# OmniMerge: A Systematic Approach to Constrained Conformational Search

Lisa Tucker-Kellogg, Tomás Lozano-Pérez
M.I.T. Artificial Intelligence Laboratory

***Abstract*— OmniMerge performs a systematic search to enumerate all conformations of a molecule (at a given level of torsion-angle resolution) that satisfy a set of local geometric constraints. Constraints would typically come from NMR experiments, but applications such as docking or homology modeling could also give rise to similar constraints.**

**The molecule to be searched is partitioned into small subchains so that the set of possible conformations for the whole molecule may be constructed by merging the feasible conformations for the subchain parts. However, instead of using a binary tree for straightforward divide-and-conquer, OmniMerge defines a subproblem for every possible subchain of the molecule. Searching every subchain provides a counter-intuitive advantage: with every possible subdivision available for merging, one may choose the most favorable merge for each subchain, particularly for the bottleneck chain(s). Improving the bottleneck step may therefore cause the whole search to be completed more quickly. Finally, to discard infeasible conformations more rapidly, OmniMerge filters the solution set of each subchain based on compatibility with the solutions sets of all overlapping subchains. These two innovations—choosing the most favorable merges and enforcing consistency between overlapping subchains—yield significant improvements in run time.**

**By determining the extent of structural variability permitted by a set of constraints, OmniMerge offers the potential to aid error analysis and improve confidence for NMR results on peptides and moderate-sized molecules.**

***Index Terms*— distance geometry, conformational search, systematic search, NMR spectroscopy, structural biology, protein folding, computational biology.**

## I. Introduction

ATOMIC-LEVEL conformations of polypeptides are too small to observe directly but are crucially important for investigating biological functions, disease-related malfunctions, and potential drug interactions. We address the problem of enumerating the atomic-level conformations of a peptide that satisfy a given set of geometrical constraints. For example, our algorithm would list the "ensemble" of structures that are consistent with the interatomic distance bounds and dihedral angle bounds that can be obtained from NMR spectroscopy experiments [1]. Moreover, by using a systematic search engine instead of random sampling, we can guarantee a minimum level of resolution for the completeness of our coverage.

Most existing algorithms for providing molecular conformations consistent with NMR constraints are stochastic, typically involving randomness in the generation of trial conformations.

Simulated annealing is very popular, plus there are stochastic "distance geometry" algorithms [2], such as DIANA [3], developed specifically for the NMR structure determination problem.

Many of the remaining, systematic approaches to the constrained conformational search problem (sometimes called *restrained* search) involve ranking conformations according to a composite cost function that includes molecular energy terms as well as compliance with the constraints. In contrast, the problem we address is to determine whether conformations are acceptable or unacceptable, with no energetic preferences or relative order among the acceptable conformations.

There is an extensive literature on systematic approaches to the constrained conformational search problem. For reviews, see the book by Andrew Leach [4], the article by Martin Saunders *et al.* [5], or other sources [6]–[10].

One important commonality among systematic approaches is that the user chooses the size of the interval(s) to use when generating starting structures. In other words, the coverage is an independent variable. In contrast, the independent variables for a stochastic search might be the number of trials, the length of time, or even the probability of coverage, but never the absolute minimum coverage.

By extending the state of the art in systematic conformational search, we can improve how error is analyzed and confidence is measured for the structures of peptides and small molecules determined by NMR [11].

## II. Strategy

In our approach, the molecule to be searched is partitioned into small subchains, the satisfying conformations for these small subchains are enumerated *ab initio*, the sets of conformations for small subchains are combined or "merged" to create a set of candidate conformations for larger subchains, those satisfying conformations are combined to create candidates for yet larger subchains, until the conformations of the whole chain are determined. However, instead of using a previous divide-and-conquer strategy, three key innovations have been introduced:

**(1)** Instead of evaluating conformations at predetermined spacings, we divide the continuous, high-dimensional space of all conformations into small hypercubes or *voxels*. Instead of accepting or rejecting prespecified point at prespecified intervals in conformation space, we perform a brief (non-systematic) search within each voxel's continuous region of space.

