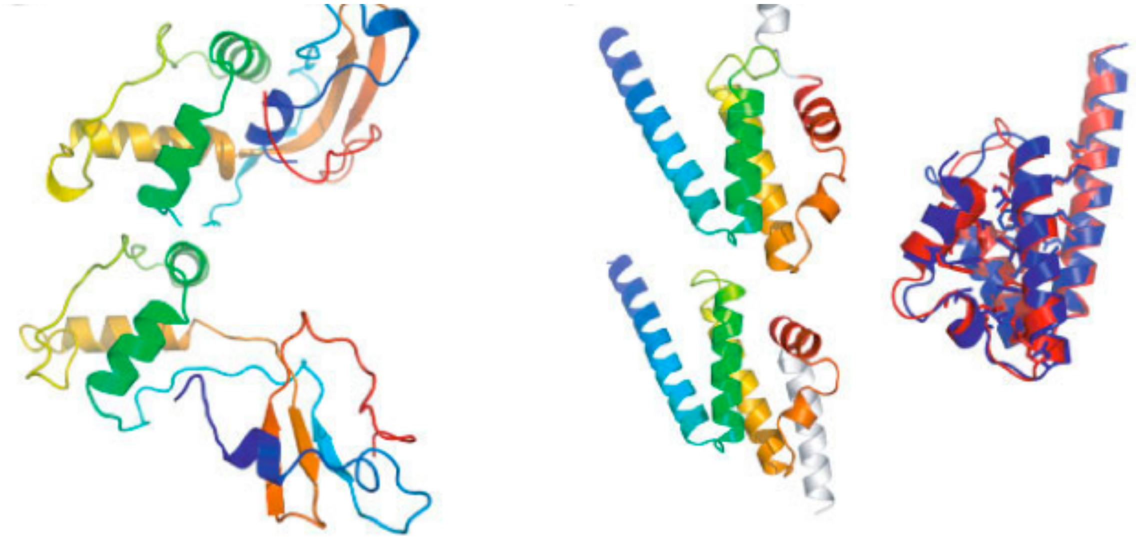
Progress in the Side-Chain Prediction Problem

Noah Youngs



Structure prediction results from Rosseta@home at CASP7 [Das et al., 2007]

1. Introduction

Accurate prediction of protein tertiary structure solely from sequence information has been an open problem for nearly half a decade. Though experimental biologists are able to determine protein structure through techniques like X-Ray Crystallography and more recently Nuclear Magnetic Resonance Spectroscopy, these methods are too slow and costly to enable the prediction of large numbers of hypothetical protein sequences. Instead, it is up to simulation techniques to try and guess what structure an amino acid sequence will fold to, before a single lab coat is donned. A variety of methods have shown promise, yet none provide consistently accurate results, at desired atomic resolution, while managing to remain computationally tractable even for large proteins. Despite the efforts of a large number of research groups, and the application of ideas from a plethora of related fields and problems, *ab initio* protein modeling is still not a reality.

The benefits to be gained from the solution of this problem, however, are quite clear. Drug-designers would be able create specific structures and target specific protein interactions on an as-yet unrealized scale. In addition, several currently incurable diseases are believed to be caused by protein misfolding, including cystic fibrosis, Alzheimer's, and Parkinson's [Cohen and Kelly, 2003]. Greater understanding of the folding process could lead to insight into the mechanisms of these diseases, or possible synthetic proteins to replace the misfolded ones. Finally, the biotech industry has in recent years seen an explosion of new and novel uses of enzymes outside the biological realm. Accurate structure prediction would be an immensely valuable tool for researchers in this field as well.

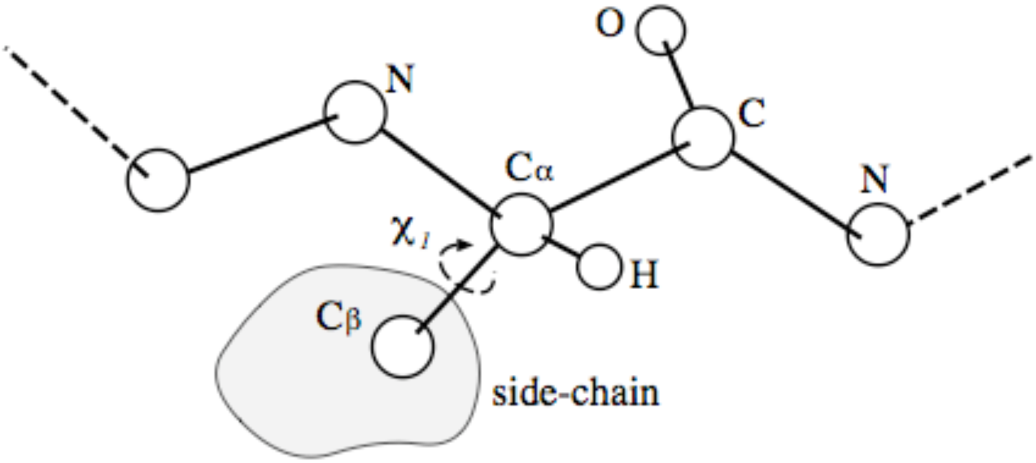
The problem is well defined, the benefits apparent, but the solution remains elusive. Here we discuss the progression of methodology for the Side-Chain Problem, a subset of the general folding problem, and compare new promising avenues of research, in the hopes that a synthesis of approaches might expedite the solution of this important problem.

