# Segmenting Motifs in Protein-Protein Interface Surfaces ⋆

Jeff M. Phillips[1], Johannes Rudolph[2], and Pankaj K. Agarwal[1]

[1] Department of Computer Science, Duke University
[2] Department of Biochemistry, Duke University

**Abstract.** Protein-protein interactions form the basis for many intercellular events. In this paper we develop a tool for understanding the structure of these interactions. Specifically, we define a method for identifying a set of structural motifs on protein-protein interface surfaces. These motifs are secondary structures, akin to $\alpha$-helices and $\beta$-sheets in protein structure; they describe how multiple residues form knob-into-hole features across the interface. These motifs are generated entirely from geometric properties and are easily annotated with additional biological data. We point to the use of these motifs in analyzing hotspot residues.

## 1 Introduction

Interactions between proteins govern many intercellular events, yet are poorly understood. These interactions lie at the heart of cell division and cell growth, which in turn dictate the pattern of health versus disease. A better understanding of protein-protein interactions will enhance our understanding of biological processes and how we can manipulate them for the benefit of human health.

*Protein-protein interfaces.* The protein-protein interface defines the essential region of a protein-protein interaction. Most attempts [10, 12, 20] at defining this interface include all atoms from one protein within some distance cutoff ($4-5$Å) from atoms of the other protein. This approach does not provide independent structural information and makes it difficult to identify features or subregions of the interface. To follow standard notation, we refer to the interface between two protein chains (say A and D) in a protein complex (say 1brs) as 1brsAD.

A more recent approach to defining the protein-protein interface [3] constructs a surface equidistant to both chains of a complex using the Voronoi diagram. This otherwise infinite structure is bounded using topological techniques. Each atom that contributes to the interface is associated with at least one polygon on the interface surface, and each polygon is associated with two

⋆ Research is supported by NSF grant CCR-00-86013 and NIH GM061822. The first author is also supported by a NSFGRF and a JB Duke Fellowship; and the last author also by NSF under grants EIA-98-70724, EIA-01-31905, and CCR-02-04118, and by a grant from the U.S.–Israel Binational Science Foundation.

atoms, one from each side. The freely available MAPS: Protein Docking Interfaces software [2] has been developed to visualize protein interfaces and their relevant properties. The MAPS software displays the interface surface embedded between the protein structures as well as a simplified flattened view. The flattened interface surface can be colored to show physico-chemical properties such as residue type, atom type, distance to closest atom, or electrostatics, as associated with each polygon. A merged view allows both sides of the interface to be viewed simultaneously.

*Features of surfaces.* Proteins have evolved extensive shape complementarity to their interacting proteins. Because protein surfaces are not flat, this leads to knob-into-hole structures at the protein-protein interface, as noted by Connolly [6]. These knobs consist of a set of atoms, often from a single residue, that protrude from one protein into the other protein of the complex. This interaction may be difficult to capture by examination of the structure of the protein complex. However, these knob-into-hole features can be readily identified visually in nearly all protein-protein complexes as bumps on the interface surface. Yet, their biological significance is not well understood, in part because there exists no standard definition or automated method of indentification. Our goal is to automatically segment these regions on the interface surface in a consistent way. We call these knob-in-hole features motifs.

Feature extraction is a common problem in computer vision [13, 5, 19], where features points are used to provide a correspondence between two surfaces. However, these techniques only identify specific points as opposed to interesting regions. For closed curves and surfaces, such as protein surfaces, interesting regions corresponding to knobs and holes can be defined using a variety techniques like elevation [1, 18]. However, these approaches do not generalize to surfaces with boundary, such as the interface surface, and thus the bumps at the interaction site corresponding knobs and holes could not be identified. On a surface with boundary, the problem of defining regions of significance is quite challenging. There is no clear measure of depth, the overall geometry can obscure local features of interest, and local features can be excessively fragmented. As such, it is hard to quantify bumps or pockets, much less segment them.

*Our contribution.* We adapt standard notions of discrete curvature on a polygonal surface to be less local. Using this globally-aware definition we grow regions with large curvature to segment regions of interest on the interface surface. We then integrate these motifs with the MAPS software to incorporate structural information into this convenient visualization tool. Finally we demonstrate some interesting properties of these motifs.

