ile	3	1.393	10	0.718	35	0.374	145	0.184
Leu	5	1.717	13	0.971	47	0.523	166	0.255
Met	10	1.127	58	0.606	422	0.272	2956	0.161
Gln	9	0.823	40	0.403	231	0.208	1312	0.103
Glu	9	0.826	40	0.404	221	0.208	1237	0.104
Lys	21	1.486	237	1.773	2786	0.516	32,963	0.230
Arg	21	2.049	240	1.327	3968	0.551	53,337	0.138

To test the covering property of each set, 1000 random side chains for each amino acid were generated and the closest member of the SCCS was identified. The average RMSD between the random side chains and the closest member of the SCSS is listed. In each case the RMSD is lower than the *R* value, showing that the SCCS covers all side chains for the given *R*. The size of the SCCS becomes much smaller once torsion energy and energy to their backbones are taken into account.

These values are smaller than *R* for each case, which shows that the set covers all the randomly generated side chains. As the number of dihedral bonds increases, the difference between *R* and the smallest RMSD to the random conformations becomes larger. This is due to the accumulated effect of computing upper bounds on the atomic deviations.

Steric hindrance does not permit all possible combinations of the dihedral angles. However, there are occasions in which strained side-chain conformations have been found in experimentally determined protein structures (17). We have restrained the size of the SCCS by the value of the self-energy of the side chain but we also allow strained conformations to be part of the set. The high energy of the strained conformation may be compensated for by low energy of the surrounding structure. Discarding or keeping strained side-chain conformations becomes part of the energy optimization problem.

Protein side-chain optimization algorithm

The optimization algorithm used here is based on the dynamic programming algorithm in Xu and Berger (8) over the tree decomposition of the residue interaction graph of the protein. The major difference is that we generate the SCCS for each residue in the protein instead of using rotamer libraries. An outline of the steps of our algorithm is:

1. For each residue, generate the SCCS-*R* for the residue.
2. For each residue, compute the energy of all side-chain conformations to each of the neighboring residues.
3. For each residue, perform the energy-resolution step.
4. Perform the DEE step.
5. Build the residue interaction graph.
6. Compute the tree-decomposition of the graph.
7. Traverse the tree from the leaves to the root to compute the value of the minimum energy.
8. Traverse the tree from the root to the leaves to extract the minimum energy side-chain assignment.

After these steps are carried out, each residue will have one side-chain conformation left that minimizes the energy of the protein up to *R*.

A precise energy function

The energy function that is optimized in the programs TreePack and SCWRL (7,14,15) is an approximation of the repulsive component of the Lennard-Jones potential. It is designed so that it does not have very large positive values when atoms clash. Furthermore, atomic radii of terminal atoms in longer residues are reduced so as "to make the steric term more forgiving…" (Bower et al. (15)). In studying energy functions for protein side-chain placement and design, Pokala and Handel also used a linearized version of the Lennard-Jones potential function "to relieve clashes due to the use of discrete rotamers" (18).

Here we optimize the exact Lennard-Jones potential of the van der Waal's energy (19). The Lennard-Jones potential between a pair of atoms *a* and *b* is given by

$$E(a,b) = \frac{A}{r^{12}} - \frac{B}{r^{6}},$$

where *r* is the distance between the atoms and where *A* and *B* are constants that depend on the atomic species.

The self-energy, $E_{self}(s_i)$, of side-chain conformation s_i, is the energy between atoms of the side chain, and the backbone atoms of the protein including those of the side chain itself.

The energy of a side-chain assignment *P* (i.e., a choice of a side-chain conformation for each residue of the protein) can be written as a sum of pairs of interacting side chains. The energy, *E*(*P*), of a side-chain assignment *P* for the protein can thus be written as

$$E(P) = \sum_{i=1}^{N} E_{self}(s_i) + \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} E_{pair}(s_i, s_j),$$

where s_i is the side-chain conformation of residue *i*, and $E_{pair}(s_i,s_j)$ is the energy of side chains s_i and s_j. The aim of this work is to assign a conformation for each residue so that *E*(*P*) is minimized. Parameters for the Lennard-Jones potential and the torsion angle potential are the same ones used in the program CHARMM (19).

Side-chain energy-resolution method

Consider an arginine or a lysine on the surface of the protein with a small number of neighboring residues. The algorithm for generating the side-chain conformations for such residues will produce many conformations that are very close to each other, yet their energy to the rest of the structure is the same or very close. Here we resolve the energy of these conformations by calculating the RMSD between side-chain conformations of the residue as well as their minimum energy to the rest of the structure. Any pair of conformations within 0.1 Å and energy difference of <0.1 Kcal/mol are merged into a single conformation. This method has minimal impact on the global minimum energy conformation yet reduces the number of side-chain conformations whose energies are indistinguishable. This energy resolution step is carried out for each residue and its computational time is very small.

