Geometric Analysis of Cross-Linkability for Protein Fold Discrimination

S. Potluri, A.A. Khan, A. Kuzminykh, J.M. Bujnicki, A.M. Friedman, and C. Bailey-Kellogg

Pacific Symposium on Biocomputing 9:447-458(2004)

GEOMETRIC ANALYSIS OF CROSS-LINKABILITY FOR PROTEIN FOLD DISCRIMINATION

S. POTLURI[1], A.A. KHAN[1], A. KUZMINYKH[2], J.M. BUJNICKI[3],
A.M. FRIEDMAN[4], C. BAILEY-KELLOGG[1]
Depts. of [1]Comp. Sci., [2]Math., and [4]Biol. Sci., Purdue Univ., West Lafayette, IN 47907, USA
[3] Intl. Inst. Molec. and Cell Biol., Warsaw, Poland

Abstract

Protein structure provides insight into the evolutionary origins, functions, and mechanisms of proteins. We are pursuing a minimalist approach to protein fold identification that characterizes possible folds in terms of consistency of their geometric features with restraints derived from relatively cheap, high-throughput experiments. One such experiment is residue-specific cross-linking analyzed by mass spectrometry. This paper presents a suite of novel lower- and upper-bounding algorithms for analyzing the distance between surface cross-link sites and thereby validating predicted models against experimental cross-linking results. Through analysis and computational experiments, using simulated and published experimental data, we demonstrate that our algorithms enable effective model discrimination.

1 Introduction

Knowledge of protein structure is vital for understanding protein function and evolution. Traditional protein structure determination techniques, X-ray crystallography and nuclear magnetic resonance spectroscopy, provide atomic detail, but despite many advances, they remain difficult, expensive, and time-consuming techniques. Recent reports from labs conducting the high-throughput protein structure initiative[1] indicate that only 10 percent of expressed and purified proteins advance to full 3D structure. Alternatively, purely computational techniques (homology modeling, fold recognition, and *ab initio*) are much faster, but due to the inherent difficulty in scoring predictions, they encounter significant ambiguity in reliably identifying correct structures.

We seek a middle ground, verifying predicted structures against *minimalist* experiments that provide relatively sparse, noisy information relatively quickly and cheaply. In particular, this paper focuses on developing and applying geometric algorithms for model discrimination using data from residue-specific cross-linking, analyzed by mass spectrometry (Fig. 1). We assume here that the models have already been generated and the experimental data have been analyzed to identify a set of cross-links. We present algorithms for checking the consistency of the identified cross-links with the structure models, in order to discriminate among the models.

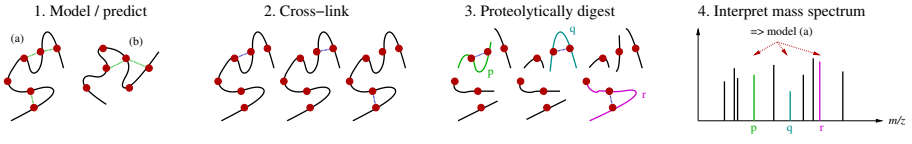

Figure 1: Cross-linking mass spectrometry protocol. (1) Computationally generate a set of possible structure models. (2) Specifically cross-link the protein using a small molecule of a fixed maximum length. (3) Digest the cross-linked protein with a protease. (4) Obtain and interpret a mass spectrum, using identified cross-links as evidence for spatial proximity and thus for a particular model.

Employing Edman sequencing and mass spectroscopy of cross-links, Haniu *et al.*[2] developed a largely correct model of human erythropoietin consistent with the cross-linking data, although no alternatives were explicitly considered. Later, Young *et al.*[3] pioneered the use of mass spectroscopy alone to correctly discriminate among threading models of Basic Fibroblast Growth Factor, FGF-2, in spite of very low sequence similarity. More recent work employs a "top-down" method to fragment proteins within a Fourier transform mass spectrometer, so as to focus on only singly cross-linked protein monomers[4]. Similarly, cross-linking has been used to determine tertiary and quaternary arrangements of proteins[5], including membrane proteins that are inherently difficult to crystallize[6,7]. The minimalist philosophy has also been applied by other groups in support of approximate structure determination. For example, a limited number of long-range distance constraints from NMR[8,9], mutagenesis followed by functional evaluation[10,11], chemical modification[12], and the pair distance distribution function from small-angle X-ray scattering[13], have all been employed.