**(2)** When dividing up a the overall search problem into multiple smaller searches of smaller parts, instead of just

choosing the obvious subchains of length $\frac{N}{2}, \frac{N}{4}, \frac{N}{8}, \ldots$ for a peptide of length $N$, OmniMerge searches a subproblem for every possible subchain of the molecule. Searching every subchain provides the advantage that every possible merge is available; by choosing the most favorable merge for each subchain, the bottleneck subchain(s) and therefore the whole search may be completed more efficiently.

**(3)** A propagation algorithm shares partial information by enforcing arc consistency between the solution sets of overlapping subchains. By filtering the solution set of each subchain, to make it consistent with the solution set of overlapping subchains, infeasible conformations are discarded rapidly.

## III. Methods

The constrained conformational search problem takes a primary structure (such as a protein sequence), some constraints on the 3–dimensional conformation of the molecule, and a desired level of resolution. The desired output is a list of voxels (at the specified resolution) that contain conformations that satisfy the constraints. For simplicity, we will often use a satisfying conformation in a voxel as a way to refer to the whole voxel.

### A. Torsion Space and the Voxel Model

The "rigid geometry" representation assumes that the lengths and angles of covalent bonds are held fixed at their equilibrium values. The only remaining degrees of freedom, dihedral rotations about single bonds, can then be represented as a vector of torsion angles. Likewise, the entire conformation space (or "torsion space") can be represented as $[0^\circ, 360^\circ)^d$, where $d$ is the number of torsions. A simplistic systematic search might impose a $d$-dimensional lattice on the space and evaluate the conformations at the gridpoints. However, the conformations that correspond to regularly spaced gridpoints are not necessarily representative of all nearby conformations. Satisfactory regions of conformation space that don't intersect the chosen gridpoints would be missed. While one might be tempted to think of all systematic search methods as guaranteeing complete coverage of the space, in fact these systematic methods only guarantee that some conformation in each region of space was considered.

There is no general, perfectly accurate way to determine whether a satisfying conformation exists in a given volume. However, we can certainly do better than evaluating one point per volume. The idea is to search each hypercube heuristically until a satisfying conformation is found or until an iteration limit is reached.

*1) Using minimization for heuristic search within voxels.:* We use minimization [12] of a constraint violation function as a heuristic for searching within a voxel for satisfying conformations. Constraints on the conformation of a molecule can be trivially converted to an objective function by summing the squared violations of the constraints. When the objective function is minimized to zero, then by definition a satisfying conformation has been found. Note that the objective function created here has nothing to do with potential energy or with ranking the desirability of conformations.

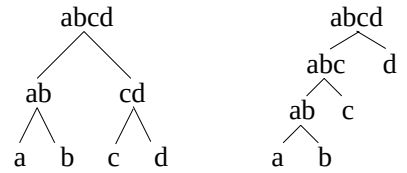


Fig. 1. Examples of alternative merge-trees for a tetrapeptide.

Minimization of the constraint violation function is a non-systematic method to evaluate a voxel. Although minimization is is not guaranteed to find satisfying conformations in every voxel that contains them, it is more accurate than using a single point per voxel.

To consider all voxels in the conformation space of a molecule is to perform a systematic conformational search of the molecule. Rather than enumerating satisfying conformations that happen to fall at regularly-spaced intervals, the goal of the search problem is now to output one satisfying conformation from each voxel that contains a feasible region.

### B. Divide-and-Conquer

The motivation behind divide-and-conquer is to perform some lower-dimensional searches on fragments of the molecule chain, prior to searching in the full $d$-dimensional space, so that infeasible regions may be excluded from consideration more quickly. For example, suppose the atoms in the immediate vicinity of torsion $i$ tend to collide with each other whenever $i$ is in the range $[\theta_l, \theta_h]$. In a naive algorithm that does not consider fragments of the molecule, one would have to examine all voxels that combine $[\theta_l, \theta_h]$ for torsion $i$ with values for the other $(d-1)$ torsions, requiring an exponential (in $d$) number of additional voxel evaluations.

With divide-and-conquer, we partition the molecule chain into smaller and smaller subchains, enumerate the solutions of the smallest subchains, and then combine the solutions for small subchains to create solutions for larger subchains, until solutions have been enumerated for the whole molecule. The repeated subdivision of the molecule chain according to a traditional divide-and-conquer strategy can be depicted using a binary tree or "merge-tree" such as in Fig. 1. The choice of merge-tree can be critically important to the efficiency of a divide-and-conquer search [1], [13].