Dead-end elimination step

After generating and energy-resolving the side-chain conformations for each residue, we use the DEE method to remove side-chain conformations that cannot be part of the minimum energy state. The first method we use is the Goldstein criteria (4), which states that, for residue *i*, a conformation s_i can be eliminated if there is another conformation, r_i, such that

$$E_{self}(s_i) - E_{self}(r_i) + \sum_{j=1, j\neq i}^{N} \min_{r_j}\{E_{pair}(s_i, r_j) - E_{pair}(r_i, r_j)\} > 0.$$

This states that regardless of the choice for all other residues, conformation r_i can eliminate s_i if choosing r_i always results in lower energy. This criterion is very powerful and results in a substantial reduction in the number of possible conformations for each residue.

For small values of R, the number of generated side-chain conformations at each residue may be large and in that case, we use additional DEE criteria. We follow the methods described in Pierce et al. (5) and Gordon and Mayo (6), which are efficient and effective. Suppose that residue k has a number of possible conformations that cause ambiguity for eliminating conformations at some other residue i. The simple-split method partitions the conformational space of the protein using the conformations of residue k. Now, within each partition, if conformation s_i is now eliminated, s_i can be eliminated from position i because it can be eliminated for all possible conformations of residue k.

The Goldstein criteria can also be applied to pairs of rotamers of neighboring residues. The energies of dead-ending pairs will not be used in the subsequent Goldstein and simple-split DEE steps, as this results in more singles eliminations. However, computation time for full doubles elimination is high and the magic-bullet method (6) is used to identify dead-ending pairs in an efficient manner. The magic-bullet pair of rotamers is the pair that minimizes the maximum energy between rotamers of the residues. We also use the split-magic-bullet method (5) in which the conformational space is split by pairs of rotamers (only the magic-bullet pairs), and in each partition, if rotamer s_i can be eliminated by some other rotamer of residue i, it is eliminated from the set of possibilities for residue i. The DEE schedule (5) that we use can be summarized as follows:

1. Repeat the following steps:
2. Goldstein singles until no further eliminations;
3. simple-split until no further eliminations;
4. magic-bullet-doubles to mark dead-ending pairs; and
5. split-magic-bullet,
6. until average number of conformations per residue ≤ 2.

Residue interaction graph and its tree decomposition

To model the dependence of residues on each other in the structure of a protein, Canutescu et al. (7) introduced the idea of building a graph. Each node in the graph represents a residue of the protein. There is an edge between nodes i and j in the graph if the interaction energy between the residues in the protein is significant.

In their program, called SCWRL3.0, Canutescu et al. (7) decompose the residue interaction graph into biconnected components where any two biconnected components share, at most, one residue. For each possible rotamer assignment to the shared residue, each biconnected component is optimized and the global minimum assignment is computed. The algorithm is deterministic and will identify the minimum energy state given the rotamer library. SCWRL3.0 can take a long time for large proteins and for proteins where the biconnected components are large. This motivated Xu and Berger (8) to discover a new algorithm for the problem, which uses the tree-decomposition of the residue interaction graph in a program called TreePack. TreePack is up to five-times faster than SCWRL3.0. Later these methods were used in SCWRL4.0. (Here, we give only an outline of these methods; for a more complete discussion, including proof of correctness of the method, see Xu and Berger (8).)

The tree decomposition of a graph was introduced by Robertson and Seymour (20). It is a data structure that is used to compute solutions of instances of many NP-hard problems if the graph used to model the problem is sparse. This is the case with the residue interaction graph of a protein because the energy between pairs of residues beyond a certain distance is very small. The tree decomposition of the residue interaction graph describes the interdependence of subsets of residues in the protein in the form of a tree. The tree is important because subsets of residues that are represented by different children of a tree-node can be optimized independently given an enumeration of the side-chain conformations of residues of their parent node. See Appendix B for the definition of a tree decomposition.

Dynamic programming equation

Given a tree decomposition of the residue interaction graph and a set of side-chain conformations for each residue, Xu and Berger (8) describe a procedure to compute the assignment of a side-chain conformation to each residue that will minimize the energy of the structure. It is the tree data structure that will allow fast calculation of the minimum energy. We follow the notation and methods given in Xu and Berger (8) for the minimization algorithm.

The calculation of the value of the minimum energy starts at the leaves of the tree and proceeds upwards. Computation of the energy at a node can start only when the energy has been computed for all of its children. At the root of the tree, all possible assignments to residues in the root node are enumerated and the energy of each of its children is added. The energy at the root node is the energy of the entire protein.

