Efficient Computational Algorithms for Docking, and for Generating and Matching a Library of Functional Epitopes II. Computer Vision-Based Techniques for the Generation and Utilization of Functional Epitopes

Ruth Nussinov[1,2]*and Haim J. Wolfson[3]

[1] Intramural Research Support Program - SAIC
National Cancer Institute
Frederick Cancer Research and Development Center
Laboratory of Experimental and Computational Biology
Bldg 469, Rm 151
Frederick, MD 21702

[2] Sackler Inst. of Molecular Medicine
Sackler School of Medicine
Tel Aviv University, Tel Aviv 69978, Israel

[3] Computer Science Department
School of Mathematical Sciences
Tel Aviv University, Tel Aviv 69978, Israel

November 22, 1999

A running title: Generation of functional epitopes

*correspondence should be addressed to R. Nussinov at NCI-FCRF Bldg 469, rm 151, Frederick, MD 21702. TEL: 301-846-5579, FAX: 301-846-5598, email: ruthn@ncifcrf.gov.

Abstract

This is the second review in a two-review series. In the first review (1) we described the computational complexity involved in the docking of a ligand onto a receptor surface. In particular, we focused on efficient algorithms designed to handle this computational task. Such a procedure results in a large number of potential, geometrically feasible solutions. The difficulty is to pinpoint which of these is the more likely candidate. While there exists a number of approaches to rank these solutions according to different criteria, such as the size of the interface or some approximation of potential energy, none of the existing methods has been shown to be consistently successful in this endeavor. If the binding site is unknown *a priori*, the magnitude of the task is awesome. Here we propose one way of dealing with this problem, i.e., *via* derivation and utilization of binding epitopes. If a library of such epitopes is available, particularly for a large number of protein families, it can be used as more likely binding sites for a given ligand.

We describe an efficient, computer-vision based method to construct binding epitopes. We focus on two ways through which such a library can be generated, (*i*) molecular surface-based, or (*ii*) residue-based. Alternatively, the two can be combined. We further describe how such a library can be used efficiently in the matching/docking procedure.

Keywords: Functional epitopes; matching; efficient algorithms; library of epitopes

Introduction

While the definitions of binding epitopes vary, in general the term refers to a recurring pattern of molecular surface, in a family (or, families) of proteins, and where at least in one family member the site is known to be a binding site. Clearly, a similar binding epitope can be found across family boundaries too. Hence, to construct a library of binding epitopes we start by picking a known site as a *pattern*, and proceed to search for similar patches of surface, or of arrangement of residues at/near the surface, in other molecules.

Detection of *a-priori* unknown active (binding) sites on enzyme or receptor surfaces is extremely important. There are a number of reasons why this task is critical to carry out. First, identification of active (binding) sites is the initial step in the design of inhibitors, and drugs (2). Second, being able to locate active sites on protein surfaces, enables their genetic modification, enhancing their activity toward specific, predetermined ligands and functions. Knowledge of the location of the binding site enables focusing on this location from the start in the docking simulations. Alternatively, one may utilize it to filter and rank the obtained solutions at a later stage in the docking procedure. While the former is a more efficient way of handling the problem, one may nevertheless wish to obtain alternate geometrically feasible configurations as well.

There are a number of ways to locate an unknown active site on a known protein structure. In an insightful review, Ringe (3) surveyed approaches to the definition of "what makes a binding site a binding site". She has presented some guidelines, and proposed that binding sites are generally depressions in the protein surface, "in which there is greater than average degree of exposure of hydrophobic groups". These generally contain disordered, easily displaced water molecules. She has further proposed that conformationally flexible residues at the binding site may be expected to be particularly useful in replacing the disordered water by the competing ligand. Laskowski et al. (4) have explored thoroughly the active sites. Their extensive analysis indicates that active sites of enzymes are characterized primarily by large clefts. Their analysis illustrates that in the vast majority of the single chain enzymes that they have examined, the ligand binds in the largest cleft. Thus, the active sites of enzymes can be frequently identified using geometrical criteria alone. Peters et al. (5) have utilized alpha-shapes (6), an attractive computational geometry tool, in an automated search for ligand binding sites on protein surfaces. This, strictly geometry-based approach has also found a correlation between deep clefts and enzyme active sites. Additionally, a correlation between patches of hydrophobic surfaces and binding sites has been noted (e.g., ref. 7–9). The results of Peters et al. (5), and of Laskowski et al. (4), have further confirmed an older observation (10) that protein–protein binding sites are characterized by more shallow, flat surfaces, unlike the enzyme active sites.

