BioGeometry NEWS
May 2004
biogeometry.cs.duke.edu

Program for June 12 BioGeometry Meeting
Polytechnic University Brooklyn

9:00	Welcome by Jack Snoeyink & Pankaj Agarwal
9:10	Protein interfaces by Johannes Rudolph (Duke)
9:55	Prediction of protein structure and protein-protein interactions by Jeff Skolnick (SUNY Buffalo)
10:45	Break
11:00	An integrated approach to protein-protein docking by Rong Chen (Scripps Research Inst)
11:30	Fast geometric approach to predict protein docking configurations by Yusu Wang (Duke)
11:50	Placing hydrogens by dynamic programming by Xueyi Wang (UNC)
12:10	Comprehensive evaluation of protein structure alignment methods: scoring by geometric measures by Rachel Kolodny (Stanford)
12:30	Lunch
1:45	Euclidean Voronoi diagram of atoms and protein structure analysis by Deok-Soo Kim (Hanyang University)
2:05	The missing fragment problem in electron density map interpretation by Itay Lotan (Stanford)
2:30	Local conformation search by distance matrix perturbations by Ioannis Emiris (National University of Athens, Greece)
2:45	Break
3:15	Using spanners to describe protein structure by Daniel Russel (Stanford)
3:35	Using Turan graphs to compute COGS (Clusters of Orthologous Groups) by Craig Falls (UNC)
3:55	A bezier-based moving mesh framework for simulation with elastic membranes by David Cardoze (CMU)
4:15	Cell talk: algebraic model checking systems to reason about biological processes by Bud Mishra (NYU)
5:05	Wrapup

For meeting details, see http://biogeometry.cs.duke.edu/meetings/ITR/04jun12/

Research

Rigid Protein-Protein Docking

This work is a joint effort by Pankaj K. Agarwal, Vicky Choi, Herbert Edelsbrunner, Johannes Rudolph, and Yusu Wang.

Motivation: The reliable prediction interactions from three-dimensional structures of proteins is one of the grand challenges in computational molecular biology. X-ray crystallography and other structure determination methods show that there exists much local shape complementarity between two proteins. We use this insight as the starting point of our approach to predict protein-protein interactions. Specifically, we start with the geometric structures of individual proteins and search for a good fit in terms of shape complementarity. Even if we focus exclusively on the shape of proteins and ignore the fact that they may deform during interaction, the high dimension of the search space makes this a difficult problem.

In our work, we have so far focused on *rigid protein docking*, in which one asks for determining a motion that positions one rigid protein relative to the other into the correct docked configuration. Implementation of a fast and accurate rigid docking algorithm will help develop methods that add the higher dimensionality of conformational changes seen in real docking problems.

Problem Statement: We use solid spheres to represent atoms and space-filling diagrams to model proteins as unions of such spheres. Let $A = \{a_1,...a_n\}$ be the set of spheres defining the first protein, and let $B = \{b_1,... b_m\}$ be the set of spheres in the second protein. We assume that A is fixed and B is allowed to move. We use a score function that approximates the van der Waals interaction by counting the pairs of atoms that are close to each other. Our goal is to find a placement of B, allowing translation and rotation, that maximizes the score. An important constraint on this placement is it keeps the number of collisions, the pairs of intersecting atoms between A and B, below a prespecified threshold. We allow a few collisions to compensate for measurement and other modeling inaccuracies. We measure the success of our algorithm by testing it on known structures of protein complexes and by measuring the root-mean-square distance between the native configuration of B and the configuration of B computed by the algorithm.

Overall Approach: Our algorithm works in two stages. The first stage, called the *candidate generation*, identifies cavities and protrusions on the surface of each protein, estimates their size and shape, and generates a set of candidate placements for B by matching these features. We rank these placements using our score function. The second stage, called the *refinement stage*, chooses a subset of the top ranked placements and refines each of them using a local-improvement algorithm, so that the score is locally maximized and the number of collisions is as small as possible. We re-rank the placements of B based on the improved scores.

Generating Candidates: We use combinatorial Morse theory to identify cavities and protrusions on the surface M of a protein [1]. First assume that M is a smooth 2-manifold. We fix the origin. For each direction u, using the persistence algorithm [4], we pair each point $p \in M$ that is critical in direction u with another critical point q and define the *persistence* of p and q as the difference in their heights in direction u. Since M is smooth, each point p on M is critical for a pair of antipodal directions u and $-u$. We prove that the pairing of critical points is the same in both directions. It follows that the persistence of p is the same in both directions and we call this persistence the elevation of p. Although the elevation function is not continuous, we make it

continuous by performing surgery on M. The local maxima with sufficiently large elevation correspond to features such as cavities and protrusions of M. The persistence of a maximum estimates the size of the corresponding feature, and the direction in which a maximum is a critical point of M estimates the direction of that feature. We extend these concepts to piecewise-linear surfaces and develop an efficient algorithm for computing the maxima. We return a subset of maxima that have large elevation.