In the top-down stage, starting at the root node, the assignments to the residues at the root node that minimizes the energy can be found. Recursively, this assignment is passed down to the children and the assignments to the residues at the children that minimize the energy can also be found. See Appendix C for the dynamic programming equation.

RESULTS

Generation of SCCS-R by using the DEE methods mentioned above, as well as the dynamic programming optimization algorithm, were all implemented in a C program called Octopus. The program was tested on 180 high-resolution protein crystal structures from the Protein DataBank (PDB); this is the same test set as used in Xu and Berger (8). In all test cases, side chains of all amino acids were deleted and the backbones were given to the program to place the side chains. Although the generation accuracy for residues is adjustable in Octopus, the results of the test cases were generated with the same set of R values for all proteins ($R = 0.1$ for all amino acids with a single dihedral; $R = 0.3$ for all amino acids with two dihedrals; $R = 0.6$ for all amino acids with three dihedrals; and $R = 1.0$ for arginine and lysine). Using larger values of R resulted in clashes for some test cases.

The main results of this work are:

1. The program Octopus consistently produces structures with negative energy in contrast to other methods. The energies of structures produced by Octopus were less than the energies of structures produced by TreePack and SCWRL4 for all test cases.
2. Octopus produces structures with energies nearly identical to crystal structures (see Fig. 1). The correlation coefficient between the energies of the tested proteins and the predicted structures is 0.92.

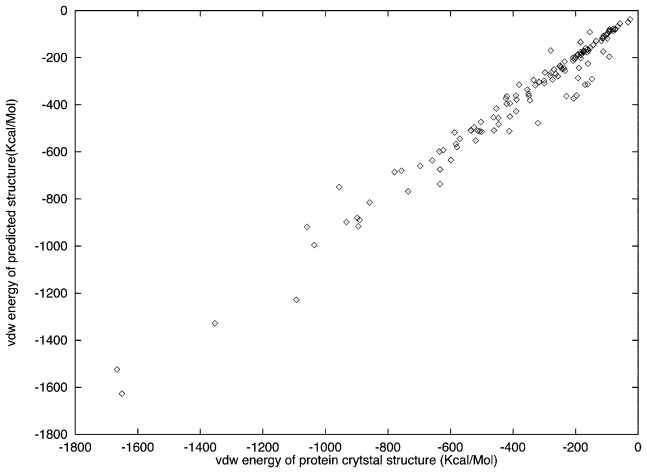

FIGURE 1 Plot of vdW energies of side-chain placement obtained with the program Octopus against those calculated for the experimental crystal structures. To measure the performance of the program, for each tested protein crystal structure, the energy of the protein is measured (*horizontal axis*) and the side chains are removed. The side chains are then replaced using the program Octopus and the energy is measured again (*vertical axis*). The correlation coefficient is 0.92. The resolution R for the SCCS was as described in the text, $R = 0.1$ for all amino acids with a single dihedral, $R = 0.3$ for all amino acids with two dihedrals, $R = 0.6$ for all amino acids with three dihedrals, and $R = 1.0$ for arginine and lysine.

3. Octopus produces structures with correct side-chain dihedral angles similar to other methods (see Fig. 2, *A* and *B*). Historically, performance of side-chain placement programs has been measured by the angular deviation of the side-chain dihedral angles from those found in the PDB structures. Energy of predicted structure as compared to the experimentally determined structure was not used due to the atomic clashes that causes the van der Waals energy to have very large values.
4. Octopus produces structures without side-chain clashes. As an example, Fig. 3 shows the location of TRP-33 in the PDB structure 1BG6 as predicted by the program TreePack on the right with the side chain clashing with many of its neighbors including the protein backbone. The prediction made by Octopus on the left is nearly identical to the crystal structure with no atomic clashes.
5. In general, Octopus is slower than both TreePack and SCWRL4. The program Octopus takes ~1–2 min of computer time on a 3-GHz Linux workstation for a 500-residue protein. TreePack takes ~0.5 s whereas SCWRL4 takes ~3 s for the same protein and same workstation. This is due to the large number of side-chain conformations that are considered by Octopus.

DISCUSSION

To our knowledge, we have developed a new approach for side-chain placement in proteins with a given backbone structure. This problem has been tackled by a number of groups who developed tools to advance this issue, such as dead-end-elimination, graph theory methods, simulated annealing, and Monte Carlo methods. Unfortunately, these sophisticated approaches produce structures with high energies and numerous side-chain clashes. We argue that the reason for these difficulties is the use of rotamer libraries upon which most of these approaches rely. Rotamer libraries create somewhat artificial boundaries for side-chain conformations and have a predefined resolution that cannot easily be modified to eliminate clashes.

Thus, we decided to use a new, to our knowledge, concept of side-chain cover sets that contain all possible side-chain conformations within a resolution R. Importantly, R can be made as low as desired during the process of side-chain placement. We also utilized the sophisticated tools of DEE and graph theory as well as similar methods developed by others.