The approaches described above characterize general properties of active (and, binding) sites. On the other hand, here we seek sites on protein surfaces which resemble specific, known active sites. Detection of such sites has several important implications and applications. Molecules with similar surface motifs frequently have similar biological features. The converse also holds, namely, local surface similarity in different proteins can lead to conclusions concerning their (similar) biological functions. The function of a protein may be elucidated from its binding site. Consequently, a comparison of binding sites of proteins should, in principle, allow a study of their mechanisms and of their interactions. Homology of a potential binding site to a known one, would also facilitate its engineered change to further optimize it, and to rationally engineer inhibitors/drugs (2).

In practice, a straightforward approach to detect an active site in proteins if their 3D structures are available, is to pick a known active site of a (model) protein, and use it as a template, searching for a similar one in the target protein(s). This can be done in two ways: either by initially describing the molecular surfaces of the proteins, and picking the points which faithfully represent the surfaces, or alternatively, using atomic coordinates. In the latter case we may either choose the coordinates of surface atoms, or of, say, C_α atoms. These C_α atoms may again be either those whose atoms line the molecular surface, or those of functionally important residues, such as the ones belonging to the catalytic triad of the serine proteins. It should be noted that such search methods have to be flexible enough to allow inexact partial matching of the template to the investigated target molecule.

We have already illustrated these approaches systematically (11), by considering the following examples. We started by describing the molecular surfaces of two groups of proteins: the serine proteases and the chorismate mutases. Within each group, members of the subgroups are compared between themselves and across the subgroup boundaries. Thus, we compared very similar structures (and, surfaces), within the subgroups, which become progressively less similar as we compare members between the subgroups. Every comparison between two structures is carried out several times, separately employing: (*i*) the geometrically described molecular surfaces, either in their entirety and/or just the active site regions; (*ii*) the coordinates of all the atoms belonging to the active sites; and (*iii*) the coordinates of the C_α atoms, either all C_α's of the entire proteins, and/or of the active site regions. (*iv*) In the serine proteases, the active sites are known. In the case of the chorismate mutases, some are known, and others (e.g., the yeast chorismate mutase) have been suggested via modeling (12). Within the groups, we compare the more, and the less, similar members. In particular, (*v*) the groups differ: In the serine proteases the catalytic triads consist of Asp, Ser and His. Not all of these residues (or, all the atoms belonging to these residues) are always on the enzymes' surfaces. On the other hand, residues involved in catalysis in the chorismate mutases are believed to be on the enzymes' surfaces.

A critical issue which should be born in mind is molecular flexibility, and, in particular, surface variability (13). Molecular surfaces are flexible, and, active site regions may be particularly so. Indeed, flexibility may be particularly advantageous for proteins with a broader range of ligands/substrates. The less specific the receptor-ligand (enzyme-substrate) interactions, and the broader the range of binding, the more flexible the binding site may be expected to be. Yet, accounting for flexibility in surface comparison routines is likely to constitute an especially difficult task. Moreover, enzymes, and proteins having a broad range of bound substrates/ligands, may also make considerable use of water molecules mediating their bound molecular counterparts (e.g., 3, 14, 15). Ladbury (15) has recently outlined the rationale, and the supportive evidence, arguing that water are likely to have a dual role in binding: it may increase the promiscuity of the binding, however it may also increase the specificity and affinity (Ladbury and references therein). Bridging water molecules are normally not accounted for in molecular surface matching techniques either, owing to the difficulty in predicting their locations on the protein surfaces. A further complication is derived from the fact that it is difficult to pinpoint which residues are critical for binding. The fact that a given residue is in contact with its cognate ligand, and that it recurs in a number of different members of the family, does not necessarily imply that it is critical for binding. There is a growing body of examples showing that mutating such residues still enables binding. What happens is that owing to side-chain flexibility, nearby side-chains move, taking over the role played by the residue which was originally at that spatial location. Hence, a binding epitope should always be taken with a grain of salt, bearing in mind all of these unknown factors. Nevertheless, as an initial guess, especially when some sloppiness in the requirements is allowed, it is useful to have.