2. The Side-Chain Problem

A. Definition

Many approaches to protein structure prediction break the problem down into two steps: backbone modeling and side-chain modeling. Several techniques exist that provide relatively good structure prediction for the backbone of a protein. One such technique is *homology modeling*, whereby backbone structure of a protein is assumed to be similar to that of other proteins homologous to the sequence. This method relies on the existence of known structures for homologues of a test protein sequence. If no such homologues exist, an alternative technique, known as *protein threading*, uses statistical techniques to compare subsets of a test protein sequence with other sequences in a databank, providing structural information without the need for homologues. Both techniques, however, rely on existing known structures. There exist *de novo* techniques for backbone prediction that have shown promising accuracy as well, but these will be discussed later.

Once the structure of the protein backbone is known, there is still the task of finding the conformation of its side-chains. Each side-chain can rotate around one or more dihedral angles, creating an immense number of possible structural forms, even with a stationary backbone. The different positions of the side-chains all affect the potential energy of the protein through well-known forces (van der Waals forces, columbic interactions, hydrogen bonding, etc.). Thus the Side-Chain Problem can be formalized as:



Protein backbone snippet with side-chain and dihedral angle χ_1, from [Akutsu, 1997]

> Given a protein backbone with n side-chains, find the set of torsion angles $\chi = (\chi_1, \chi_2 \ldots \chi_{n'})$ such that the potential energy function of the protein is minimized. (Note it is possible that n' > n since some side-chains have more degrees of rotational freedom).

B. Difficulty

The number of possible combinations of side-chain positions grows quite rapidly as the number of residues in a protein increases. Even breaking down possible rotational angles into discrete 10° movements, a group of five residues each with two dihedral angles already yields about 10^{15} possible conformations [Lee and Subbia, 1990]. As more residues are added, even if one decreases the possible angles to just a few likely ones for each dihedral, the size of the space that must be searched for the minimum energy conformation quickly becomes huge. For example, taking protein 1CD4 (PDB code), with 173 residues, and using only the most likely torsion angles at a resolution of 1.7 angstroms, there are $2\text{x}10^{111}$ possible side-chain conformations[Tuffery et al., 2004]. Even with a trillion^4 computers that could each search a trillion^4 combinations every second to see which had the least energy, it would still take 60 million years to look at every combination and find the smallest. Furthermore, the average protein size is about 300 residues

Beyond a "large search space" heuristic argument, there also exist several complexity proofs relating to the Side-Chain Problem. Abstracting the shapes of residues to simple blocks, and even only allowing a few possible angular values for each dihedral, it is provable that the problem of finding an assignment of angles that prevents any residue from intersecting any other residue is NP-complete [Akutsu, 1997]. This implies that there does not exist a polynomial-running time algorithm that can solve the Side-Chain Problem. By reformulating the Side-Chain Problem as a semidefinite program (a convex optimization problem over positive semitdefinite matricies), it can be shown that not only is the Side-Chain Problem not computable in polynomial time, it can not even be *approximated* in polynomial time [Chazelle et al., 2004].

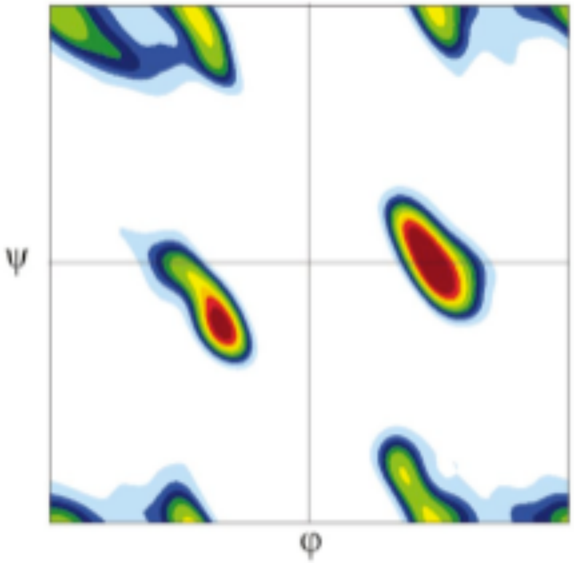
It is starting to become clear why the protein-folding problem is taking so long to resolve.