Being able to decrease the value of R in particular cases turned out to be crucial for eliminating clashes. In a first attempt to compare energies of crystal structures with those obtained with the Octopus program as shown in Fig. 1, we found a few drastic outliers when working with a suboptimal resolution; however, lowering the resolution resulted in perfect agreement of energies between the selected PDB entries and the structures calculated with Octopus.

Overall, the approach presented here can place side chains correctly within predefined backbone structures, with low negative energies, nearly identical to the experimental crystal structures. To our knowledge, this is not achievable with other available approaches, such as TreePack or SCWRL4.0.

The approach presented here and the implementation in the Octopus program has numerous potential applications, as enumerated here:

1. We anticipate that it will become an invaluable tool in homology modeling for replacing nonconserved amino acids.
2. The Octopus program will be able to place side-chain conformations in NMR structures of large proteins, and in particular of membrane proteins that have been determined from backbone constraints only, such as residual dipolar couplings, paramagnetic relaxation enhancements, and a few nuclear Overhauser effect (NOE) restraints. An example is the recently determined structure of the mitochondrial uncoupling protein UCP2 (21).
3. This approach will also be suitable for placing side chains correctly in other NMR structures that are based on a restricted number of NOEs, such as between the methyl groups of isoleucines, leucines, and valines, and amide protons. Such constraints are typically obtained for large proteins that are deuterated except for these methyl groups to eliminate dipolar broadening. Examples are the 37-kDa structure of the T-TE didomain of the nonribosomal peptide synthetase EntF (22), or the 50-kDa structure of the C-domain of the same system (7).