## 2 Discrete Curvature and Watershed Procedures on Polygonal Surfaces

The protein-protein interface surface as defined in [3] is the piecewise-linear surface, that is everywhere equidistance from the closest atom in each of the two proteins. We represent this structure as a set of triangles that are glued together with edges and vertices denoted $\Sigma = \{T, E, V\}$, respectively. All interface surfaces we deal with separate two proteins[1], and thus $\Sigma$ is an orientable piecewise-linear 2-manifold with boundary in $\mathbb{R}^3$. We can thus arbitrarily choose one of these proteins and let the normal direction of every triangle $t \in T$ point towards this protein.

Vertices, edges, and triangles are *simplices.* A vertex is a 0-simplex, an edge is a 1-simplex, and a triangle is a 2-simplex. The *star* of a simplex $\sigma$, denoted $\mathrm{St}(\sigma)$, is the set of simplices incident to $\sigma$. More generally, if $\Upsilon$ is a set of simplices, then $\mathrm{St}(\Upsilon)$ is the set of simplices incident to any simplex in $\Upsilon$. The *link* of a set of simplices $\Upsilon$, denoted $\mathrm{Lk}(\Upsilon)$, is the boundary of the closure of $\mathrm{St}(\Upsilon)$. All interior vertices have the same number of incident edges and incident triangles while boundary vertices have one fewer incident triangles than they do incident edges. All vertices have at least 3 incident edges. For each triangle $t \in \mathrm{St}(v) \cap T$, let $\alpha_{v,t}$ be the angle between the two edges of $t$ incident to $v$, as shown in Figure 1.

### 2.1 Discrete Curvature

To formalize the notion of how much the interface surface is locally bending, we appeal to the idea of curvature. Any point $p$ on a smooth surface can be assigned a value of Guassian curvature and mean curvature [7]. We extend these notions to vertices on a polygonal surface. The standard [15] definition for *discrete Gaussian curvature* of a vertex $v$ is

$$K(v) = 2\pi - \sum_{t \in \mathrm{St}(v) \cap T} \alpha_{v,t}.$$

Intuitively, $K(v)$ describes the angle defect, or how far the angles of the surface surrounding $v$ are from those of a flat surface.

There is no standard definition for the discrete mean curvature of a vertex; it is only defined on edges. We present a new definition for discrete mean curvature, $H$, on vertices. For a triangle $t \in T$, let $\mathbf{n}_t$ denote the unit vector in the direction normal to triangle $t$. We define the normal direction of a vertex $v$ to be the weighted mean of the normal directions of all of its incident triangles, as follows

$$\mathbf{n}_v = \left( \sum_{t \in \mathrm{St}(v) \cap T} \alpha_{v,t} \mathbf{n}_t \right) \Bigg/ \left( \sum_{t \in \mathrm{St}(v) \cap T} \alpha_{v,t} \right). \tag{1}$$

For a triangle $t$ and a vertex $v$ incident to $t$, we define the face vector $\mathbf{f}_{v,t}$ to be the unit vector from $v$ to the mid point of the edge opposite $v$ on $t$. Now we

[1] An interface surface can be defined to separate more than two proteins in a larger complex. In this case the surface is not an orientable 2-manifold because an edge can have three adjacent triangles. However, we can always divide this larger surface into orientable 2-manifold components.

define *discrete mean curvature* of $v$ as

$$H(v) = \left( \sum_{t \in \mathrm{St}(v) \cap T} \alpha_{v,t} (\mathbf{n}_v \cdot \mathbf{f}_{v,t}) \right) \Bigg/ \left( \sum_{t \in \mathrm{St}(v) \cap T} \alpha_{v,t} \right) , \qquad (2)$$

where $\cdot$ indicates a dot product (see Figure 1). Intuitively, $H(v)$ denotes the average deviation from $\mathbf{n}_v$ of the normals of all triangles incident to $v$, weighted by their incident angles. For a vertex $v$, if $K(v)$ is positive, then the sign of $H(v)$ determines if the surface surrounding $v$ is curving towards the normal direction, or away from the normal direction.

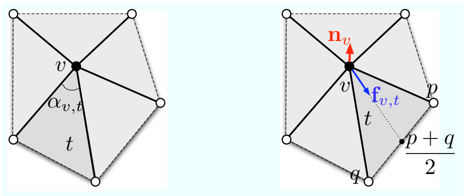

**Fig. 1.** Star of vertex $v$ to illustrate the calculation of $K$ (left) and $H$ (right).

If a polygonal surface is a piecewise-linear approximation of a smooth surface, then the area of triangles, instead of angles can be used as weights in (1) and (2). Since, an interface surface does not approximate a smooth surface, we use the angles of the incident triangles as weights. As the surface is subdivided, the curvature of the surface does not change, and neither do the incident angles.