An efficient matching algorithm: The Geometric Hashing

The computer-vision based matching algorithm we briefly describe here is uniquely suitable to carrying out such active site similarity comparisons. It is independent of the residue order on the protein chain. Hence, it can handle comparisons of active sites, which frequently consist of isolated, unconnected residues. The critical feature is that while their order in different enzymes may differ, they can nevertheless still be arranged in similar spatial configurations. Moreover, since this algorithm is order-independent, it can compare both C_α atoms, as well as surface atoms and points describing molecular surfaces. In the latter two cases, there is clearly no order which is involved. This algorithm views molecules, or their surface descriptions, as collections of points in 3D space. Using analogous procedures for matching surface point coordinates and for matching atoms, whether C_α or side-chain, contributes to make the comparisons of these datasets consistent.

Several techniques for protein structure comparison have been developed during the last decade (e.g., ref. 16–24). These fall into three categories: (*i*) The first includes techniques based on dynamic programming. These involve an adaptation of the Needleman-Wunsch type algorithm to matching of 3D structures (for example, ref. 17). (*ii*) The second involves matching 3D structures of fragments of contiguous amino acids (16,19). Fragments whose superpositioning is achieved *via* similar transformations are clustered, yielding a complete 3D structural match. (*iii*) The third approach utilizes algorithms derived from Computer Vision (25). These view protein structures as collections of points (atoms) in 3D space, seeking to optimally match the largest subset of these points (22–24, 26). Each of these methods has advantages and disadvantages. Dynamic programming techniques are fast. As they match the inter- C_α-atom distances between any two residues in the two proteins, they tolerate insertions and deletions. However, the order of the residues is strictly conserved. The second approach can match fragments whose order in the chains may differ; On the other hand, it is unable to match single, isolated residues, and it is unable to handle insertions or deletions within the fragments. The third approach does not suffer from either of these deficiencies: namely, the order of the residues in the chain is immaterial, and deletions and insertions are tolerated as well. Both of these advantages are the straightforward outcome of the disregard for the connectivity of the amino acids in the polypeptide chain. In Computer Vision, we match unordered pixels, with connectivity being entirely meaningless (25). Furthermore, while it appears that giving up the connectivity constraint might result in extremely time consuming algorithms which render the method as non practical, this is not the case. The speed of the Geometric Hashing is impressive. Comparison of the structures of two proteins of average size (say, about 220 amino acids each) takes about 1 second on a Silicon Graphics workstation.

Below we first describe the algorithm in a concise form for the comparison of the critical points. The same procedure is used for the comparison of the surface atom centers. We next proceed to describe the Geometric Hashing for the comparisons of C_α atoms, in the generation of epitopes from protein interfaces. Fig. 1 provides a flow chart of the algorithm, for further clarity.

Using the Geometric Hashing to derive epitopes *via* surface matching

The choice of the type of description of the molecular shape is critical for a comparison. A variety of methods have been developed to represent molecular surfaces (e.g., ref. 6, 27, 28). The solvent accessible surface description of Richards (29) as represented by the Connolly surface dots (30, 31) is often used as a basis for concise shape description (e.g., 32–35). The main requirements for a method is to accurately describe the molecular surface features, be concise and be convenient to use in a matching technique. The method that we utilize here describes the molecular surfaces in terms of *critical points*. These critical points are accurately positioned at the peaks of hills and at the bottom of the valleys (35, 35).

Given the surfaces of two molecules as defined by their critical points, our goal is to find a rigid transformation (rotation and translation) which would superimpose one molecular surface onto another, in a way such that a sufficient number of the surface critical points of the two molecules will be matched. Here, we refer to one of the molecules whose surface is compared as the *model*, and the other as the *target*. To do that we utilize the Geometric Hashing (e.g., 22–24, 26) (See Fig. 1). The basic idea of the Geometric Hashing method is that of a representation of an object by storing redundant, transformation-invariant, information in a hash table. The matching algorithm consists of two stages, *preprocessing* and *recognition*. The *preprocessing* stage can be executed off-line, and only once. In this stage, the *transformation invariant* features are extracted from the model. They are saved in a hash table. This is like a learning stage. In the *recognition* stage, similar transformation invariant features are computed for the target. These are used to access the hash table in order to find possible instances of the model. We construct the transformation invariant representation for the molecular surface by representing each critical point in each admissible reference frame. This representation contains redundant information, which enables matching unconnected surface motifs. We use a Cartesian coordinate system as a reference frame. The transformation invariant features are the coordinates of the surface critical points which are computed in this frame. Two reference frames, from the model and from the target proteins, are matched if a "substantial enough" number of coordinates are similar, and if additional constraints, which ensure quality matches, are met. A reference frame is built for each ordered pair of critical points satisfying the criteria (noted briefly below), and the mean direction of their normals. The application of the Geometric Hashing to active sites *via* surface molecular comparisons is illustrated in the flow chart of Figure 2 (11).