The divide-and-conquer idea has appeared in many forms throughout the history of conformational search. Divide-and-conquer has been applied to docking searches [14] as well. In the mid-1980's, Scheraga and colleagues developed a "buildup" procedure [15]–[17] which has been successfully applied to a variety of applications [18]. Gippert *et al.* [13], [19] published a set of programs for systematic search including a divide-and-conquer module for use on top of an *ab initio* search module. Countless other methods build databases of protein structure fragments and then build models for new proteins by piecing together compatible fragments from their databases [20], [21]. These "model building" [4] methods are a form of "divide-and-conquer" with only one level of division and reconnection.

Our combine operation, like Gippert's, uses the satisfying conformations for two pieces of a chain to define candidate conformations for the whole chain. Suppose the only satisfactory conformation for peptide subchain ***abcd*** has torsion angles of $[\phi_1, \phi_2, \phi_3, \phi_4]$, and suppose the only feasible conformation for subchain ***efgh*** uses torsions $[\phi_5, \phi_6, \phi_7, \phi_8]$. Then the only torsion angles that could possibly satisfy the constraints on all the torsion angles in the merged chain ***abcdefgh*** would be $[\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6, \phi_7, \phi_8]$. No additional torsions (voxels) need be considered. If there were 2 feasible conformations for $abcd$ and 3 feasible conformations for $efgh$, then we would have to consider only the 6 combinations of angles to determine the full solution set for $abcdefgh$. With divide-and conquer, one searches for acceptable conformations (or voxels) for each single amino-acid individually, then combines the solutions of every other pair of amino acids into dipeptides using only the acceptable conformations of the individual halves. Then one combines the dipeptides into tetrapeptides, the tetrapeptides into octapeptides, and so on until all the potentially acceptable conformations for the whole molecule have been considered.

### C. OmniMerge

While performing a variety of conformational searches using divide-and-conquer, we observed that the vast majority of run time is consumed during the final combine operation to create the whole molecule. The time taken to search the "leaves" and short subchains in the merge-tree or to perform the low-dimensional merges is utterly dwarfed by the time for performing the higher-dimensional merges. Therefore we asked if there is any conceivable benefit to be reaped during high-dimensional searches (for example during the final merge) if we performed some additional lower-dimensional searches.

A key idea is searching all possible subchains instead of just the particular subchains specified in one merge-tree. Combining individual amino acids into dipeptides, combining those into tetrapeptides, and so on is only one possible way to perform the divisions of divide-and-conquer. There are $\frac{N(N-1)}{2}$ subchains of a peptide with $N$ amino acids, and there are $\frac{1}{N}\begin{pmatrix} 2(N-1) \\ N-1 \end{pmatrix}$ [22] possible merge-trees for $N$ residues.

The most obvious binary tree, dividing the peptide in half repeatedly, is not necessarily optimal. For example, the right half of the subchain might be entirely unconstrained (and thus would have a huge number of feasible conformations) except when constrained with respect to anchors on the left half of the molecule. Depending on the number of satisfying conformations per subchain, it is possible for a poor choice of merge-tree to take exponentially more time than an optimal choice [1].

Suppose we performed all possible dipeptide merges ($ab$, $bc$, $cd$, $de$, $ef$, ...) instead of just the pairs specified in some merge-tree (such as $ab$, $cd$, $ef$, ...). Obviously this would require twice as much work, but for the moment let us assume the cost of this extra work is insignificant compared with the cost of the final merge. Let OmniMerge be the algorithm that searches all possible subchains of the whole molecule, in order of increasing size, and that chooses the children for each combine operation so as to minimize the number of candidate conformations (voxels) being combined.

For example, suppose all subchains of length 1, 2, and 3 have been searched (i.e., they have known sets of voxels that satisfy the constraints). Then to complete the search of a subchain of length 4, such as $abcd$, we can choose whether to perform the combine operation using the solution set of $a$ with the solution set for $bcd$, or $ab$ with $cd$, or $abc$ with $d$. Because the solution sets for all possible subdivisions of $abcd$ are already available, we are not forced by default to combine $ab$ with $cd$. Choosing the most efficient subdivision means choosing the pair of "child" subchains with solution sets that would yield the smallest number of merged candidate conformations.

The preparation required to make this choice possible may seem like a great deal of overhead, but because the search space can grow exponentially with the number of torsion angles, the extra work required to perform this overhead on all the smaller subchains can consume less time than the work due to choosing a suboptimal subdivision for a larger subchain.