While traditional structure determination techniques provide substantial overdetermination, minimalist experimental methods for rapid confirmation are noisy and yield only very sparse information. This places a significant burden on computational analyses to carefully characterize model geometry and maximize discriminatory power, in order to be robust to experimental noise and ambiguity. This paper develops a suite of new algorithms, trading complexity vs. accuracy, for analysis of cross-linkability in predicted structure models. The algorithms provide better discriminability and robustness than previously published approaches, and thus promise to enable broader applicability of cross-linking to protein fold identification.

2 Cross-Linkability Analysis

2.1 Problem Formulation

A cross-linker serves as a molecular ruler by linking only "close-enough" pairs of residues. Since the atoms of the cross-linker occupy physical space, the measurements are greatly constrained. We assume here that the cross-linker is energetically excluded

Input:
• Polyhedral *protein surface* S, representing the boundary of the body from which the cross-linker is excluded.
Let S_{int} denote the interior of the body.
• A set P of point *cross-linking sites* on S, representing potentially cross-linked atoms.
Computation:
Cross-linking paths between site pairs $p_i, p_j \in P$ and exterior to S_{int}.
Output:
For each pair of sites $p_i, p_j \in P$, *cross-linking distance* $D_*(i, j)$ as the minimum of the lengths of cross-linking paths between p_i and p_j.

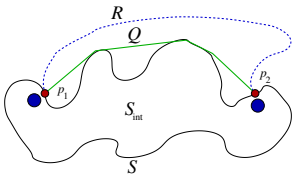

Figure 2: Cross-link problem formulation and 2D schematic illustrating surface S, atoms, cross-linking sites p_1 and p_2, and cross-linking paths Q (achieving cross-linking distance) and R.

from penetrating the protein interior. Since cross-linked residues (e.g. `Lys`) must be on or near the protein surface in order for the cross-linker to react with them, we represent cross-linked atoms (e.g. `Lys` N^{ζ}) by points on a solvent accessible surface[14]. For example, one could find the closest surface point, or a set of "close-enough" such points, reachable from an atom without intersecting the van der Waals spheres of other atoms. While the cross-linked atoms have considerable mobility in solution, we assume that they are fixed for these algorithms. (Dynamics may be accounted for by applying the algorithms to multiple conformations.) We also assume the cross-linker is infinitely flexible. Alternatives will be addressed in a separate publication. With this representation, cross-linkability is determined by testing whether or not the distance between cross-linking sites, measured exterior to the protein, is short enough for the cross-linking molecule. Fig. 2 formalizes the problem and terminology.

The basic protein surface representation we employ is a triangulation of the solvent accessible surface, where vertices indicate locations of a probe molecule's center (typically water) when in contact with the protein, and edges connect triangle vertices. In order to allow for uncertainty in the atomic coordinates of models, we have found it desirable to ignore part or all of the protein side chains. For example, C^{α} coordinates, as employed by Young[3], completely ignore side chains, while C^{β} coordinates ignore many atoms but retain the side chain direction. We have developed an iterative "peeling" algorithm to remove exposed side chain atoms while leaving internal ones intact so that no voids are introduced. The algorithm first identifies solvent accessible residues (with solvent accessible area above some threshold), and then removes those

side chain atoms that are solvent accessible, starting from the end and moving towards the C^{α} in subsequent iterations. This approach guarantees that, upon termination, all and only the outer atoms are removed.

The problem of computing cross-linking distance requires finding the shortest path between two points. This is a well-studied problem in graph theory and networks (e.g. Dijkstra's algorithm [15]). The complexity of geometric shortest path algorithms (e.g. for robotics) grows rapidly with the dimension. Our cross-linking problem can be viewed as finding the shortest obstacle-avoiding path, treating the protein body as an obstacle. When the path is not constrained to a discrete graph, but can include bends, the number of combinatorially different paths becomes exponential. Several approximation algorithms for finding the shortest path have been developed [16].