We run the above algorithm on both proteins A and B. Using the features returned by the algorithm on each surface, along with the estimates on their size and direction, we compute a set of placements of B at which these features match sufficiently well. We rank these placements using our score function and return a fixed number of the highest ranked placements.

Local Improvement: The input of the local-improvement algorithm is protein A and one of the placements B_i returned by the first stage of the algorithm. The local-improvement algorithm computes a local rigid motion of B_i to improve the score while keeping the number of collisions small [3]. Our algorithm is based on the following two intuitions: (i) worthwhile target positions for spheres in B_i are collision-free and locally maximize the score; (ii) an effective collection of target positions is approximately congruent to the configuration of corresponding sphere centers. Building upon these intuitions, we have developed an iterative algorithm that has two nested loops. Each iteration of the outer loop finds a local motion of B_i that improves the score but as a side-effect may increase the number of collisions. Each iteration of the inner loop reduces the number of collisions while keeping the score high. These two steps are repeated a fixed number of times, and we return the best configuration computed during the iteration.

The figure on the right shows the success-rate of the local improvement algorithm if started at local perturbations with specified rotation angle and translation distance away from the correct position.

Implementation and Results: The implementation of the two intuitions is delicate and requires a small number of experimentally-determined constants, including thresholds how far from an atom we search for target positions and the weights of atoms in the computation of a least-square optimum rigid motion that moves the atoms of B_i closer to their target positions.

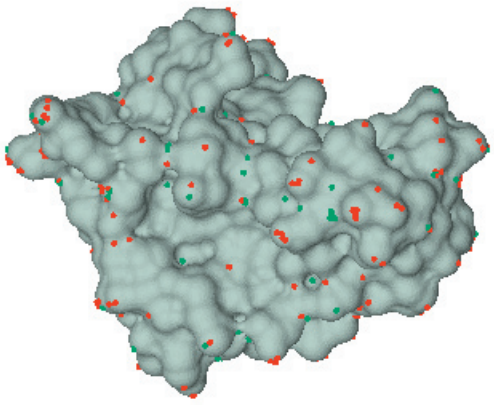
We have tested the two-stage approach on a collection of 17 protein complexes. At the end of the first stage, our algorithm returns 100 configurations $B_1,...B_{100}$ for protein B that have high scores. We then run the local improvement algorithm on (A, B_i), for each $1 \le i \le 100$ and re-rank the set of outputs by the improved scores. For all the 17 protein complexes, the configuration with highest score after the second stage has a low root-mean-square distance. In particular, for 10 out of the 17 cases, the root-mean-square distance of the top-scored configurations is less than 1.0 Å. For five of the remaining cases, it is below 1.5 Å, and for the last two, it is around 2.5 Å. For all protein complexes, including the last two cases, the local improvement algorithm significantly reduces the number of collisions and increases the score of near-native configurations.

[1] P. K. Agarwal, H. Edelsbrunner, J. Harer, and Y. Wang. Extreme elevation on a 2-manifold. Proc. 20th Annu. Sympos. Comput. Geom, 2004, to appear.

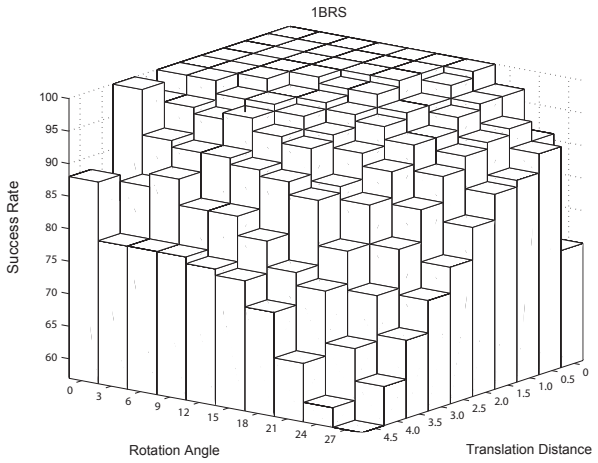
[2] S. Bespamiyatnikh, V. Choi, H. Edelsbrunner, and J. Rudolph. Accurate bound protein docking by shape complementarity. Manuscript, Dept. Comput. Sci., Duke Univ., Durham, NC, 2003.

[3] V. Choi, P. K. Agarwal, H. Edelsbrunner, and J. Rudolph. Local search heuristic for rigid protein docking.Submitted to Workshop on Bioinformatics, 2004.

[4] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. Discrete Comput. Geom. 28 (2002), 511--533.



The one hundred maxima with highest elevation on the surface of 1BRS (864 atoms)



Experimental results for running the local improvement algorithm on the 1BRS complex. The success rate is high as long as the rotation and translation distance to the correct position is not too large.

The BioGeometry project is funded by the National Science Foundation under grant CCR-00-86013. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.