### D. Propagation

As we search every subchain of the molecule, many of the subchains will overlap each other, which is a form of redundancy. Any time we have two subchains that share at least one residue, we would like to reuse some of the information about what torsion angles are allowed or disallowed for those residues. One way to formalize this desire is with the concept of arc consistency (or constraint propagation), a common artificial intelligence technique for general constraint satisfaction problems. However, the name "constraint propagation" may be misleading because instead of applying the technique directly to the constraints on conformations (the distance constraints and dihedral constraints provided by NMR experiments), we create new, higher-level constraints requiring that overlapping subchains must have solution sets that are compatible with each other. To be more precise, each conformation of one subchain must be compatible with at least one conformation of the overlapping subchain, or else it is illegal. We use a simple propagation algorithm to enforce arc consistency between the solution sets of overlapping subchains. (Pseudocode is provided in reference [1].) This reduces the number of candidate conformations for each subchain and results in a substantial improvement to run time.

## IV. Results & Discussion

We focus here on demonstrating the advantages of searching all subchains and of using propagation. Experiments showing the performance improvements due to the voxel model and due to divide-and-conquer are available in reference [1].

The performance of OmniMerge may conceivably be better or worse than straight divide-and-conquer, depending on whether the default subchains used by divide-and-conquer are well-suited to the molecule and the constraint set. We

now show that when propagation is included, OmniMerge may outperform divide-and-conquer even when the default merge-tree is reasonably well-suited to the constraints on the molecule.

### *A. A Highly-Constrained Peptide with 40 Degrees of Freedom*

The structure of the 9-residue *Strep*-tag peptide is know from X-ray crystallography of the peptide-streptavidin complex [23]. (See Fig. 2.) The name of its entry in the protein data bank [24] is 1RST.)

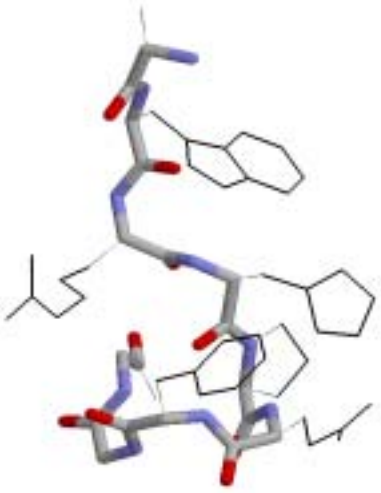

Fig. 2. Atomic structure of the *Strep*-tag, taken from its crystal structure in complex with streptavidin (1RST). The backbone bonds are shown as colored cylinders and the sidechains as black lines. Residue 1 is at the top.

Any pair of atoms separated by a distance between 2.5 and 6.0 Å in the crystal structure was constrained to a 0.1 Å interval about the measurement from the crystal structure. Of the 40 torsions, 32 rotate fully and 8 are peptide bonds ($\omega$ torsion angles) with rotation limited to be within $175°$–$185°$. The 1RST peptide has several long sidechains that contribute considerably to the conformational search problem. Because the sidechains were so flexible, even in the presence of the constructed constraints, we chose to search them at lower resolution than the backbone angles. The resolution of the search was $40°$ for backbone angles and $120°$ for sidechains.

The van der Waals radii were set to 85% of the half sigma values[1] from the CNS library `parallhdg.pro` [25], [26]. The summed squared violations were restricted to $\leq 0.0005$ Å$^2$. Voxels were evaluated using up to two passes of minimization with up to 50 steps per pass.

There are many ways to prune a "full" binary tree to serve as a merge-tree for a molecule with length not equal to an exact power of 2. We define a simplistic "default" merge-tree by taking the next largest full binary tree (with $2^{\lceil logN \rceil}$ leaves) and deleting the rightmost vertices until $N$ leaves remain. In case this default merge-tree happens to be unusually unfavorable for 1RST, we also inspected the 1RST crystal structure visually and designed another merge-tree that we thought might place nearby residues (residues likely to have active constraints) together at low levels of the merge-tree, while also preserving balance (minimizing depth) for the tree. See Fig.3.

[1]Sigma is the distance at which two atoms of the same type have zero van der Waals interaction energy. Any closer and they would start to repel each other.