### 2.2 Watershed Procedures

Let $h : \Sigma \rightarrow \mathbb{R}$ be a function defined on a surface $\Sigma$. If $\Sigma$ is $\mathbb{R}^2$, we can imagine $h$ representing the height of a terrain. Suppose we start pouring water on the terrain and monitor the structure of the flooded regions. The connectivity of the flooded regions changes at the critical points of the terrain: minima, maxima, and saddle points. Lakes are created at minima, islands disappear at maxima, and lakes merge or islands form at saddle points. At saddle points, instead of always merging lakes or creating islands, a dam can be built to define a boundary between two regions. This approach, known as the watershed algorithm [17], can be extended to height functions on 2-manifolds and the dam locations can be used to segment $\Sigma$.

However, there is no clear notion of a height function on an arbitrary 2-manifold. Using $K$ or $H$ as the height does not segment $\Sigma$ well because curvature is a local property. Hence, we modify the above watershed algorithm in two ways. First we run the watershed algorithm from two sides of $\Sigma$ simultaneously using $K$, but use the sign of $H$ to determine on which of the two sides a flooded component lies. We create dams when two components from opposite sides meet, and we merge components from the same side. We face another technical problem if we use $K$ as the height function. Consider the motif (in purple) on the interface surface 1atnAD shown in Figure 2. The value of $K(v_1)$ is large, but $K(v_2)$ is small. A watershed algorithm directly using the function $K$ would not identify the motif shown; rather it would just segment the tip of the motif. By altering the value of $K(v)$ in a careful manner, the algorithm lets components grow in large steps. How we redefine the values of $K(v)$ will be explained in more detail in Section 3.1.

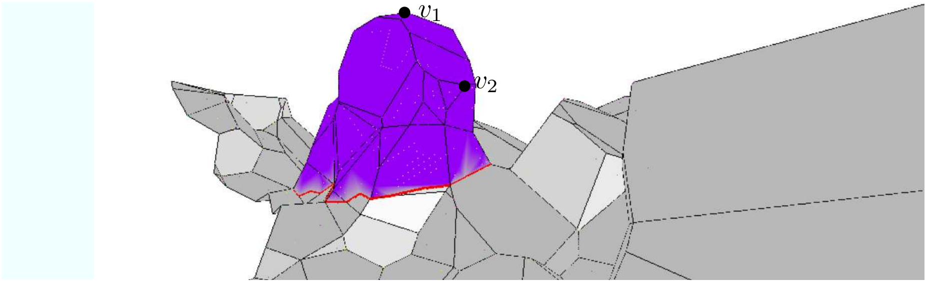

**Fig. 2.** One motif on interface surface 1atnAD. $K(v_1)$ is large, but $K(v_2)$ is not.

## 3 Algorithm for Finding Motifs

Each motif we construct is a component of the subset of the interface surface visited by the modified watershed algorithm. A motif is represented by the set of vertices in its interior. We let $M$ denote both a motif and its set of vertices. A sign $\varsigma(M) \in \{-, +\}$ is assigned to each motif $M$ when it is created. The *rim* of $M$, denoted by $\mathrm{rim}(M)$, is the set of vertices in $\mathrm{Lk}(M)$. If there are two motifs $M$ and $M'$, with $\varsigma(M) \neq \varsigma(M')$, and there is a vertex $r \in \mathrm{rim}(M) \cap \mathrm{rim}(M')$, then $r$ is called a *dam* vertex. A dam vertex never becomes an interior vertex of a motif.

Algorithm 3.1 outlines our algorithm. We set a threshold $\tau$. At each step the algorithm chooses a vertex $v$ with $K(v) \geq \tau$. It makes $v$ a dam vertex, adds $v$ to a motif, creates a new motif with $v$ as its interior vertex, or merges two motifs that share $v$ on their rims. It then recomputes the value of $K(\cdot)$ on the "affected" rim vertices. The worst-case running time of the algorithm is $O(|V|^2)$. However the algorithm took 1 to 75 seconds on the 143 interface surfaces we tested; these surfaces consisted of between 943 and 9096 simplices.