Reducing the complexity. Since the points are represented in many reference frames, the transformation invariant representation of a rigid body used in the Geometric Hashing is redundant. This enables recognition of different, unconnected surface motifs. In practice, it is sufficient to explore only a fraction of the reference frames and their invariants. This already guarantees recovering a correct solution. We consider a reference frame only if the distance between the surface points defining it, is within the specified constraints. Similar constraints are imposed on the transformation invariants under a given reference frame. The *maximal distance* constraint is based on a *proximity heuristic*. The proximity heuristic assumes that a match between two objects contains relatively dense regions of matching points. When we apply the proximity consideration, the geometric hashing detects only matches between *local* surface patches. These necessarily include only nearby points, and do not cover the entire molecular surface. Such matches are referred to as *seed matches*. A transformation between local surface patches defined by a seed match is straightforwardly computed. Applying this transformation, the seed matches are extended in the next step, to cover other areas of the molecular surface. Note, that minimal distance constraints are employed as well to ensure numerical stability of the geometric calculations.

Constructing a reference frame. A reference frame is constructed for each ordered pair of critical points which satisfy the specified criteria. We define a Cartesian reference frame unambiguously using two points and one direction, if the line, connecting the points, and the direction are non-collinear and non-parallel. This direction is defined by the mean of the surface normal directions at the two critical points. Owing to surface variability, this procedure makes the calculation more robust.

Preprocessing stage. For each close enough pair of *model* critical points, we define a reference frame. For each critical point which is located not too far from the points which define the reference frame, we calculate the coordinates relative to this reference frame. These save as an address to a hash table, where the reference frame information is saved.

Recognition Stage. For each two points of the target which are located not too far apart, we again define a reference frame. For each critical point which is close enough, we calculate its coordinates in this reference frame. To find matching model critical points, we access the hash table at the address which is defined by the coordinates. In practice, all table bins containing an error region around the specified bin location are considered.

Voting constraints. We register a "vote" for a pair of model and target reference frames if the coordinates are within the stipulated error distance.

Obtaining a match. In the recognition stage, pairs of matching reference frames are produced. For each such pair, we define a vote counter and the point-pairs for which the coordinates are matched are stored in a list. If the score of a reference frame-pair exceeds a minimal number of votes, the superposition of these model and target reference frames produce an alignment which contains at least that minimal number of matching surface point and normal pairs from the model and the target. The rotation and translation giving a best (minimal) least squares superposition of the matching point sets is computed. These resulting critical point pairs along with their corresponding transformation are referred to as a *seed match.*

Clustering and extending the seed matches

Clustering. We next cluster seed matches which have similar transformations. Each transformation is defined by 6 parameters: 3 rotational and 3 translational. Initially, we regard each seed match as a cluster. The clustering algorithm proceeds in an iterative manner, joining seed matches by the proximity of their parameters.

Extension. We may obtain additional matching pairs by aligning the transformed model critical points onto the target critical points and comparing the distances between the points and their associated normal directions. Since the critical points describing the protein surface may be rather dense, points can have too many close "neighbors". Thus, a heuristic iterative matching algorithm is used to minimize the sum of the distances between the newly matched pairs.

Finding connected components

When searching for similarity within complete molecular surfaces, we would like to have only solutions which match connected surface patches, rather than points distributed randomly over the surface. To do this we apply a connectivity filter. A point is called a "neighbor" of another point if the distance between them is under a pre-specified threshold. As a match represents two superimposed surface patches which are close to each other, we may apply a connectivity filter only to the surface of one molecule, for example, a target. Assume that each target point of an extended match is a vertex of a graph. An edge connects vertices a and b if a and b are neighbors. Traversing the graph using a breadth first search (BFS) algorithm, we compute all the connected components of this graph. Only large connected components are retained for farther processing.