Here we specialize the shortest path problem to take into consideration the special geometry of proteins. We obtain a hierarchy of novel lower- and upper-bound algorithms for estimating cross-linking distance. Due to space constraints, we present here only high-level pseudocode (Fig. 3), examples (Fig. 4), and sketches of some correctness and complexity arguments.

2.2 Lower Bound Algorithms

The Euclidean distance $d(p_i, p_j)$ between cross-linking sites provides an obvious lower bound, D_{line}, on cross-linking distance. This straight-line approach does not account for the model's surface geometry, and provides relatively little information, but has been employed for model discrimination by Young *et al.* [3]

A tighter bound is obtained by sampling cross-sections of the protein at points along the segment connecting cross-link sites. Our *disk algorithm* (Figs. 3, 4a) computes a lower bound D_{disk} by sampling a set C of points on the $\overline{p_i p_j}$ segment and in S_{int}, and then constructing a sequence of disks with centers in C perpendicular to $\overline{p_i p_j}$ and contained entirely within the body $S \cup S_{\mathrm{int}}$ (they intersect the protein surface only by their boundary circles). The convex hull of the union of the disks and endpoints captures some of the essential surface geometry and provides for immediate computation of a lower bound path. The distance from one site to the other is measured along a path in the intersection of the boundary of the convex hull with a plane containing the segment $\overline{p_i p_j}$.

$D_{\mathrm{disk}}(p_i, p_j)$ depends on the sample points C, which we treat as fixed for the following arguments. For all p_i, p_j, $D_{\mathrm{line}}(p_i, p_j) \leq D_{\mathrm{disk}}(p_i, p_j)$ because the length of each path from p_i to p_j is at least the Euclidean distance. For all p_i, p_j, $D_{\mathrm{disk}}(p_i, p_j) \leq D_*(p_i, p_j)$) follows from the fact that if the length of a path P from p_i to p_j is less than $D_{\mathrm{disk}}(p_i, p_j)$, then P intersects the interior of at least one of the disks. Thus, if there exists a cross-linking path P_* with $|P_*| = D_*(p_i, p_j) < D_{\mathrm{disk}}(p_i, p_j)$, then P_* contains an interior point of at least one of the disks. By construction, each interior

DiskDistance (S, p_i, p_j)
  $C \leftarrow$ a set of sample points on $[p_i, p_j]$ in S_{int}
  $D \leftarrow \{(d(c, p_i), r) \mid c \in C,\ r = \min\{d(c, p) \mid p \in S, \overline{p_i p_j} \perp \overline{cp}\}\}$
  $b \leftarrow (0, 0); e \leftarrow (d(p_i, p_j), 0)$
  $H \leftarrow$ vertices of convex hull of $D \cup \{b, e\}$, sorted from b to e
  return $\sum_{p=0}^{|H|-1} d(H_p, H_{p+1})$

PlaneDistance (S, p_i, p_j)
  $C \leftarrow$ a set of sample points on $[p_i, p_j]$ in S_{int}
  $\Theta \leftarrow$ a set of sample plane normals not perpendicular to $\overline{p_i p_j}$
  return $\max_{c \in C}(\max_{\theta \in \Theta}(\min\{d(p_i, p) + d(p, p_j) \mid p \in S \cap \text{plane}(c, \theta)\}))$

ShortcutDistance (S, p_i, p_j)
  $\mathcal{P} \leftarrow$ a set of sample paths on graph of S, from p_i to p_j
  for each $P = \langle p_i = v_1, v_2, \ldots, v_n = p_j \rangle \in \mathcal{P}$
    $G_P \leftarrow (V, E) : V = \{v_1, \ldots, v_n\},\ E = \{\{v_k, v_l\} \mid \overline{v_k v_l} \cap S_{\text{int}} = \emptyset\}$
    $d_P \leftarrow$ length of shortest p_i to p_j path on G_P
  return $\min_{P \in \mathcal{P}} d_P$

VisibilityDistance (S, p_i, p_j)
  $G \leftarrow (V, E) : V =$ vertices of , $E = \{\{v_k, v_l\} \mid \overline{v_k v_l} \cap S_{\text{int}} = \emptyset\}$
  return length of shortest p_i to p_j path on G

Figure 3: Cross-linking distance bounding algorithms.

point of each of the disks belongs to S_{int}, so P_* intersects S_{int}, a contradiction.