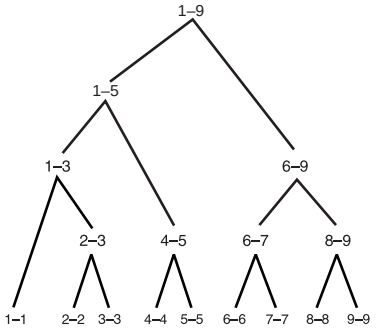


Fig. 3. A manually designed merge-tree for 1RST.

We searched the conformation space of the 1RST peptide using the bond lengths and bond angles from the 1RST crystal structure, not idealized values. However, because the 1RST crystal structure is highly refined for its particular conformation, using ideal van der Waals parameters would cause unavoidable clashes. Instead of reducing the van der Waals radii or increasing the allowed threshold for the summed squared violations, we omitted all steric constraints between atoms in the 1–4 positions of bonds.

Fig. 4 shows a comparison of run times for systematic conformational searches of 1RST using divide-and-conquer and using OmniMerge. The divide-and-conquer trials use either the default or the manually-constructed merge-tree, and the OmniMerge trials were performed with and without propagation.

| Search Method | Run Time (seconds) | Number of Minimizations | Conformations Found |
|---|---|---|---|
| divide-and-conquer with default merge-tree | 2046 | 6712 | 657 |
| divide-and-conquer with manually-constructed tree | 533 | 5189 | 662 |
| OmniMerge (no Propagation) | 536 | 7271 | 679 |
| OmniMerge with Propagation | 241 | 4390 | 664 |

Fig. 4. Run times for searching the 1RST peptide.

As expected, the manually-constructed merge-tree provides significantly better subdivisions of the molecule than a blind default merge-tree; run time performance improves by a factor of four. While we might not expect a merge-tree designed by human instinct to be perfectly optimal, it is surprising that OmniMerge without propagation can match its performance. With propagation, OmniMerge runs twice as fast. The reduced number of minimizations with and without propagation is sufficient to account for the faster speed. The reason propagation reduces the number of minimizations is that some voxels are discarded by virtue of having torsion ranges that are incompatible with the ranges of torsions in overlapping subchains. The simple inspection required to check for compatibility is

much faster than evaluating a voxel using multiple passes and iterations of minimization.

### B. A Uniformly-Constrained Helix

We constructed a more difficult set of test cases using a 16-residue helix of polyalanine. (See Fig. 5.) The length of

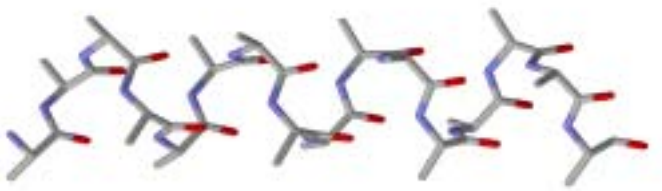

Fig. 5. Sixteen residues of alanine with $\phi = -57.0°$, $\psi = -57.0°$, and $\omega = 180.0°$.

$2^4 = 16$ residues was chosen so that the default merge-tree would be well-balanced, and short-range constraints were established randomly and uniformly along the molecule as follows: constraints were generated by forming the molecule into a helix, measuring all interatomic distances, selecting randomly among atom pairs with a distance between 2.5 and 6.0 Å, and constraining those pairs to maintain an interatomic distance that is within some threshold (called the "padding") of the measurement. Four independent constraint sets were generated, according to the parameters in Fig. 6.

| Trial | Number of Atom Pairs Constrained | Distance Constraint Padding (Å) | Sum-Sq Violation Allowed (Å$^2$) | Satisfying Conformations Found |
|---|---|---|---|---|
| A | 70 | 0.05 | 0.0005 | 6 |
| B | 73 | 0.05 | 0.0005 | 15 |
| C | 503 | 0.5 | 0.005 | 15 |
| D | 385 | 0.5 | 0.005 | 30 |

Fig. 6. Four sets of parameters used for generating constraints and running searches on 16-residue peptides of polyalanine. The right column shows the number of satisfying conformations found by the searches.