Using the Geometric Hashing to derive epitopes *via* comparisons of protein interfaces

As pointed above, the Geometric Hashing algorithm is uniquely applicable to carry out structural comparisons which cannot be handled by other, order-dependent approaches. The generation of a dataset of non-redundant protein-protein interfaces (36) is such an application. The availability of such a dataset, where each protein interface represents a family of related interfaces, is essential for the derivation of related binding epitopes. The generation of the dataset requires carrying out structural comparisons between two interfaces, where each interface is composed of at least two protein chains. In particular, only pieces of each of the chains actually constitute an interface. Depending on the definition of an interface, the residues of the first protein which belong to an interface are roughly those whose distances from the residues of the second protein are under pre-defined thresholds. Hence, clearly, an interface may be composed of isolated residues, and of fragments of residues in different order. A structural comparison method which is order-dependent cannot carry out comparisons of interfaces, or comparisons of interfaces with structures of single chains. Here we utilize the Geometric Hashing to generate families of related interfaces, progressively clustering similar ones, choosing a representative, clustering the representatives, lowering the required matching thresholds, and repeating the procedure. Repeating these cycles several times, at the end we are left with the required families, at different levels of similarity. We next proceed to pick up the representative binding interface, compare it with all the members of the family which it represents, analyzing the preserved residues, and in parallel, the variability which is observed within the family. Such an analysis enables detection of the residues which are more likely to be conserved, and those whose existence is not critical to the binding. Even those whose presence appears essential still require testing, as it may be the case that these too may be mutated with no marked effects on the binding. The next step involves picking a representative interface as a *model*, and searching for similar ones in a protein structure whose binding sites, or binding capabilities, are unknown. This is again carried out using the Geometric Hashing.

Nevertheless, the Geometric Hashing technique also suffers from some drawbacks. By far the most prominent one is its sensitivity to random matches. Since single, isolated points (C_α atoms in the protein structure comparison) are matched, such a method is much more prone to chance matches of these point-atoms, when they happen to fall at the same location in 3D space. Furthermore, as the prime determinant in the choice of the matching atom pairs is the distance from each other, the choice that is being made is not always the biologically sound one. We have managed to overcome to a large extent this problem through the incorporation of chain connectivity considerations into the final stages of the matching in the Geometric Hashing and into the scoring, improving the quality of the obtained substructural motifs. Isolated C_α atom matches are still allowed, enabling matchings of active sites. Nevertheless, we give a preference to architectural considerations, alleviating this problem of random C_α matches. Furthermore, it equivalences many of the commonly encountered topological matches, circumventing the difficulty outlined above. The results of the matches are depicted graphically by a graphic routine (36). This routine, originally written to display water clusters (37), has been adapted to such a purpose. The resulting Geometric Hashing, with its graphic capabilities is a useful tool.

Conclusions

Considerable work has been carried out on the interactions, and energetics of protein–protein binding, whether of the entire interface, or focusing on the contributions of single residues. The latter are particularly important for the comprehension of drug-resistant mutations (38, 40-52). The principles that govern the interactions of protein-protein interfaces are still not fully understood. The derivation, and analysis of a dataset of protein–protein interfaces may well provide measures of the driving forces stabilizing protein–protein interactions. Additionally, it may supply some critical information necessary for ligand (inhibitor, drug) design.

The recent explosion in the accumulation and use of co-crystal structures of protein homo- or heterodimers that are relevant therapeutic targets, may enable the design of small molecules with high specificity and affinity, which may conceivably block, or alter, protein–protein interactions. However, this task has been extremely difficult, with few limited successes. Design of compounds that are capable of interfering with protein-protein interactions requires an understanding of the origin of affinity and specificity in the interface.

It has been shown that the free energy of binding is not distributed evenly across interfaces; instead, there are hot spots of binding energy consisting of a small subset of residues in the dimer interface. These hot spots are enriched in tryptophan, tyrosine, and arginine, and are surrounded by a shell of energetically apparently less critical residues. These have been suggested to be important by occluding bulk solvent from the hot spot (38).

Utilizing a method such as the Geometric Hashing, we may carry out a comprehensive and systematic structural analysis of protein–protein interfaces. Since the crystal structures of the families of these interfaces (39) which we have created are geometrically similar, the dataset may be used to analyze structural characteristics of protein–protein interfaces. In particular, it may be used to address questions like, for example, given a geometric similarity among members of a protein–protein interface family, how similar are the binding surfaces? Which residues at the surface of these proteins are conserved? What are the major determinants of the binding (hydrophobicity, electrostatic, etc)? Which amino acids are preferred, avoided, or neutral at each interface? Which are poor interface formers? Addressing these have important implications for both protein-protein binding and protein ligand design. Furthermore, an insight into these basic questions, along with the derived binding epitopes, enables searches for homologous, potential sites in other structures, where the existence/location of these sites are unknown.