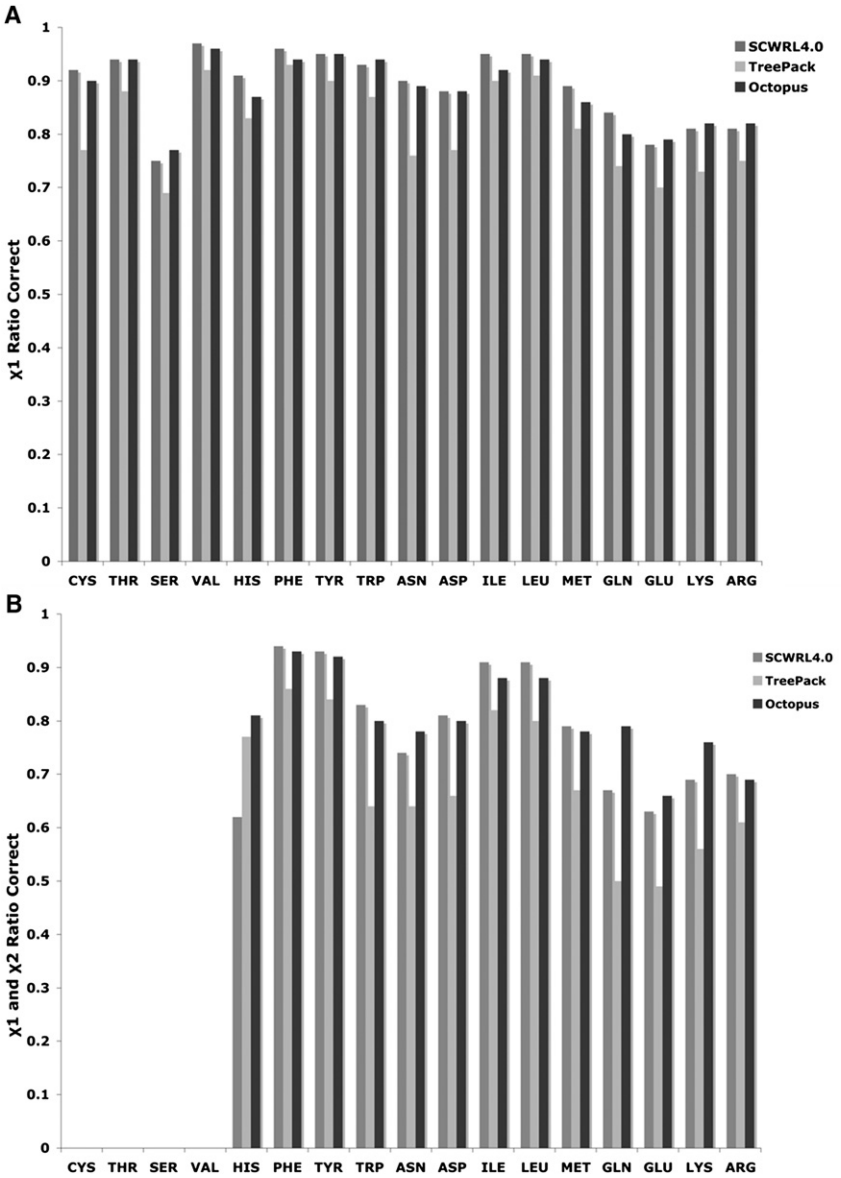

FIGURE 2 Comparison of the prediction of (*A*) χ_1 alone, and (*B*) χ_1 and χ_2. A dihedral angle is deemed "correct" if the deviation from the value in the protein crystal structure is no more than 40°. The prediction accuracy of a dihedral angle is the ratio of number of conformations within the "correct" range to the number of occurrence of the amino acid in the test set. Octopus is generally better than TreePack and comparable to SCWRL. In general, the prediction accuracy of hydrophobic residues is better than side chains with oxygen and nitrogen atoms.

4. Another application will be the placement of side chains in protein interfaces that are obtained based on NMR chemical shift mapping, cross-saturation experiments, mutation data, and molecular docking. The short computation times will enable the evaluation of binding energies for multiple docking orientations combined with side-chain rearrangements with Octopus.
5. The Octopus program is ideally suited for placing small molecules into binding sites on proteins identified with chemical shift mapping or NOE measurements. It will allow flexible docking with side-chain rearrangements within a rigid backbone. This will be suitable for many small molecules or drug leads that bind weakly to target proteins without causing backbone rearrangements.
6. The side-chain placement will facilitate assignments of NOE cross peaks in NMR structures in advanced stages of structure determination when the backbone fold is close to being defined.

There are certainly other applications in protein design and recognition of cofactors as well.

CONCLUSION

In this work we present a new, to our knowledge, approach that is able to place side chains in a given backbone structure without clashes and at negative energies essentially identical to those of the parent crystal structures. This is achieved by using side-chain cover sets, which allow placing side chains at any desired fine resolution, and on the fly. The approach does not rely on predefined rotamer libraries where the resolution is limited by the rotamer classes. We obtain low and

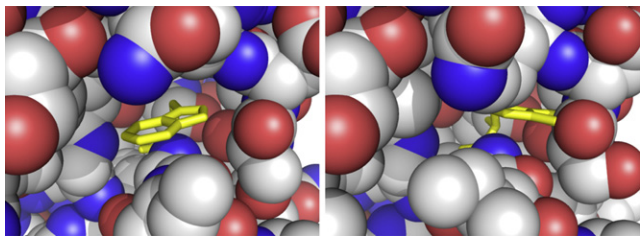

FIGURE 3 Comparison of side-chain placement with Octopus (*left*) and TreePack (*right*) programs. As an example, side-chain placement was performed for a dehydrogenase (1BG6), and the placement of TRP-33 was inspected. (*Left*) Placement with Octopus, which is almost identical to the crystal structure. (*Right*) Placement with TreePack where atoms of the side chain of TRP-33 are clashing with several neighboring atoms. This figure was produced with the program PyMol (25).

correct energy values with the standard and exact Lennard-Jones potential that does not have to be modified to alleviate clashes. The approach has numerous applications ranging from homology modeling and ligand binding to facilitating NMR structure determination.

Future work will include automatic determination of the value of R for different residues: generation of an isolated residue does not need to be performed at high resolution, but generation of a residue located in the presence of many other residues such as in the docking site of a protein-protein complex should be performed at the highest practical resolution. Another conclusion of this work is that use of rotamer libraries is not necessary and that generation of side-chain conformations is a viable option with the added advantage of predicting structures with similar energies found in the PDB. We plan to use the Octopus program in docking small molecules with complete side-chain flexibility—something other approaches are currently unable to do.

APPENDIX A: GENERATION OF SIDE-CHAIN COVER SETS

Generation of cover sets for single dihedral side chains

A dihedral angle is specified by four atoms, (a_1, a_2, a_3, a_4). Numbering atoms such that an atom is further from the backbone with increasing index (for example, for valine, if a_1 is the backbone nitrogen, a_2 is the α-carbon, a_3 is the β-carbon, and a_4 is the γ_1-carbon) if we fix a_1, a_2, and a_3 and rotate a_4 about the a_2–a_3 bond, the dihedral angle will change. We can generate many different conformations for the side chain of valine by changing its dihedral angle.

To meet the density requirement for the SCCS-R, we move a_4 over the entire 2π range by $2R$ Å increments in successive rotations. The angular increment must therefore be $2R/d_1$ radians where d_1 is the length of the perpendicular from a_4 to the a_2–a_3 bond (the rotation axis). To see that the set generated using an angular increment of $2R/d_1$ radians is an SCCS-R, consider an arbitrary side-chain conformation; the position of its a_4 atom will be, in general, between the positions of the a_4 atoms of two successive conformations in the set. In the worst case, it will be exactly between the positions of the a_4 atom of two successive conformations. Thus there is at least one member of the set where the deviation of the a_4 atom is at most R Å from the arbitrary conformation of the side chain and the RMSD of a_4 is, at most, R.