3. Initial Methods

A. Search Space Reduction

Several initial approaches to the Side-Chain Problem revolved around trying to reduce the size of the search space. The most intuitive way to do this is imply to discretize the possible angular values for each dihedral angle. In the 1960's two researchers, Ramachandran and Sasisekharan, plotted the two dihedral angles of backbone amino acids against each other on the same axis for a variety of experimental protein structures, discovering something quite interesting: there were only a few regions that had consistent data. Ramachandran was able to use this information to take all the possible combinations of torsion angles, and reduce them to just a few likely assignments, known as *rotamers*. This idea proved extendable to side-chain dihedral angles as well, and virtually every method that attempts to solve the Side-Chain Problem does so in *rotamer* space rather than angular space, drastically reducing the number of possible angles for each dihedral.

A more recent development in the reduction of possible side-chain conformation combinations came in 1991, with the Dead-End Elimination Theorem [Desmet et al., 1991]. Dead-End Elimination takes the global energy function, which is quadratic in the number of residues, not exponential, and thus computationally tractable, and uses it to form conditions under which certain rotamers are absolutely incompatible with the global minimum energy conformation. Thus certain rotamers, though probable for a given side-chain, can be discounted as dead-ends in terms of the protein structure as a whole, further reducing the search space.



Rachandran plot for glycine, from [Hovmoller et al., 2002]

Even with these reductions, however, the number of possible side-chain conformations remains large, and an efficient search strategy is necessary to provide useful results in a reasonable amount of time.

B. Search Strategies

A number of different strategies have been contemplated for searching rotamer space. One such strategy, simulated annealing, relies on the fact that even though the space is discretized, there are still regions of continuity where the energy is lower [Lee and Subbiah, 1991]. The intuition is that if all the rotamer assignments were correct, except for one, the global energy of the side-chains would still be very low. Thus if the search process pays attention to regions where energy is decreasing, it is more likely to end up near a minimum. Simulated annealing works by adding a temperature parameter constraining the step-length of the random search, one that decreases when a “valley” is found in the energy function. Thus the search is guided towards energy minima, however there is no guarantee that the minimum is global, and not local.

The local minima problem can be circumvented by a Monte Carlo approach, in which different random initial configurations are chosen for simulated annealing [Holm and Sander, 1992]. Among the many minima found by the Monte Carlo process, the one with the lowest energy is very likely to be the global minimum rather than a local minima, assuming enough trials have been performed. Unfortunately, by its nature, Monte Carlo processes are computationally intensive, since they require repeated searching from different starting values. This cost can be mitigated with the use of learned weights, but is still large.

A further enhancement to the search processes came with the use of Machine Learning techniques, namely neural networks. By using neural nets to create a distribution of side-chain dihedral angles, it is possible to further guide the Monte Carlo simulated annealing process [Hwang and Liao, 1995]. The distribution of dihedral angles can be used to trim away some of the possible rotamers, speeding up the Monte Carlo iterations. Yet even with all these refinements, the size and resolution of computable proteins, as well as the accuracy attainable, were far from desired levels.

C. Side-Chain With a Rotamer Library (SCWRL)

Bringing a new approach to the table, the SCWRL algorithm, first proposed in 1997, circumvented the need to search rotamer space for a global minimum [Bower et al., 1997]. The strategy instead was to use a rotamer library to choose the most favorable conformation for each residue, and then systematically move down through the less-favorable rotamers until one is found that does not conflict with the given backbone. This procedure is bound to create clashes between side-chains, but these are simply shoved into interacting "clusters", and resolved one at a time. If too many side-chains start interacting with each other, the cluster is subdivided and the process is iteratively repeated, until there are no more clashes. At this point, side-chains will theoretically be positioned at the lowest energy rotamers they could attain without causing clashes, up to an approximation factor.

SCWRL quickly became one of most widely used pieces of software for structure prediction, owing to its public access, and relative speed and ease of use. The algorithm went through several revisions over the next few years, but suffered from poor performance on non-native backbones, lack of incorporation of van der Waals forces, and overuse of search-space-reduction heuristics that sometimes eliminated the global minimum energy conformation as a possibility. SCWRL3.0 was introduced in 2003 to overcome these problems with a novel algorithm inspired by graph theory.

SCWRL3.0 begins with Dead-End Elimination to remove incompatible rotamers from the search space (rather than the previous, more complicated SCWRL heuristics). It then attempts to assign the best rotamer as before, and then creates an undirected graph from the clusters of interacting side-chains [Canutescu, et al., 2003]. This biconnected graph (a connected graph that cannot become disconnected with the removal of a single vertex) is broken down into subgraphs, and clusters that intersect multiple subgraphs are resolved first. The result is an algorithm dependent on the size of the largest cluster of interacting side-chains, rather than on the size of the entire side-chain space for the protein.

Comparing SCWRL3.0 with older versions of the algorithm, as well as other contemporary methods, similar or better accuracy was demonstrated, with computational time decreased by orders of magnitude, and previously intractable protein sizes now computable [Canutescu, et al., 2003].