Acknowledgment

We thank Dr. J. V. Maizel for discussions, and for encouragement. The research of R. Nussinov has been sponsored by the National Cancer Institute, DHHS, under Contract No. 1-CO-74102 with SAIC. The research of R. Nussinov and H.J. Wolfson in Israel has been supported in part by grant No. 95-00208 from the BSF, Israel, by a grant from the *Israel Science Foundation* administered by the *Israel Academy of Sciences*, by the Magnet fund, administered by the Israel Ministry of Industry, and by a basic science grant administered by Tel Aviv University and the Adams Brain Center grant. The contents of this publication do not necessarily reflect the views or policies of the DHHS, nor does mention of trade names, commercial products, or organization imply endorsement by the U.S. Government. The research of H.J. Wolfson is partially supported by the Hermann Minkowski-Minerva Center for Geometry at Tel Aviv University.

References

1. Nussinov, R.; Wolfson, H.J.; *Combinatorial Chemistry & High Throughput Screening.* **1999**, To be published.

2. Apostolakis, J.; Caflisch, A.; *Combinatorial Chemistry & High Throughput Screening.* **1999**, 2, 91–104.

3. Ringe, D.; *Curr. Opin. Strct. Biol.* **1995**, 5, 825–829.

4. Laskowski, R. A.; Luscombe, N. M.; Swindells, M. B.; Thornton, J. M.; *Prot. Sci.* **1996**, 5, 2438–2452.

5. Peters, K. P.; Fauck, J.; Frommel, C.; *J. Mol. Biol.* **1996**, 256, 201–213.

6. Edelsbrunner, H.; Mucke, E. P.; *ACM Transact. Graph.* **1994**, 13; 43–72.

7. Young L.; Jernigan R. L.; Covell D. G.; *Protein Sci* **1994**, 3, 717–729.

8. Vakser I.A.; Aflalo C.; *Proteins: Struct. Funct. Genet.* **1994**, 20, 320–329.

9. Clackson, T.; Wells, J. A.; *Science* **1995**, 267, 383–386.

10. Connolly, M.L.; *Biopolymers* **1986**, 25, 1229–1247.

11. Rosen, M.; Lin, S. -L.; Wolfson, H.; Nussinov, R; *Protein Eng.* **1998**, 11, 263–278.

12. Xue, Y.; Lipscomb, W. N.; *Proc. Natl. Acad. Sci. USA* **1995**, 92, 10595–10598.

13. Norel R.; Lin S.L.; Xu D.; Wolfson H.J.; Nussinov R. *Structure, Motion, Interaction and Expression of Biological Macromolecules*, Ed. Sarma, R.H. & Sarma, M.H., Eds. Adenine Press, Albany, N.Y., **1998**, 33–51.

14. Bhat, T. N.; Bentley, G. A.; Boulot, G.; Greene, M. I.; Tello, D.; Dall'Acqua, W.; Souchon, H.; Schwarz, F. P.; Mariuzza, R. A.; Poljak, R. J.; *Proc. Natl. Acad. Sci., U.S.A.* **1994**, 91, 1089–1093.

15. Ladbury, J.E.; *Chemistry & Biology* **1996**, 3, 973–980.

16. Alexandrov, N. N.; Takahashi, K.; Go, N.; *J. Mol. Biol.* **1992**, 225, 5–9.

17. Orengo, C. A.; Taylor, W. R.; *J. Theor. Biol.* **1990**, 147, 517–551.

18. Holm, L.; Sander, C.; *J. Mol. Biol.* **1993**, 233, 123–138.

19. Vriend, G.; Sander, C.; *Proteins*, **1991**, 11, 52–58.

20. Subbarao, N.; Haneef, I.; *Prot. Engng.* **1991**, 4, 877–884.

21. Luo, Y.; Lai, L.; Xu, X.; Tang, Y.; *Prot. Eng.* **1993**, 6, 373–376.

22. Nussinov, R.; Wolfson, H. J.; *Proc. Natl. Acad. Sci (USA)* **1991**, 88, 10495–10499.

23. Bachar, O.; Fischer, D.; Nussinov, R.; Wolfson, H.; *Protein Eng.* **1993**, 6, 279–289.

24. Tsai, C. -J.; Lin, S. L.; Wolfson, H.; Nussinov, R.; *Techniques in Protein Chem* **1996**, VII, 419–429.

25. Lamdan, Y.; Schwartz, J. T.; Wolfson, H. J.; In *Proceedings of IEEE International Conference on Robotics and Automation.* Philadelphia, Pennsylvania, April 1988.