Using an angular increment of $2R/d_1$ generates a SCCS-R of minimal size. Using a larger increment makes the set lose its covering property, and a smaller increment is not necessary.

Calculation of angular increments for two dihedral side chains

To calculate the angular increments for the two nested dihedral angles (a_1, a_2, a_3, a_4) and (a_2, a_3, a_4, a_5), i.e., θ_1 and θ_2, which ensure that we generate a SCCS-R, we write an expression for the maximum RMSD to an arbitrary conformation

$$R^2 = \frac{1}{2}\left(\delta^2(a_4) + \delta^2(a_5)\right), \tag{1}$$

where $\delta(a_4)$ and $\delta(a_5)$ are the maximum deviations of a_4 and a_5, respectively. From the case of a single dihedral side chain, $\delta^2(a_4) = {d_1}^2{\theta_1}^2/4$. To calculate $\delta^2(a_5)$, consider two successive positions for a_4, i.e., p_1 and p_2, from each of which there will be successive rotations by θ_2 increments that generate two circles centered at p_1 and p_2. The position of a_5 that would have the largest possible deviation from any of the points on the circles is when it lies exactly between the two circles and also exactly between two pairs of points on the circles, i.e., in the middle of a rectangle whose sides are bounded by $d_2\theta_1$ and $d_3\theta_2$ (if that is not the case, i.e., a_5 is closer to one circle over the other, its deviation would be smaller than what is computed here). Thus, we have

$$\delta^2(a_5) = \frac{d_2^2\theta_1^2}{4} + \frac{d_3^2\theta_2^2}{4}. \tag{2}$$

Letting $\theta_2 = k_1\theta_1$, the maximum RMSD of the two atoms can now be computed from

$$R^2 = \frac{\theta_1^2}{8}\left(d_1^2 + d_2^2 + d_3^2k_1^2\right). \tag{3}$$

The number of possible positions for a_4 and a_5 is given by $2\pi/\theta_1$ and $2\pi/\theta_2$, respectively, and the size of the SCCS is

$$N = \frac{4\pi^2}{k_1\theta_1^2}. \tag{4}$$

For a fixed R, we find the optimal k_1 that will result in the smallest size SCCS-R given the upper bounds on the deviations of a_4 and a_5. This can be obtained by solving for k_1 in

$$\frac{dN}{dk_1} = 0. \tag{5}$$

Using Eq. 3, we thus obtain

$$k_1^2 = \frac{d_1^2 + d_2^2}{d_3^2}. \tag{6}$$

Substituting in Eq. 3, the values of θ_1 and θ_2, in radians, that will ensure that the smallest size SCCS-R will be

$$\theta_1 = \frac{2R}{\sqrt{d_1^2 + d_2^2}} \text{ and } \theta_2 = \frac{2R}{d_3}. \tag{7}$$

Generation of cover sets for three-dihedral side chains

Rotating about the three dihedrals bonds by θ_1, θ_2, and θ_3, respectively, we calculate the maximum RMSD of an arbitrary conformation

$$R^2 = \frac{1}{3}\left(\delta^2(a_4) + \delta^2(a_5) + \delta^2(a_6)\right), \quad (8)$$

where a_6 is the atom at the end of the third dihedral and $\delta(a_6)$ is its maximum deviation. Let d_4 be the maximum length (over all possible conformations) of the perpendicular from a_6 to the a_2–a_3 bond (the first dihedral bond), d_5 be the maximum length (over all possible conformations) of the perpendicular from a_6 to the a_4–a_4 bond, and d_6 be the length of the perpendicular from a_6 to the a_4–a_5 bond. The maximum deviation of a_6 is calculated in the same manner as the deviation of a_5 in the previous section, and we have

$$R^2 = \frac{\theta_1^2}{12}\left(d_1^2 + d_2^2 + d_4^2 + \left(d_3^2 + d_5^2\right)k_1^2 + d_6^2k_2^2\right), \quad (9)$$

where $\theta_3 = k_2\theta_1$. The number of possible positions for a_6 is given by $2\pi/\theta_3$ and the size of the SCCS is

$$N = \frac{8\pi^3}{k_1k_2\theta_1^3}. \quad (10)$$

To obtain a SCCS-*R* with the smallest size given the atom deviations, we solve for k_1 and k_2 in

$$\frac{\partial N}{\partial k_1} = 0 \text{ and } \frac{\partial N}{\partial k_2} = 0 \quad (11)$$

to get

$$k_1^2 = \frac{d_1^2 + d_2^2 + d_4^2}{d_3^2 + d_5^2} \text{ and } k_2^2 = \frac{d_1^2 + d_2^2 + d_4^2}{d_6^2}. \quad (12)$$