4. Modern Methods and Benchmarks

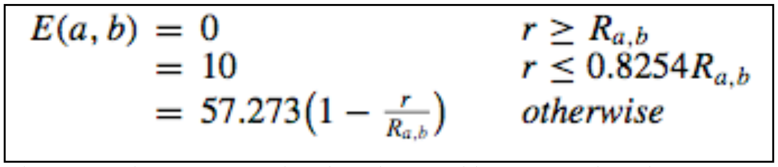
A. Critical Assessment of protein Structure Prediction (CASP)

As the number of different methods for protein structure prediction increased, it became necessary to have some kind of standardized rubric in order to judge their relative accuracies and speeds. Thus CASP was born, a bi-annual conference in which structure predictors could test their mettle. In order to fairly judge methods, a collection of unpublished experimental protein structures is collected for each CASP proceeding, and various labs are granted five attempts at predicting the structure from the given sequence. The predictions are submitted anonymously, and graded by an independent panel.

The trend uncovered by CASP proceedings over the years has been a steady, but slow, improvement in structure prediction capabilities. Unfortunately, after SCWRL3.0 there was little improvement in side-chain prediction over the next few years, and the report from CASP7 in 2006 described progress from the previous trials as “modest at best” [Kryshtafovych et al., 2007].

B. Tree-Decomposition

SCWRL3.0 opened the door for graph theory results to be applied to the Side-Chain Prediction Problem, one of the most notable being Tree Decomposition in TreePack [Xu and Berger, 2006]. TreePack uses the same simple energy function and notion of a residue-interaction graph as SCWRL, but solves the rotamer assignment problem in a different manner. The residue-interaction graph is decomposed into clusters, which are in turn fitted to a tree. This is a common procedure used on sparse graphs in NP-hard problems, and allows the transformation of a large graph into a smaller tree, with low-width. Once the tree decomposition is performed, the resulting cluster tree is traversed once from leaves to root, in order to determine optimal rotamer assignments, and then again from root to leaves to determine feasible rotamer assignments. The result is a fast algorithm that returns a set of rotamer assignments with near-minimum energy.



$$\begin{aligned} E(a,b) &= 0 && r \geq R_{a,b} \\ &= 10 && r \leq 0.8254 R_{a,b} \\ &= 57.273\left(1 - \tfrac{r}{R_{a,b}}\right) && \textit{otherwise} \end{aligned}$$

Simple pairwise inter-atomic energy scoring function used by TreePack and SCWRL3.0 [Xu and Berger, 2006]

Xu and Berger take a step usually omitted by the authors of other methods, which is to actually prove the running time of their algorithm. Since the computational cost of the algorithm is only dependent on the width of the tree decomposition, the total cost of the TreePack algorithm can be shown to be $O(Nn_{rot}^{O(N^{2/3}\log N)})$, where N is the length of the protein and n_{rot} is the average number of rotamers per residue. This bound broke the $O(Nn_{rot}^{N})$ barrier, and the authors were further able to use Dead-End Elimination and several approximation relaxations to achieve a polynomial running time of $O(Nn_{rot}^{4})$ on many proteins.

In practice, TreePack runs on average 5 times faster than SCWRL3.0 (running all the way up to 90 times faster in one case). TreePack was also able to predict the structures of several protein sequences that had been too large for SCWRL to deal with. Finally, despite its much greater speed, TreePack was able to maintain comparable accuracy to SCWRL3.0, and beat out other contemporary models, even on predictions of proteins with non-native backbones. But accuracy in general was still far too low to consider the side-chain problem solved.

In response to TreePack, SCWRL4 was released, which incorporated the use of tree-decomposition into the SCWRL algorithm, along with a dynamic programming optimization, and a denser energy function [Krivov et al., 2009]. Although no comparison with TreePack is given, the authors do show an increase in predictive power across the board from SCWRL3.0 to SCWRL4. The running time of the algorithm, however, is the same order of magnitude, if not slightly slower, highlighting the constant trade-off between speed and accuracy that plagues the protein-modeling world.

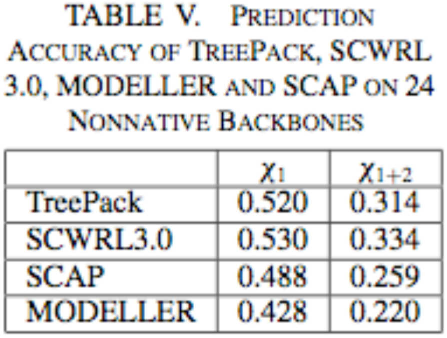


TABLE V. Prediction Accuracy of TreePack, SCWRL 3.0, MODELLER and SCAP on 24 Nonnative Backbones

	χ_1	χ_{1+2}
TreePack	0.520	0.314
SCWRL3.0	0.530	0.334
SCAP	0.488	0.259
MODELLER	0.428	0.220

Accuracy of side-chain first and first+second dihedral angle predictions using leading methods, from [Xu and Berger, 2006]