The complexity of the disk algorithm depends on the implementation of the various geometric tests. Selecting sample points requires testing inside/outside of the polyhedral surface, and determining disk radii requires finding distances to surface points on the perpendicular. We employ a straightforward inside/outside test counting the number of intersections of a ray from the sample point with the triangles of the protein surface, requiring total $O(CT)$ time, where T is the set of triangles of S. We compute disk radii by first sorting surface vertices in order along the segment $\overline{p_i p_j}$, and then for each sample point, using binary search to find vertices of triangles that potentially intersect the disk at the sample point. This requires output-sensitive time $O(CT_C \log T)$, where T_C is the set of triangles found by the search. We note that if a very finely sampled set of points is desired (trading off increased complexity for increased accuracy), a plane sweep algorithm could be employed, keeping track of surface triangles intersecting the current plane and iterating by vertices in order of

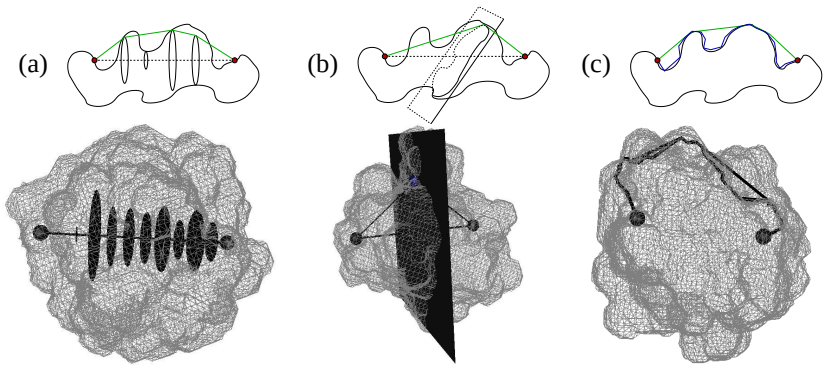

Figure 4: 2D schematics and examples on protein FGF-2 for (a) disk, (b) plane, and (c) shortcut algorithms.

their projections onto $\overline{p_i p_j}$.

A complementary lower bound, D_{plane}, considers single cross-sections at multiple angles and positions. Our *plane algorithm* (Figs. 3, 4b) employs this idea to compute a lower bound D_{plane} by finding, at each sample point and each admissible plane orientation, the shortest path from one cross-link site to the other via a point on the intersection of the plane and the protein surface. The longest such path determines the lower bound. Correctness of the plane algorithm follows from the fact that the cross-linking path must pass through each such plane without intersecting S_{int}.

The complexity analysis for the plane algorithm is similar to that for the disk algorithm. The disk algorithm considers the sample points simultaneously, at a uniform cross-section angle, while the plane algorithm considers the sample points independently, at variable angles. Both the lower bounds and the computational complexity of these algorithms depend not only on S, p_i, p_j, but also on the sample points (and for plane, sample normal directions). The two degrees of freedom sampled for the plane orientations result in more intersection tests than are required for the disk algorithm.

2.3 Upper Bound Algorithms

An immediate upper bound on the cross-linking distance is obtained by taking the convex hull of the protein surface, finding paths outside S_{int} from the cross-linking sites to representative points on the surface of the hull, and finding shortest paths on the hull surface between these points. The correctness of the upper bound D_{hull} computed by this *hull algorithm* follows immediately, since the hull is exterior to the protein. D_{hull} depends on the paths from the sites to the hull surface, and is useful when the computation of these paths is easy (e.g. a line segment not intersecting S_{int} can be identified). By applying Chen and Han's [17] single-source shortest-paths

algorithm for polyhedral surfaces, the complexity for a single site p_i to all other $p_j \in P$ is $O(V^2)$, where V is the set of hull vertices.