This exercise is not expected to be a favorable case for OmniMerge for several reasons. Because of the complete uniformity of the molecule and the relatively uniform number of constraints for each part of the molecule, we expect there will be little difference in cost between alternative merge-trees. Thus, we expect that the merges performed by OmniMerge (using the best subdivision for each search) will not be significantly better than the merges in the default merge-tree. Most significantly, the number of subchains searched by OmniMerge is quadratic in the length of the molecule, while the number of searches performed by divide-and-conquer is always linear.

Furthermore, we expect the searches of alternative large subchains to be a waste of time because there are no constraints on pairs of atoms separated by more than five residues in the sequence. The only new constraints that will become active when performing a combine operation on medium or large subchains will be short-range constraints at the interface between the two children. If a small subchain that overlaps the interface has already been searched and its results propagated, then the high level merges will involve virtually no pruning. We do expect propagation to be useful, although simply not performing unnecessary searches of "extra" overlapping subchains might be superior.

We searched the polyalanine peptide at $120°$ resolution for all bonds, with up to five passes of minimization per voxel, according to four different trial conditions and sets of constraints. All peptide bonds were constrained to between $175°$ and $185°$. Hydrogen atoms were excluded from the model.

Trial A constrained 70 pairs of atoms to within 0.05 Å of their measured distance in the helix and all constraints were considered satisfied when the summed squared violations were less than 0.0005 Å$^2$. All search methods found 6 satisfying conformations. Parallel statistics for each of the four trials appear in Fig. 6. Fig. 7 shows the results of searches using the four trial conditions.

| Search Method | Trial A Run Time | Trial B Run Time | Trial C Run Time | Trial D Run Time |
|---|---|---|---|---|
| divide-and-conquer with default merge-tree | 80.3 | 26.8 | 58.5 | 180 |
| OmniMerge (no Propagation) | 1002 | 749 | 103 | 199 |
| OmniMerge with Propagation | 20.5 | 47.7 | 13.6 | 63.2 |

Fig. 7. Run times (in seconds) for searching a 16-long peptide of polyalanine, subject to four different sets of uniform, random constraints.

As expected, OmniMerge without propagation performs poorly; it requires an order of magnitude more time than regular divide-and-conquer. Also as expected, propagation always improves performance. Perhaps the most interesting aspect of this result is the profound contribution of propagation, bringing the overall run time of OmniMerge into the same range as divide-and-conquer.

Uniform constraint sets, designed to sabotage OmniMerge, failed to degrade performance below the level of regular divide-and-conquer. This suggests that OmniMerge (with propagation) may be robust enough for general use in place of regular divide-and-conquer, regardless of the molecule and the constraint set.

### C. Costs and Benefits of Searching all Subchains

Quite counterintuitively, the run time of OmniMerge is sometimes better than divide-and-conquer. If the subproblems defined by a particular merge-tree do not exploit the available constraints very well, it may be preferable to solve all subproblems (including the ones that don't exploit the constraints very well) rather than to solve only the default subproblems.

The key insight is that during most divide-and-conquer searches, most of the run time is consumed by one merge operation, usually the final merge of the right and left halves of the molecule. Because OmniMerge searches all subchains, it has many alternative left and right fragments to choose from when confronting the final bottleneck step. For a molecule with 10 residues, there will be 9 choices: one residue on the left and nine on the right, two on the left and eight on the

right, etc. Although the time required to search every possible fragment can be substantial, is can easily be less than the time to perform the final merge using a poor default choice of fragments. A crucial concept throughout this research is to improve the speed of searching a chain by making a good choice of which right and left fragments to combine.

Like OmniMerge, the "buildup" procedure of Scheraga *et al.* [15]–[17] also searches all subchains of the molecule. However, instead of choosing the most favorable left and right child subchains to merge (which is the key idea of OmniMerge), the buildup procedure always creates an $i$-residue subchain by merging the subchain for the first $i-1$ residues with the subchain for the last $i-1$ residues. The benefit of a large overlap between the right and left children is that longer-range constraints will have already been satisfied. (For example, if the merge is to create subchain 1–8, then constraints involving residues 1–7 or residues 2–8 have already been satisfied; the only newly active constraints would be between residues 1 and 8.) The disadvantage of a large overlap between the children is that longer subchains tend to have a larger number of satisfying conformations. (The number of conformations for the right child times the number of conformations for the left could be very large.) This disadvantage is not a problem for Scheraga *et al.* because they keep only a limited number of energetically-favorable conformations for each subchain.