C. Rosetta

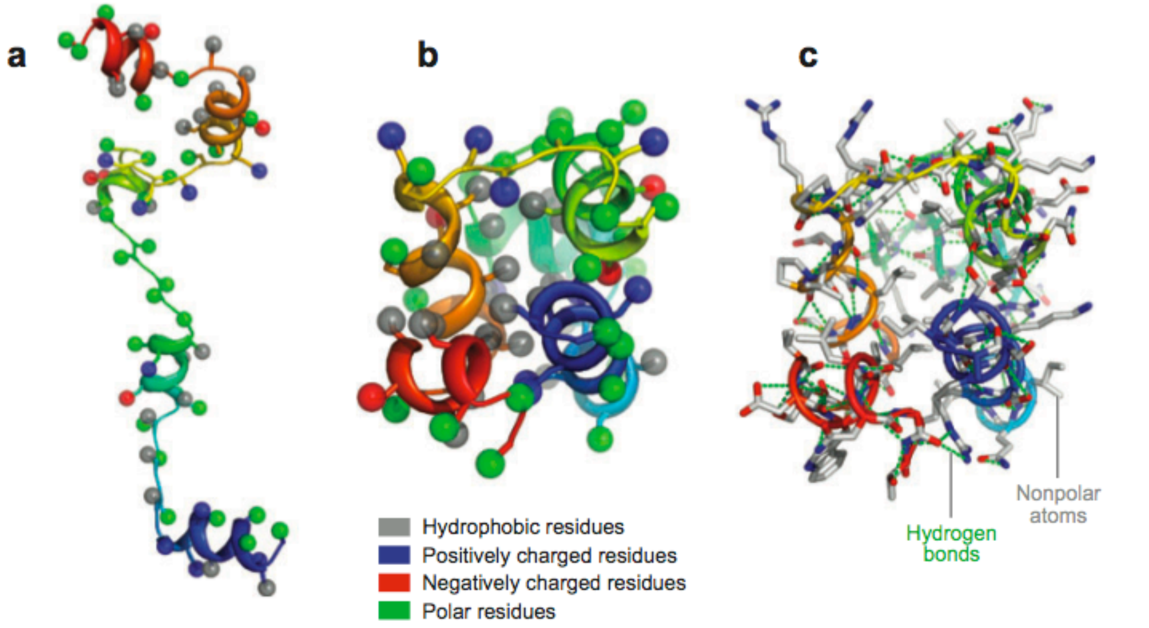
Another worthy mention in the protein structure prediction field is the Rosetta folding software, from the Baker Lab. Rosetta brings back the Monte Carlo simulated annealing search approach, but streamlines it, combining the side-chain conformation search with backbone perturbations in its structure prediction algorithm, and its protein-protein docking prediction algorithm (one step further than structure prediction, where two structures are introduced and the docking site between them estimated). Rosetta manages to incorporate more detailed energy functions than predecessors, while maintaining computational efficiency, with a combination of conformational sampling, low and high resolution passes, and pre-packing.

$$\begin{aligned} S = {} & w_{\mathrm{atr}} S_{\mathrm{atr}} + w_{\mathrm{rep}} S_{\mathrm{rep}} + w_{\mathrm{sol}} S_{\mathrm{sol}} + w_{\mathrm{sasa}} S_{\mathrm{sasa}} + w_{\mathrm{hb}} S_{\mathrm{hb}} \\ & + w_{\mathrm{dun}} S_{\mathrm{dun}} + w_{\mathrm{pair}} S_{\mathrm{pair}} + w_{\mathrm{elec}}^{\mathrm{sr\text{-}rep}} S_{\mathrm{elec}}^{\mathrm{sr\text{-}rep}} + w_{\mathrm{elec}}^{\mathrm{sr\text{-}atr}} S_{\mathrm{elec}}^{\mathrm{sr\text{-}atr}} \\ & + w_{\mathrm{elec}}^{\mathrm{lr\text{-}rep}} S_{\mathrm{elec}}^{\mathrm{lr\text{-}rep}} + w_{\mathrm{elec}}^{\mathrm{lr\text{-}atr}} S_{\mathrm{elec}}^{\mathrm{lr\text{-}atr}} \qquad (7 \end{aligned}$$

Rosetta energy score S for an atom, combinging attractive and repulsive van der Waals, solvation energy, hydrogen bond energy, rotamer probabilities, and coulombic potentials, each with a weight term that varies depending on which phase of iteration is currently being performed, from [Gray et al., 2003]

In order to save computation time, Rosetta first calculates backbone moves with simplified side-chain representations, and then optimizations the side-chain conformations in detail once every eight iterations [Gray et al., 2003]. This combination of low and high resolution passes prevents wasted effort, while still providing the necessary detail to keep the side-chains moving towards the minimum energy conformation. In addition, Rosetta utilizes a procedure similar to the initial steps of SCWRL, in which subsets of the protein sequence are “pre-packed”, with their rotamers set to the optimal conformation for that subsequence. Any clashes introduced by this pre-packing are then addressed in the course of normal iteration.