The convex hull approach takes "shortcuts" across the mouth of concavities by traversing the hull of the protein, but can miss shortcuts through the concavities. A complementary approach is to start with a sample of paths on the protein surface, rather than on the hull, and then take shortcuts where possible to reduce the lengths of these paths. More precisely, a shortcut of a path replaces the subsequence of vertices $\langle p_k, p_{k+1}, \ldots, p_l \rangle$ with the sequence $\langle p_k, p_l \rangle$ when the segment $\overline{p_k p_l}$ doesn't intersect S_{int}. We call such a pair p_k, p_l a *visible* pair. Our *shortcut algorithm* (Figs. 3, 4c) applies this approach to compute an upper bound D_{shortcut}. Since initial paths are on the surface and shortcuts do not penetrate the body, this is a correct upper bound.

The complexity of the shortcut algorithm depends on the approaches to generating paths, computing visibility, and selecting shortcuts. Our current implementation generates diverse paths by repeatedly performing a breadth-first search from p_i to p_j (taking time linear in the number of surface vertices) and removing edges for path vertices before the next iteration. Other approaches are also possible to achieve diversity. We shortcut a path by an iterative greedy refinement algorithm, starting at p_i and at each iteration jumping to the vertex furthest in the path and still visible. Visibility can be tested by computing surface triangle intersections, as discussed regarding the disk algorithm, yielding $O(TP^2)$ total time to shortcut a path P. An alternate approach that we are exploring is to test intersection of a segment with each of the protein atom spheres, using an atomic radius expanded by that of the solvent. In either case, efficient data structures could reduce the number of triangles tested. Dijkstra's single-source shortest path algorithm[15] could be employed instead of the greedy shortcutting, requiring $O(TP^2)$ time to guarantee optimal shortcutting. We find that in practice the greedy approach usually makes substantial progress per iteration and is closer to linear than quadratic in path length.

Rather than considering shortcuts on a few sample paths, we can compute, at the cost of complexity, a complete visibility graph for the protein surface. A visibility graph[18] indicates all visible pairs of vertices. Given a visibility graph, we can apply standard shortest paths algorithms (e.g. Dijkstra's algorithm[15]). Our *visibility algorithm* (Fig. 3) uses this approach to compute an upper bound $D_{\mathrm{visibility}}$. As with the shortcut algorithm, correctness as an upper bound is immediate.

A straightforward construction of the visibility graph, using the techniques mentioned above for shortcutting, requires $O(TV^2)$ time, where T and V are respectively the set of triangles and vertices of S. This preprocessing is used for all cross-linking site pairs; Dijkstra's algorithm then requires additional $O(V^2)$ time for each site.

2.4 Protein Model Discrimination

In order to discriminate among a set of predicted protein models, we must test for each of them the feasibility of the distances for all observed cross-links. We note that less information can be gained from the absence of evidence for a cross-link under a bottom-up mass spectrometry approach, since several factors other than cross-linking distance can contribute to the absence. More powerful reasoning from negative evidence will be possible in future work, particularly following the application of top-down mass spectrometry for cross-linking analysis[4].

When employed with observed cross-links, lower and upper bounds provide complementary information for model discrimination. A lower bound can provide evidence against a model, when the estimated distance for an observed cross-link exceeds the expectation for the cross-linker. An upper bound can provide evidence for a model, when the estimated distance for an observed cross-link is less than the maximum distance. We adopt a simple strategy assuming cross-links are independent and sum their scores: $+1$ when an upper bound is satisfied, -1 when a lower bound is violated, and 0 when neither holds. (It is impossible for both to hold.)

3 Results

We have tested the performance of our algorithms for model selection with both published experimental and simulated data. Fibroblast growth factor (FGF-2) is the primary target because of available data[3] and structure (PDB id 4FGF). Competing models were obtained for the published template structures[3] via the protein fold-recognition meta-server[19]; two of the models are of the same fold (β trefoil) as 4FGF. The `Lys`-specific cross-linker BS^3 was used. To further demonstrate the utility of our approach, we chose two CASP4[20] targets with many high-quality models: deoxyribonucleoside kinase (PDB id 1J90) and α-catenin (PDB id 1L7C).

We applied our algorithms, using N^{ζ}, C^{γ}, C^{β}, or C^{α} atoms (with surfaces appropriately peeled), and found the C^{β} to provide the best results. The C^{α} straight-line measurement of Young *et al.*[3] provides a control, although we could not exactly reproduce their model discrimination results (presumably due to differences in the details of the protein models).