Figure 3 displays from two angles the interface 1brsAD. The motifs are colored in two shades, representing either a knob of chain A protruding into a hole in chain D or vice-versa. We only show motifs larger than $20\text{Å}^2$ to avoid clutter. There are three large motifs (in front on the left view and on the left in the right

**Algorithm 3.1** Find-Motifs($\Sigma, \tau$)

```
$U$: set of interior vertices of $\Sigma$.
while ($\max_{v \in U} K(v) \geq \tau$) do
    Let $v = \arg\max_{v \in U} K(v)$; $U := U \setminus \{v\}$.
    if ($v \in \mathrm{rim}(M) \cap \mathrm{rim}(M') \wedge \varsigma(M) \neq \varsigma(M')$) then
        Mark $v$ as a dam vertex.
    else
        if ($v$ is not a rim vertex) then
            Create a new motif $M = \{v\}$; $\varsigma(M) = \mathrm{sign}(H(v))$.
        else
            Merge all motifs whose rim contains $v$ into a single motif $M$.
            Add $v$ to $M$.
        Compute $\mathrm{rim}(M)$.
        for all $r \in \mathrm{rim}(M)$ do
            Recompute $K(r)$ as described in Section 3.1.
```

view) that form significant knob-into-hole structures. The four smaller motifs mirror this knob-into-hole pattern, albeit less dramatically.

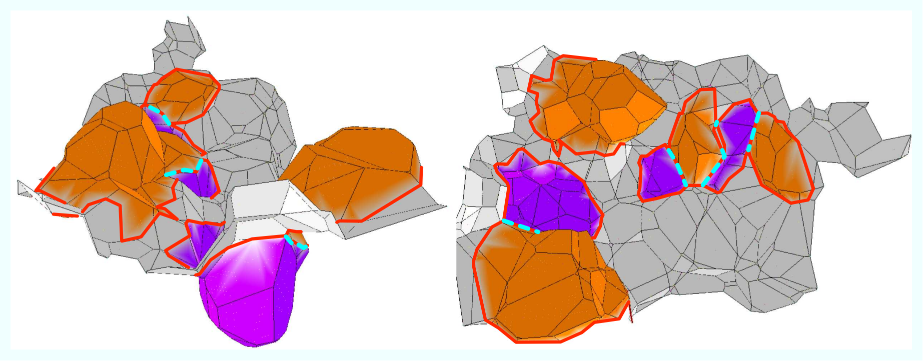

**Fig. 3.** Interface surface and motifs for 1brsAD from two views. Darkly shaded motifs and lightly shaded motifs have opposite signs of $H$, meaning they are protruding in different directions. The rims are the edges in bold, except between the dam vertices where they have light dashes.

### 3.1 Computing the Curvature on the Rim

To compute the curvature for vertices on the rim we implicitly remove the vertices of the motif and treat this part of the surface as if Saran wrap was placed over the removed motif. The idea is that, if an entire motif is chopped off then the surface around it should appear flat; however, if part of the motif is chopped off then the vertices on the rim are now at the tip of a motif and thus should still have large curvature. Actually reconstructing this Saran wrap surface is difficult, and in general can not be done without adding new vertices. However, we can conservatively approximate the star of any vertex in the rim without constructing the full Saran wrap surface. For a vertex $r \in \mathrm{rim}(M)$ we call $\mathrm{St}(r) \setminus \mathrm{St}(M)$—the part of the star of $r$ outside of $M$—the *upper star* of $r$ and we denote it $\mathrm{St}_+(r)$. Similarly, the *lower star* of $r$ is denoted $\mathrm{St}_-(r) = \mathrm{St}(r) \cap \mathrm{St}(M)$. We compute the upper normal of $r$,

$$\mathbf{n}_r^+ = \left( \sum_{t \in \mathrm{St}_+(r) \cap T} \alpha_{r,t} \mathbf{n}_t \right) \Bigg/ \left( \sum_{t \in \mathrm{St}_+(r) \cap T} \alpha_{r,t} \right) .$$

If $\varsigma(M)$ is negative (the surface is curving away from $\mathbf{n}_r^+$) we define $h_r$ as the highest vertex in $\mathrm{rim}(M) \setminus \{r\}$, in the direction of $\mathbf{n}_r^+$. If $\varsigma(M)$ is positive (the surface is curving towards $\mathbf{n}_r^+$), we define $h_r$ as the lowest vertex in $\mathrm{rim}(M) \setminus \{r\}$. We then construct a new star of $r$ by using $\mathrm{St}_+(r)$ and replacing $\mathrm{St}_-(r)$ with the two faces and the edge connecting $h_r$ to the part of $\mathrm{St}_+(r)$ that is incident to $M$, as in Figure 4. The choice of $h_r$ from $\mathrm{rim}(M) \setminus \{r\}$ causes the new surface around $r$ to be as flat as possible. Using the reconstructed star of $r$, $K(r)$ can be computed as before.