Rosetta also introduced the concept of conformational sampling, by which Monte Carlo search is aided by a library of sequence structures [Das and Baker, 2008]. As previously discussed, simulated annealing is sensitive to local minima, so Rosetta performs Monte Carlo search to find as many local minima as possible. In order to keep this process feasible for large numbers of Monte Carlo trials, the initial searches are performed at a very low resolution, with the structure library of residue conformations used as a distribution to aid in the calculation of energy score, and keep the error induced by the low resolution as small as possible. The next step is to pick a set of the lowest minima uncovered by the Monte Carlo search (not just the lowest one, since the low resolution and conformational sampling introduce uncertainty into the energy score) and re-optimize the side-chains with full-atomic resolution, making small changes to backbone torsions as well if necessary.



Conformational sampling procedure: **a**. Low-resolution representation of the protein **b.** Minimum energy conformation found at low-resolution **c.** High resolution side-chains introduced, rotamers re-optimized for the final conformation, from [Das and Baker, 2008]

The results of the algorithm are accurate and tractable (if not lighting-quick), with Rosetta consistently placing well in the CASP rankings of structure predictions.

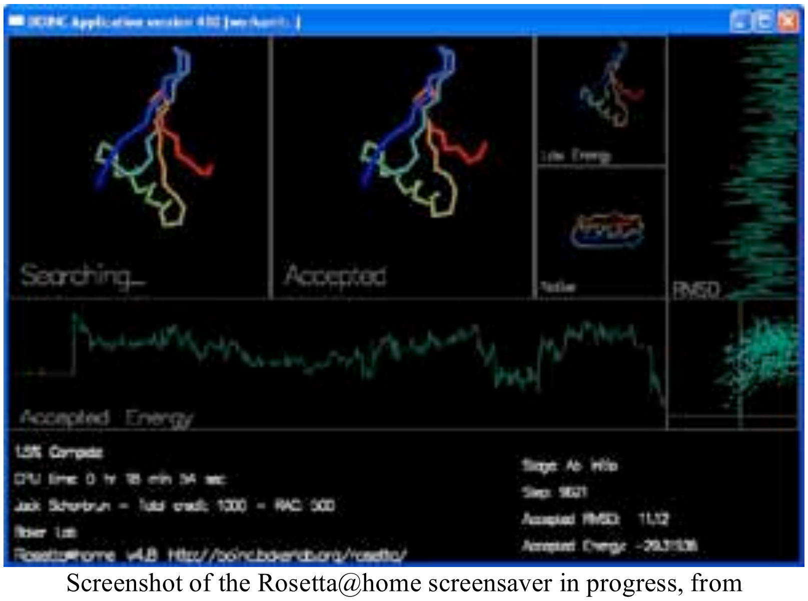
5. New Directions

A. Distributed Computing

Instead of seeking to decrease the difficulty of the problem, distributed computing attempts to increase the power of our algorithms, making previously intractable problems possible with sheer brute-force. This is accomplished by harnessing the power of private computing, farming out tasks to individual CPUs to be completed during down time, and then collecting the results in a central processor for integration. The SETI@home program famously attempted to do this in order to analyaze radio telescope data for signs of extraterrestrial life, accumulating more than 400,000 years of CPU time in just 36 months.

This technique was first applied to protein folding with the Folding@home project, from the Pande Lab at Stanford. Folding@home is different from other methods discussed here in that it is a molecular dynamics simulator, seeking not only to determine the final structure of a folded protein, but also how the folding processes itself plays out in real time. This is made possible in a distributed environment by a technique know as "ensemble dynamics", which utilizes parallel stochastic simulations with a high likelihood of there being one of the group that will exhibit correct behavior [Larson et al., 2002].

The distributed approach has since been applied to search-style protein structure prediction, and to side-chain conformation prediction, with Rosetta@home. Rosetta@home takes advantage of the fact that Monte Carlo searches are entirely parallelizable, thus removing one of the larger computational constraints from the Rosetta structure prediction and side-chain optimization process. Rosetta@home also includes an application called FoldIt, in which human users can help the Monte Carlo search by attempting to fold protein segments themselves (people are surprising good at folding proteins without any knowledge of energy functions or potentials.)