Visualizations like those in Fig. 4 provide evidence of the ability of our algorithms to better approximate cross-linking distance. To quantitatively characterize discriminatory power, we computed, for each distance between 1 and 45 Å, the number of possible `Lys` pairs in 4FGF whose length exceeds the threshold and compared the number for experimentally identified cross-links (to be maximized) and unidentified ones (to be minimized). Greater difference between these numbers at a threshold indicates better abstraction of structural features and enhanced ability of the method

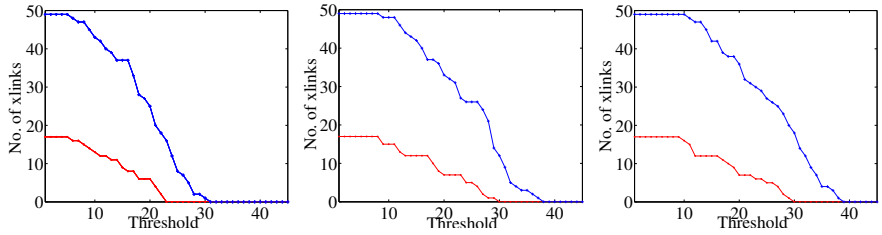

Figure 5: Comparison of cross-linking distances for (left) C^α straight-line, (middle) C^β disk, and (right) C^β plane methods. The x-axis indicates a distance and the y-axis the number of experimentally-identified (blue lower line; 18 maximum) and not (red upper line; 48 maximum) cross-links exceeding that threshold.

employed to separate identified from unidentified for a cross-linker of that length. Fig. 5 compares the straight-line distance against two of our lower bound methods. The area between the curves (summing the count difference over the range) is 641 for C^α straight-line, 826 for C^β disk, and 887 for C^β plane, demonstrating the more informative bounds provided by our algorithms.

In model discrimination, Young *et al.*[3] employ a maximum value of 24 Å for feasible cross-linking distance; we use the same threshold for testing both upper and lower bounds. This value accounts for the BS^3 length (11.4 Å), the distance from the reactive N^ζ to the representative cross-linking site, and a small amount of uncertainty. Fig. 5 shows that some of the experimentally-determined cross-links have distances exceeding even this threshold (e.g. D_{disk}(`Lys21`,`Lys125`) is 29.5 Å). These large distances were confirmed visually. Possible explanations include experimental errors, artificial distortion of the protein, or extensive natural flexibility. Artificial distortion (e.g. by partial denaturation due to multiple cross-links), may be alleviated by better choice of experimental conditions. The work of Falke[21] suggests it is possible to obtain cross-links more than 10 Å longer than expected, in mobile situations, although the rate of cross-linking falls off by orders of magnitude. To study such flexibility, we intend to apply our algorithms to multiple frames of a molecular dynamics simulation, boosting the need to trade off efficiency and tightness of bound. We note that infrequent conformations might in general be detected rarely by mass spectrometry, and thus could be treated as noise in a probabilistic analysis. The cross-link experiment could also be altered to exploit differences in rates.

We further quantified discriminatory power by comparing differences in estimated cross-link distances between models. Treat the set of cross-linking distances for a model as a point in ℓ-dimensional space (for ℓ cross-links), and compute differences (Euclidean distance) between these points. A larger difference is indicative of greater discriminatory power, since the cross-linker's fixed length is more likely to separate the points on some dimension (cross-link). We compared our disk C^β

algorithm to the control straight-line C^{α}, and found that our algorithm yields an average of 0.2–0.3 Å larger average differences for both experimentally observed and all possible cross-links, when either comparing 4FGF to all other models, 4FGF to non-β-trefoil models, or each model to all other models.

We tested our methods by ranking the correct structure vs. the models, scoring with either the Young approach of counting violations (straight-line distance > 24 Å) or our discrimination method combining disk (lower bound) and shortcut (upper bound) distances. We analyzed the effects of cross-link sparsity and noise by choosing datasets consisting of a random subset of the identified plus a random set of the unidentified cross-links. Fig. 6 illustrates the average rank of the correct structure over 100 such simulations for each of several different numbers of observed and unobserved cross-links. (We apply the conservative choice of ranking the correct structure worst in case of a tie.) With smaller subsets of identified cross-links, the two methods are comparable. Larger subsets tend to include more cross-links labeled infeasible by the disk bound, and our method degrades.