### *D. Hurdles for Larger Applications*

We have performed some trials applying systematic search to larger peptides and small proteins, but at a moderately coarse resolution, the number of voxels that satisfy the constraints can be prohibitive to enumerate. Even when using extraordinarily tight constraints, the search time for moderate resolution can still be very large. For lower resolution searches of large molecules, incompleteness becomes the primary obstacle, since the objective function within each voxel often contains too many local extrema.

We have found cases with large molecules where a low-resolution search fails to find any satisfying conformations, even if the constraints have been designed around some known conformation. In other words, low resolution gives us the expected problem of false negatives. Then, as we increase the resolution of the search infinitesimally, the search finds an astronomical number of conformations for the whole molecule. This may be surprising but the reasons for it become clear with hindsight. Many boundaries of conformation space are hyperplanes instead of rugged manifolds, and if the orientation of a constraint boundary is degenerate or exactly parallel to the definition of the voxels (such as $\theta \leq 120^\circ$), then the size of the solution set will not necessarily be continuous as a function of resolution. The simplest case is when the feasible region of conformation space is a tall, thin rectangular slab. When the search fails to find any voxels with satisfying conformations, that means the feasible region, if it exists, must be narrower than the interval of the search. When the search explodes (when the solution sets become intractable), the feasible region must be wider than the interval of the search in several dimensions. Both situations can occur simultaneously in large molecules because there can be both overconstrained and underconstrained degrees of freedom in the same molecule at the same time.

## V. Conclusion

Determining the conformations of biological molecules is a high scientific priority for biochemists and for the pharmaceutical industry.

The purpose of the systematic search method is to enumerate all conformations of a molecule (at a given level of torsion angle resolution) that satisfy a set of local geometric constraints. Instead of using a previous divide-and-conquer strategies, the OmniMerge algorithm includes three main innovations: the voxel model, searching all subchains (which gives OmniMerge its name), and propagation.

The algorithm was implemented and its effectiveness (particularly time-efficiency) was compared with divide-and-conquer using a non-uniform peptide, for which OmniMerge performed better, and using a uniform, "worst-case" peptide, for which OmniMerge was neither better nor worse than divide-and-conquer. If this is indeed a worst case comparison, then OmniMerge (including propagation) is on average a more efficient systematic algorithm than divide-and-conquer for searching small and moderate-sized molecules.

## Acknowledgement

This work was supported in part by the Singapore-MIT Alliance.

## References

[1] L. Tucker-Kellogg, "Systematic conformational search with constraint satisfaction," Ph.D. dissertation, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA. USA., 2002.

[2] G. M. Crippen and T. F. Havel, *Distance Geometry and Molecular Conformation*. Somerset, England: Research Studies Press, Ltd., 1988.

[3] P. Güntert, W. Braun, and K. Wüthrich, "Efficient computation of three-dimensional protein structures in solution from nuclear magnetic resonance data using the program diana and the supporting programs CALIBAS, HABAS and GLOMSA," *J. Mol. Biol.*, vol. 217, pp. 517–530, 1991.

[4] A. R. Leach, *Molecular Modeling: Principles and Practice*. Addison–Wesley, 1996.

[5] M. Saunders, K. N. Houk, Y.-D. Wu, W. C. Still, M. Lipton, G. Chang, and W. C. Guida, "Conformations of cycloheptadecane. a comparison of methods for conformational searching," *Journal of the American Chemical Society*, vol. 112, pp. 1419–1427, 1990.

[6] A. E. Howard and P. A. Kollman, "An analysis of current methodologies for conformational searching of complex molecules," *Journal of Medicinal Chemistry*, vol. 31, no. 9, pp. 1669–1675, 1988.

[7] H. Bohm, G. Klebe, T. Lorenz, T. Mietzner, and L. Siggel, "Different approaches to conformational analysis: A comparison of completeness, efficiency, and reliability based on the study of a nine-membered lactam," *Journal of Computational Chemistry*, vol. 11, no. 9, pp. 1021–1028, 1990.

[8] D. D. Beusen and E. F. B. Shands, "Systematic search strategies in conformational analysis," *Drug Discovery Today*, vol. 1, no. 10, pp. 429–437, 1996.

[9] J.-P. Doucet and J. Weber, *Computer-Aided Molecular Design: Theory and Applications*. Academic Press, 1996, ch. 7.2 Exploring the conformational space, pp. 206–220.