1.5% Complete
CPU time: 0 hr 18 min 34 sec
Jack Schonbrun - Total credit: 1000 - RAC: 500
Team: Lab
Rosetta@home v4.8 http://boinc.bakerlab.org/rosetta/

Stage: Ab Initio
Step: 5621
Accepted RMSD: 11.12
Accepted Energy: -29.31536

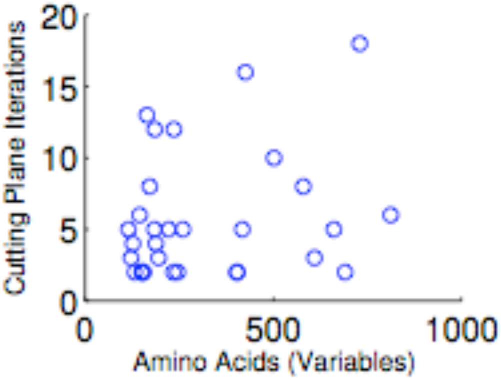
Screenshot of the Rosetta@home screensaver in progress, from http://boinc.bakerlab.org/rosetta/rah_graphics.php

B. Linear-Program Relaxation

Realizing that at its core, the Side-Chain Prediction problem is simply a minimization of a huge objective function comprised of the sum of inter-atomic energy potentials, it makes sense to try and apply standard optimization techniques to the problem. One such avenue of research formulates the Side-Chain Prediction Problem as a integer program with constraints stemming from the energy function used, and then relaxes to a linear program and attempts to solve. Ordinary linear program solvers are far too slow to be able to perform this processes on the number of variables needed for a side-chain conformation, however, an interesting shortcut can be taken. First, the set of rotamers can be treated as a probability distribution over the side-chain assignments, and then the integer program obtained is equivalent to a "maximum a posteriori" (MAP) assigment across the variables… the most likely position of rotamers, i.e. the minimum energy conformation. Then when the integer program is relaxed to a linear program, a Bayesian network technique know as belief propagation can be used to solve for the MAP assignment. In addition, the structure of the Side-Chain Problem can be exploited to employ Tree-Reweighted Belief Propagation for greater speed [Yanover et al., 2006].

The linear program approximation to the Side-Chain Problem can be modified to use either the simple energy function employed by SCWRL3.0, or the more detailed energy function used by Rosetta. Both forms were solvable by the LP approximation in a matter of minutes for most of the trial proteins, whose sizes ranged up to 1,000 residues [Yanover et al., 2006].

Using a new technique for belief propagation, called a cutting plane algorithm, Sontag and Jaakkola were able to improve the speed of the side-chain LP approximation, as well as solve several trial cases that Yanover et al. had been unable to compute earlier [Sontag and Jaakola, 2007]. These results were further improved upon in 2008, with the introduction of the Max-Product Linear Programming algorithm that further reduced the time necessary to compute the MAP assignment for the Side-Chain Problem [Sontag et al., 2008].



Number of iterations required to solve the Side-Chain Problem using LP relaxation and Map assignment, from [Sontag and Jaakola, 2007].

This method is quite fast compared to Monte Carlo searches, however it requires more robust testing, especially on proteins with non-native backbones, as well as integration into a general structure prediction system. Still it remains a promising avenue of research.

C. Anton

Another brute-force attempt at the molecular dynamics version of protein structure prediction is the Anton supercomputer, being built by a research team at the D.E. Shaw Investment Management Goup. While Folding@home uses parallelism to calculate parallel trajectories in search of a correct fold, Anton utilizes the ability of its parallel processors to quickly and efficiently communicate with each other in order to build a single, meticulous trajectory from start to finish [Shaw et al., 2007]. Making use of specialized hardware configuration, and well-understood force-potentials for its energy function, Anton promises to be able to compute lengthy atomic trajectories with unprecedented accuracy. The actual results, of course, will have to wait until the project is completed…

6. Conclusion

Many current avenues of research seem to be trending towards useful results in the protein structure prediction problem, yet somehow none emerge as a clear path to the final resolution of the problem.

Anton, while impressive in its sheer brawn and ability to predict structure completely *de novo*, seems lacking in the fact that many protein structures *can* be predicted from homologues or other know sequences. In addition, though the molecular dynamics process will be quite useful to science, raw structure prediction would also be extremely useful, and much of the massive parallelism Anton devotes to millisecond after millisecond of trajectories could be used to perform Monte Carlo searches on dozens of protein structures for their final conformation. In some senses, it seems like Anton is flexing its muscles just for the sake of flexing.

Linear program relaxation methods show vast potential for speed, but suffer from their relative newness in that they have not been rigorously tested, especially against experimental results. In addition, it remains to be seen whether the linear program framework can be extended to include the backbone as well, or whether the increase in size will confound the current message-passing algorithms. It would also be quite valuable to ascertain whether or not parallelization of the linear program relaxation process is possible, which with its already fast speed, might enable some sort of massive Monte Carlo LP relaxation approach that could circumvent any problems in accuracy or local minima.

Finally, Rosetta shows steady improvement and reliability, yet never seems quite fast enough or quite accurate enough. The Rosetta@home project is very promising, but it remains to be seen if the bottlenecks induced by the lack of communication between distributed processors will hamper results too drastically. An Anton-type machine designed for Rosetta calculations would be quite a powerful tool, maybe even one that continued to use Monte Carlo search for its backbone shifts and low-resolution passes but that incorporated linear program relaxation methods in its high resolution packing of side-chains.