Finally, we analyzed model discriminability by varying the number of simulated "good" and "bad" cross-links and finding the average rank of the correct structure as above. For tests with our method, good cross-links were chosen from those with shortcut C^{β} distance below 24 Å in the correct structure, and bad cross-links from those with disk C^{β} distance greater than 24 Å. Similarly, good and bad cross-links for the straight-line method were chosen using the 24 Å threshold. Fig. 7 shows results for FGF using each method to analyze the corresponding simulated dataset. These results test discriminability and robustness to sparsity and noise — over many different sets of feasible/infeasible cross-links, our distances distinguish the correct structure from the models better than do straight-line distances. Fig. 8 shows our results on the CASP4 targets; straight-line is again inferior (not shown).

4 Conclusions

We have developed and applied a set of lower- and upper-bound algorithms for estimating cross-linking distance. The algorithms trade off complexity and tightness of bound. We have shown that by taking into account protein surface geometry, our algorithms provide better model discriminability, in terms of cross-link separability, distance differences, and discrimination effectiveness. We illustrated the robustness of our techniques by simulating sets of good and bad cross-link data. Our results demonstrate that information from relatively rapid and inexpensive experiments permit model discrimination in spite of sparse information and the presence of noise.

The current work can be further extended in several ways. Protein dynamics can be taken into consideration. As more experimental data become available, better classifiers can be developed to apply distance estimates to model discrimination. While

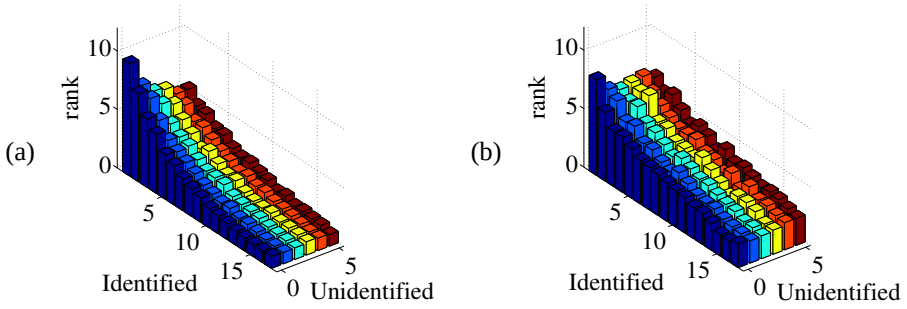

Figure 6: Discrimination using experimental data for FGF-2 with (a) straight-line C^{α}, (b) combined disk and shortcut C^{β}. The x- and y-axes indicate number of cross-link pairs identified and unidentified, respectively; the z-axis shows the average rank of the actual structure over 100 random subsets.

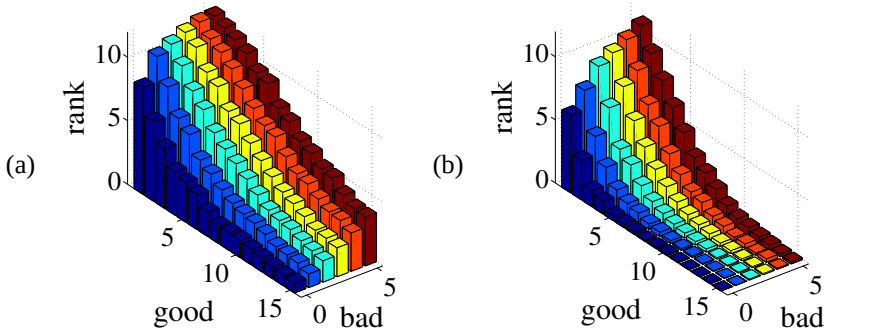

Figure 7: Discriminability for FGF-2 with (a) straight-line C^{α}, (b) combined disk and shortcut C^{β}. The x- and y-axes indicate number of good and bad cross-link pairs, respectively, chosen according to the same methods; the z-axis shows the average rank of the actual structure over 100 random subsets.