The values of θ_1, θ_2, and θ_3, in radians, that will ensure the smallest size SCCS-*R*, will be

$$\theta_1 = \frac{2R}{\sqrt{d_1^2 + d_2^2 + d_4^2}},\ \theta_2 = \frac{2R}{\sqrt{d_3^2 + d_5^2}} \text{ and } \theta_3 = \frac{2R}{d_6}. \quad (13)$$

For four-dihedral side chains, the calculation of their angular increments that ensures a SCCS-R with the smallest size, given the atom deviations, proceeds in the same manner and produces increments with the same pattern.

APPENDIX B: TREE DECOMPOSITION OF A GRAPH

Definition

Let $G = (V, E)$ be a graph, a tree decomposition of G is a pair (T, X) satisfying the the conditions:

1. $T = (I, F)$ is a tree where I is the set of nodes of the tree and F its set of edges.
2. $X = \{X_i\} i \in I, X_i \in V\}$ and $\cup_{i\in I} X_i = V$. Each node in the tree T is a subset of V and the union of the subsets is V.
3. For every edge $e = \{v, w\} \in E$, there is at least one $i \in I$ such that both v and w are in X_i.
4. For all $i, j, k \in I$. If j is a node on the path from i to k in T, then $X_i \cap X_k \subseteq X_j$.

The tree decomposition of a graph is not unique; for example, the trivial decomposition consisting of a tree with a single node will result in a very inefficient optimization algorithm. It is thus desirable to have a tree decomposition where the size of the largest subset of residues per tree node is small. The notion of tree-width was also introduced by Robertson and Seymour (20) to differentiate between tree decompositions for the same graph. The tree width of a tree decomposition is $max_{i\in I}\ \{|X_i| - 1\}$, i.e., the node with the maximum number of vertices of the graph minus 1. The tree width of a graph G is the minimum width over all possible tree decompositions of G. Computing the tree width of a graph is NP-hard (23); however, many heuristics are known (24) for computing tree decompositions of a graph with small, but not necessarily smallest, tree width. One such heuristic is the minimum degree heuristic (24), which we use here.

APPENDIX C: DYNAMIC PROGRAMMING EQUATION

Let $D(i)$ denote the possible side-chain conformations for residue i and $D(X)$ denote the possible side-chain conformations of the subset of residues X. Also let $A(X)$ be an assignment of conformations to residues of subset X from $D(X)$. Let X_r be the parent of node X_j in the tree and $C(j)$ be the set of children of the node X_j. If all the residues in $X_r \cap X_j$ are removed from the tree, the tree splits into two separate subtrees. Assume that the side-chain conformations of residues in $X_r \cap X_j$ have been assigned conformations $A(X_r \cap X_j)$. Define $F(X_j, A(X_r \cap X_j))$ to be the minimum energy of the subtree rooted at X_j given that the side-chain assignment for $X_r \cap X_j$ is given by $A(X_r \cap X_j)$. Because the energy at a tree node depends only on the assignment to the residues in common with its parent (and not on the assignment of its siblings in the tree), the minimum energy of the tree rooted at X_j can be written as the recursive equation

$$F\left(X_j, A\left(X_r \cap X_j\right)\right) = \min_{A\in D\left[X_j - X_r \cap X_j\right]} \sum_{i\in C(j)} F\left(X_i, A\left(X_j \cap X_i\right)\right) + E\left(A\left(X_j\right)\right).$$

The calculation of the value of the minimum energy starts at the leaves of the tree and proceeds upwards. Computation of the energy at a node can start only when the energy has been computed for all of its children.

For node X_i, let X_j be its parent. For each possible assignment $A(X_i \cap X_j)$, from $D(X_i \cap X_j)$, enumerate the assignments of residues in $X_i - X_i \cap X_j$, and find the assignment that will minimize the energy of the subgraph induced by X_i. This energy is recorded in a table (as the energy of the residues in $X_i \cap X_j$) because it will be used later to minimize the energy at the parent node. In addition, the assignment from $X_i - X_i \cap X_j$ that minimized the energy is recorded so that in the next stage, the assignments that minimize the energy for the protein can be found. At the root of the tree, all possible assignments to residues in the root node are enumerated and the energy of each of its children is added. The energy at the root node is the energy of the entire protein.

In the top-down stage, starting at the root node, the assignments to the residues at the root node that minimizes the energy can be found. Recursively, this assignment is passed down to the children and the assignments to the residues at the children that minimize the energy can also be found.

A.F. thanks Deani Cooper, Sabastian Hiller, and John Myers for valuable discussions.