As protein structure predictors prepare for CASP9 this year, it will be interesting to see who has made the most progress over the last two years, and perhaps some indication will be provided as to when we can at last put the problem of protein structure prediction to rest.

7. References

[1] Akutsu, T., NP-hardness results for protein side-chain packing. In Genome Informatics 8, S. Miyano and T. Takagi, Eds. 180–186, 1997.

[2] M. Bower, F. Cohen, R. Dunbrack, Jr., “Prediction of protein side-chain rotamers from a backbone-dependent rotamer library: a new homology modeling tool.” Journal of Molecular Biology, 267:1268-1282, 1997.

[3] A. Canutescu, A. Shelenkov, R. Dunbrack, Jr., “A graph-theory algorithm for rapid protein side-chain prediction.” Protein Science, 12.9:2001-2014, 2003.

[4] B. Chazelle, C. Kingsford, M. Singh, “Semidefinite programming approach to side-chain positioning with new rounding strategies.” INFORMS Journal on Computing, 16.4:380-392, 2004.

[5] F.E. Cohen, J.W. Kelly, “Therapeutic approaches to protein-misfolding diseases. Nature.” 426 905–909, 2003.

[6] R. Das, D. Baker, “Macromolecular modeling with Rosetta.” Annual Revue of Biochemistry, 77:363-82, 2008.

[7] R. Das, B. Qian, S. Raman, R. Vernon, J. Thompson, P. Bradley, S. Khare, M. Tyka, D. Bhat, D. Chivian, D. Kim, W. Sheffler, L. Malmstrom, A. Wollacott, C. Wang, I. Andre, D. Baker. “Structure prediction for CASP7 targets using extensive all-atom refinement with Rosetta@home.” Proteins: Structure, Function, and Bioinformatics 69.S8: 118-128, 2007.

[8] J. Desmet, M. De Maeyer, I. Lasters, “The dead-end elimination theorem and its use in protein side-chain positioning,” *Nature,* 356:539-542, 1992.

[9] J. Gray, S. Moughon, C. Wang, O. Schueler-Furman, B. Kuhlman, C. Rohl, D. Baker, “Protein-protein docking with simultaneous optimization of rigid-body displacement side-chain conformations.” Journal of Molecular Biology, 331:281-299, 2003.

[10] L. Holm and C. Sander, Fast and Simple Monte Carlo Algorithm for Side-Chain Optimization in Proteins: Application to Model.” PROTEINS: Structure, Funciton, and Genetics, 14:213-223, 1992.

[11] S. Hovmöller, T. Zhou & T. Ohlson, “Conformations of amino acids in proteins.” Acta Cryst. vol. D58, p. 768-776, 2002.

[12] J. Hwang and W. Liao, “Side-chain by neural networks and simulated annealing optimization.” Protein Engineering, 8.4:363-370, 1995.

[13] G. Krivov, M. Shapovalov, R. Dunbrack, Jr., “Improved prediction of protein side-chain conformations with SCWRL4.” Proteins: Structure, Function, and Bioinformatics, 2009.

[14] A. Kryshtafovych, K. Fidelis, J. Moult. “Progress from CASP6 to CASP7.” Proteins: Structure, Function, and Bioinfomatics, 69.S8:194-207, 2007.

[15] S. Larson, C. Snow, M. Shirts, V. Pande, “Folding@home and Genome@Home: Using distributed computing to tackle previously intractable problems in computational biology.“ Computational Genomics, Richard Grant, editor, Horizon Press, 2002.

[16] C. Lee, S. Subbiah, “Prediction of protein side-chain conformation by packing optimization,” Journal of Molecular Biology, 213:373-388, 1991.

[17] D. Shaw, M. Deneroff, R. Dror, J. Kuskin, R. Larson, J. Salmon, C. Young, B. Batson, K. J. Bowers, J. Chao, M. Eastwood, J. Gagliardo, J.P. Grossman, C. Ho, D. Ierardi, I. Kolossváry, J. Klepeis, T. Layman, C. McLeavey, M. Moraes, R. Mueller, E. Priest, Y. Shan, J. Spengler, M. Theobald, B. Towles, S. Wang, "Anton, a Special-Purpose Machinefor Molecular Dynamics Simulation." International Symposium on Computer Architecture, San Diego, CA, 2007.

[18] D. Sontag, T. Jaakkola, "New outer bounds on the marginal polytope." Advances in Neural Information Processing Systems 20, 2007.

[19] P. Tuffery, C. Etchebest, S. Hazout, R. Lavery, "A critical comparison of search algorithms applied to the optimization of protein side-chain conformations." Journal of Computational Chemistry, 14.7:790-798, 2004.

[20] J. Xu and B. Berger. "Fast and accurate algorithms for protein side-chain packing." Journal of the ACM, 53.4:533-557, 2006.

[21] C. Yanover, T. Meltzer, Y. Weiss, "Linear Programming Relaxations and Belief Propagation: an Empirical Study." Journal of Machine Learning Research, 7:1887-1907, 2006.