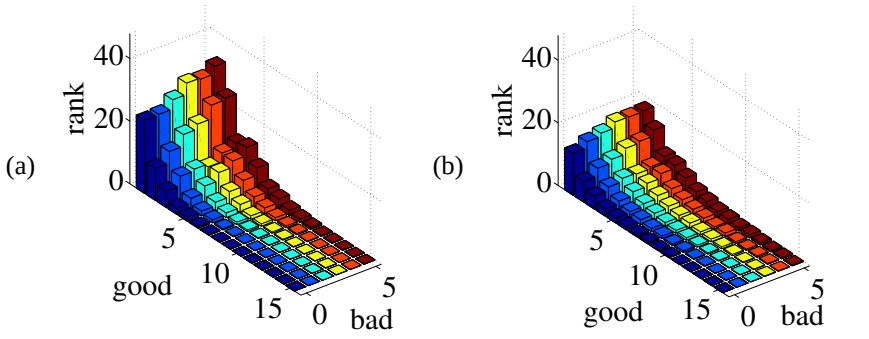

Figure 8: Discriminability, as in Fig. 7, with combined disk-shortcut C^{β} using simulated data for (a) deoxyribonucleoside kinase and (b) α-catenin models.

cross-links were considered independent here, a more complex framework would capture dependencies with respect to differential reactivity, competing cross-links, and so forth. Our analysis can be used in planning experiments, e.g. proposing a cross-linker of the best length or the substitution of particular residues to lysine.

Acknowledgments

This work is supported in part by a US NSF CAREER award to CBK (IIS-0237654); and EMBO/HHMI Young Investigator and Foundation for Polish Science Young Scholar award to JMB. Thanks to Mike Stoppelman, Xiaoduan Ye, and other members of our labs for helpful discussions and related work.

References

1. Natl. Inst. Gen. Med. Sci. `http://www.structuralgenomics.org`.
2. M. Haniu, L. O. Narhi, T. Arakawa, S. Elliott, and M. F. Rohde. *Protein Sci*, 9:1441–51, 1993.
3. M.M. Young et al. *PNAS*, 97:5802–5806, 2000.
4. G. H. Kruppa, J. Schoeniger, and M. M. Young. *Rapid Commun Mass Spectrom*, 17(2):155–62, 2003.
5. A. Scaloni et al. *J Mol Biol*, 277:945–958, 1998.
6. J. B. Swaney. *Methods Enzymol*, 128:613–626, 1986.
7. I. Kwaw, J. Sun, and H. R. Kaback. *Biochemistry*, 39:3134–3140, 2000.
8. J. Skolnick, A. Kolinski, and A. R. Ortiz. *J Mol Biol*, 265:217–241, 1997.
9. P. M. Bowers, C. E. M. Strauss, and D. Baker. *J Biomol NMR*, 18:311–318, 2000.
10. S. Elliott et al. *Blood*, 87(7):2702–13, 1996.
11. A. Bohm et al. *J Biol Chem*, 277(5):3708–17, 2002.
12. F. Zappacosta et al. *Protein Sci*, 6(9):1901–9, 1997.
13. W. Zheng and S. Doniach. *J Mol Biol*, 316:173–87, 2002.
14. B. Lee and F. M. Richards. *J Mol Biol*, 55(3):379–400, 1971.
15. E. W. Dijkstra. *Numerische Mathematik*, 1:269–271, 1959.
16. J. S. B. Mitchell. *Geometric shortest paths and network optimization*. Handbook of Computational Geometry, 2000.
17. J. Chen and Y. Han. In *Proc ACM Symp Comp Geom*, pp. 360–369, 1990.
18. J.C. Latombe. *Robot Motion Planning*. Kluwer, 1991.
19. M.A. Kurowski and J.M. Bujnicki. *Nucleic Acids Res*, 31(13):3305–7, 2003. `http://genesilico.pl/meta`.
20. J. Moult et al. *Proteins*, S5:2–7, 2001.
21. C. L. Careaga and J. J. Falke. *J Mol Biol*, 226:1219–35, 1992.