This research was supported by National Institutes of Health grant No. GM 47467.

REFERENCES

1. Pierce, N. A., and E. Winfree. 2002. Protein design is NP-hard. *Protein Eng.* 15:779–782.
2. Garey, M., and D. Johnson. 1979. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman, New York.
3. Desmet, J., M. De Maeyer, …, I. Lasters. 1992. The dead-end elimination theorem and its use in protein side-chain positioning. *Nature.* 356:539–542.
4. Goldstein, R. F. 1994. Efficient rotamer elimination applied to protein side-chains and related spin glasses. *Biophys. J.* 66:1335–1340.
5. Pierce, N., J. Spriet, …, S. Mayo. 2000. Conformational splitting: a more powerful criterion for dead-end elimination. *J. Comput. Chem.* 21:999–1009.
6. Gordon, D., and S. Mayo. 1998. Radical performance enhancements for combinatorial optimization algorithms based on the dead-end elimination theorem. *J. Comput. Chem.* 19:1505–1514.
7. Canutescu, A. A., A. A. Shelenkov, and R. L. Dunbrack, Jr. 2003. A graph-theory algorithm for rapid protein side-chain prediction. *Protein Sci.* 12:2001–2014.
8. Xu, J., and B. Berger. 2006. Fast and accurate algorithms for protein side-chain packing. *J. Assoc. Comput. Mach.* 53:533–557.
9. Lee, C., and S. Subbiah. 1991. Prediction of protein side-chain conformation by packing optimization. *J. Mol. Biol.* 217:373–388.
10. Ponder, J. W., and F. M. Richards. 1987. Tertiary templates for proteins. Use of packing criteria in the enumeration of allowed sequences for different structural classes. *J. Mol. Biol.* 193:775–791.
11. Dunbrack, Jr., R. L., and M. Karplus. 1993. Backbone-dependent rotamer library for proteins. Application to side-chain prediction. *J. Mol. Biol.* 230:543–574.
12. Lovell, S. C., J. M. Word, …, D. C. Richardson. 2000. The penultimate rotamer library. *Proteins.* 40:389–408.
13. Bellman, R. 1957. Dynamic Programming. Princeton University Press, Princeton, NJ.
14. Krivov, G. G., M. V. Shapovalov, and R. L. Dunbrack, Jr. 2009. Improved prediction of protein side-chain conformations with SCWRL4. *Proteins.* 77:778–795.
15. Bower, M. J., F. E. Cohen, and R. L. Dunbrack, Jr. 1997. Prediction of protein side-chain rotamers from a backbone-dependent rotamer library: a new homology modeling tool. *J. Mol. Biol.* 267:1268–1282.
16. Xiang, Z., and B. Honig. 2001. Extending the accuracy limits of prediction for side-chain conformations. *J. Mol. Biol.* 311:421–430.
17. Dunbrack, Jr., R. L. 2002. Rotamer libraries in the 21st century. *Curr. Opin. Struct. Biol.* 12:431–440.
18. Pokala, N., and T. M. Handel. 2005. Energy functions for protein design: adjustment with protein-protein complex affinities, models for the unfolded state, and negative design of solubility and specificity. *J. Mol. Biol.* 347:203–227.
19. MacKerell, Jr., A. D., B. R. Brooks, …, M. Karplus. 1998. CHARMM: the energy function and its parameterization with an overview of the program. *In* The Encyclopedia of Computational Chemistry, *Vol. 1.* P. v. R. Schleyer, P. R. Schreiner, N. L. Allinger, T. Clark, J. Gasteiger, P. Kollman, and H. F. Schaefer, III, editors. John Wiley, London, UK: 271–277.
20. Robertson, N., and P. Seymour. 1986. Graph minors. II. Algorithmic aspects of tree-width. *J. Algorithms.* 7:309–322.
21. Berardi, M. J., W. M. Shih, …, J. J. Chou. 2011. Mitochondrial uncoupling protein 2 structure determined by NMR molecular fragment searching. *Nature.* 476:109–113.
22. Frueh, D. P., H. Arthanari, …, G. Wagner. 2008. Dynamic thiolation-thioesterase structure of a non-ribosomal peptide synthetase. *Nature.* 454:903–906.
23. Arnborg, D., L. Corneil, and A. Proskurowski. 1987. Complexity of finding embedding in a k-tree. *SIAM J. Algeb. Disc. Meth.* 8:277–284.
24. Berry, A., P. Heggerne, and G. Simonet. 2003. The minimum degree heuristic and the minimal triangulation process. *Lecture Notes in Computer Science, Springer-Verlag.* 2880:58–70.
25. Schrödinger, L. L. C. 2010. The PyMOL Molecular Graphics System, V. 1.3r1. Schrödinger, New York.