[10] M. Saunders, *Encyclopedia of Computational Chemistry*. John Wiley and Sons, 1998, vol. Volume 4, ch. Systematic and Random Search Methods for Finding Conformers of Molecules, pp. 2948–2950.

[11] C. M. Rienstra, L. Tucker-Kellogg, C. P. Jaroniec, M. Hohwy, B. Reif, M. T. McMahon, B. Tidor, T. Lozano-Pérez, and R. G. Griffin, "De novo determination of peptide structure with solid-state magic-angle spinning nmr spectroscopy." *Proceedings of the National Academy of Sciences, USA*, vol. 99, no. 16, pp. 10 260–5, 2002.

[12] C. T. Lawrence, J. L. Zhou, and A. L. Tits, "User's guide for CFSQP version 2.5: A C code for solving (large scale) constrained nonlinear (minimax) optimization problems, generating iterates satisfying all inequality constraints," Institute for Systems Research, University of Maryland, College Park, MD 20742, Tech. Rep. Technical Report TR-94-16r1, 1997.

[13] G. P. Gippert, "New computational methods for 3d NMR data analysis and protein structure determination in high-dimensional internal coordinate space," Ph.D. dissertation, Scripps Research Institute, La Jolla, California, 1995.

[14] J. Wang, P. A. Kollman, and I. D. Kuntz, "Flexible ligand docking: A multistep strategy approach," *Proteins: Structure, Function, and Genetics*, vol. 36, pp. 1–19, 1999.

[15] M. R. Pincus, R. D. Klausner, and H. A. Scheraga, "Calculation of the three-dimensional structure of the membrane-bound portion of melittin from its amino acid sequence," *Proceedings of the National Academy of Sciences (USA)*, vol. 79, pp. 5107–5110, 1982.

[16] M. Vásquez and H. A. Scheraga, "Use of buildup and energy-minimization procedures to compute low-energy structures of the backbone of enkephalin," *Biopolymers*, vol. 24, no. 8, pp. 1437–1447, 1985.

[17] K. D. Gibson and H. A. Scheraga, "Revised algorithms for the build-up procedure for predicting protein conformations by energy minimization," *Journal of Computational Chemistry*, vol. 8, pp. 826–834, 1987.

[18] J. Lee, H. A. Scheraga, and S. Rackovsky, "New optimization method for conformational energy calculations on polypeptides: Conformational space annealing," *Journal of Computational Chemistry*, vol. 18, pp. 1222–1232, 1997.

[19] G. P. Gippert, P. E. Wright, and D. A. Case, "Distributed torsion angle grid search in high dimensions: a systematic approach to NMR structure determination," *J. Biomol. NMR*, vol. 11, no. 3, pp. 241–263, 1998.

[20] T. A. Jones and S. Thirup, "Using known substructures in protein model building and crystallography," *EMBO Journal*, vol. 5, no. 4, pp. 819–822, 1986.

[21] L. Holm and C. Sander, "Database algorithm for generating protein backbone and side-chain co-ordinates from a C alpha trace application to model building and detection of co-ordinate errors," *Journal of Molecular Biology*, vol. 218, no. 1, pp. 183–194, 1991.

[22] D. E. Knuth, *The Art of Computer Programming, Second Edition*. Addison-Wesley, 1973, vol. 1, ch. 2.3.4.4 Enumeration of Trees, pp. 385–389.

[23] T. G. Schmidt, J. Koepke, R. Frank, and A. Skerra, "Molecular interaction between the strep-tag affinity peptide and its cognate target, streptavidin," *Journal of Molecular Biology*, vol. 255, pp. 753–766, 1996.

[24] H. M. Berman, J. Westbrook, Z. F. *et al.*, and P. E. Bourne, "The protein data bank," *Nucleic Acids Research*, vol. 28, pp. 235–242, 2000.

[25] A. T. Brunger, P. D. Adams, G. M. C. *et al.*, and G. L. Warren, "Crystallography and NMR system (CNS): A new software system for macromolecular structure determination," *Acta Crystallographica D*, vol. 54, pp. 905–921, 1998.

[26] R. A. Engh and R. Huber, "Accurate bond and angle parameters for X-ray protein structure refinement," *Acta Crystallographica, Section A*, vol. A47, pp. 392–400, 1991.