26. Fischer, D.; Wolfson, H.; Lin, S. L.; Nussinov, R.; *Protein Science* **1994**, 3, 769–778.

27. Zachmann, C.D.; Heiden, W.; Schlenkrich, M.; Brickmann, J.; *J. Comp. Chem.* **1992**, 13, 76–84.

28. Connolly, M.L.; *Biopolymers* **1992**, 32, 1215–1236.

29. Richards, F.M.; *Ann. Rev. Biophys. Bioeng.* **1977**, 6, 151–176.

30. Connolly, M.L.; Ph. D. thesis, Univ. of California at Berkeley. **1981**.

31. Connolly, M.L.; *J. Appl. Cryst.* **1983**, 16, 548–558.

32. Jiang, F.; Kim, S. H.; *J. Mol. Biol.* **1991**, 219, 79–102.

33. Shoichet, B.K.; Kuntz, I.D.; *J. Mol. Biol.* **1991**, 221, 79–102.

34. Lin, S.L.; Nussinov, R.; Fischer, D.; Wolfson, H.L.; *Proteins: Struct. Funct. Genet.* **1994**, 18, 94–101.

35. Lin, S.L.; Nussinov, R.; *J. Mol. Graphics.* **1996**, 14, 78–90.

36. Tsai, C. -J; Lin, S. -L.; Wolfson, H.; Nussinov, R.; *J. Mol. Biol.*, **1996**, 260, 604–620.

37. Tsai, C. -J.; Ph.D. dissertation, Univ. of Pittsburgh, 1992.

38. Bogan, A. A.; Thorn, K. S.; *J. Mol. Biol.* **1998**, 280, 1–9.

39. Bernstein, F.C.; Koetzle, T.F.; Williams, G.J.B.; Meyer, E.F.; Brice, M.D.; Rodgers, J.R.; Kennard, O.; Shimanouchi, T.; Tasumi, M.; *J. Mol. Biol.* **1977**, 112, 535–542.

40. Rick, S. W.; Topol, I. A.; Erickson, J. W.; Burt, S. K.; *Prot Sci* **1998**, 7, 1750–1756.

41. Conte, L., L.; Chothia, C.; Janin, J.; *J. Mol. Biol.* **1999**, 285, 2177–2198.

42. Russell, R. B.; Sasieni, P. D.; Sternberg, M. J. E.; *J. Mol. Biol.* **1998**, 282, 903–918.

43. Stites, W. E.; *Chem Rev* **1997**, 97, 1233–1250.

44. Getzoff, E. D.; Geysen, H. M.; Rodda, S. J.; Alexander, H.; Tainer, J. A.; Lerner, R. A.; *Science* **1987**, 235, 1191–1196.

45. Dougan, D. A.; Malby, R. L.; Gruen, L. C.; Kortt, A. A.; Hudson, P. J.; *Protein Eng.* **1998**, 11, 65–74.

46. Craig, L.; Sanschagrin, P. C.; Rozek, A.; Lackie, S.; Kuhn, L. A.; Scott, J. K.; *J. Mol. Biol.* **1998**, 281, 183–201.

47. Preissner, R.; Goede, A.; Frommel, C.; *J. Mol. Biol.* **1998**, 280, 535–550.

48. Gilson, M. K.; Given, J. A.; Bush, B. L.; McCammon, J. A.; *Biophys J.* **1997**, 72, 1047–1069.

49. Lee, C. Y.; Yang, P. K.; Tzou, W. S.; Hwang, M. J.; *Protein Eng.* **1998**, 11, 429–437.

50. Botti, S. A.; Felder, C. E.; Sussman, J. L.; Silman, I.; *Protein Eng.* **1998**, 11, 415–420.

51. Clackson, T.; Wells, J. A.; *Science* **1995**, 267, 383–386.

52. McCoy, A. J.; Epa, V. C.; Colman, P. M.; *J. Mol. Biol.* **1997**, 268, 570–584.

Figure Legends

Figure 1. A flow chart of the Geometric Hashing, computer vision based algorithm.

Figure 2. A flow chart of the basic steps of the surface comparisons.