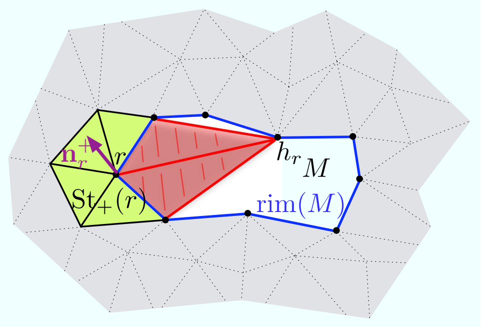

**Fig. 4.** Recreation of $\mathrm{St}(r)$ on $\mathrm{rim}(M)$. $\mathrm{St}_+(r)$ is colored lighter. The Saran wrap surface is colored darker.

### 3.2 Choosing the Threshold $\tau$

Algorithm 3.1 can be run incrementally by gradually lowering $\tau$. At certain values of $\tau$ a new vertex $v$ is handled in Step 3. Let $\tau_v$ denote the threshold $\tau$ at which $v$ first becomes part of a motif $M$. Often the value $K(r)$ for a vertex $r \in \mathrm{rim}(M)$ increases when reevaluated on the rim and sometimes becomes greater than $\tau_v$. In this case $\tau_r = \tau_v$ and $r$ is immediately handled before $\tau$ is lowered. This process often continues until a large motif has been created. As a result, motifs usually grow in large spurts at particular thresholds, as seen in Figure 5. Eventually, as $\tau$ becomes quite small, $U$ becomes empty. This means flat parts of the surface have become part of motifs, often concatenating several large motifs. Choosing an appropriate value of $\tau$ in Algorithm 3.1 will halt the algorithm before this later stage happens.

Although any threshold between 0.3 and 0.4 radians seems to work reasonably well for most surfaces, we can do better by choosing $\tau$ for each surface individually. For instance, surfaces bent around a protein often require larger thresholds, while relatively flat surfaces often require smaller ones. Instead of trying to quantify this bend, we find the threshold that optimizes a score function over the motifs.

As a vertex $v$ is added to a motif $M$ it is given a weight

$$w(v) = \mathrm{Area}(\mathrm{St}_{+}(v)) \cdot \tau_v$$

which is equal to $\tau_v$ times the area that $M$ is incremented. The value $\tau_v$ is never increasing during the algorithm, so vertices added earlier are given more weight.

Motifs are also given a penalty for merging. The penalty $p(M)$ is initially set to 0. When $M$ merges with another smaller motif, $M'$, $p(M)$ is incremented by $\mathrm{Area}(M')$. When a smaller motif is merged with a larger motif, it disappears so its score is essentially set to 0. Each motif $M$ is given a score

$$s_\tau(M) = \sum_{v \in M} w(v) - p(M)$$

for its state at a threshold $\tau$. Thus larger motifs are given more weight, especially if they are created at an early stage. However, if they merge, their scores are heavily penalized. An entire surface is given a score

$$S_\tau(\Sigma) = \mathrm{sign}(\xi_\tau(\Sigma))\sqrt{|\xi_\tau(\Sigma)|}\ , \text{ where } \ \xi_\tau(\Sigma) = \sum_{M \in \Sigma} \mathrm{sign}(s_\tau(M)) \cdot s_\tau(M)^2 .$$

As $\tau$ decreases, $S_\tau(\Sigma)$ usually increases to an optimal value and then decreases as the motifs grow too large and merge. The value of $S_\tau(\Sigma)$ as well as the values $s_\tau(M)$ for all motifs in $\Sigma$ are plotted in Figure 5 for the interface surface for 1nmbN{L,H}. The score is calculated by running Algorithm 3.1 for $\tau$ set to .50 through .25 at .01 intervals. The threshold which returns the largest score $S_\tau(\Sigma)$ is used, which in this case is $\tau = 0.34$. The interface surface and motifs for 1nmbN{L,H} are also shown for different values of $\tau$. Only motifs with surface area larger than 20Å$^2$ are displayed to avoid clutter.

## 4 Applications

The reproducible and automated identification of motifs on the interface surface provides a novel tool for biochemists to characterize protein-protein interactions. We present here two preliminary forays into possible applications. Just as the designation of $\alpha$-helices and $\beta$-sheets has helped enhance the understanding of

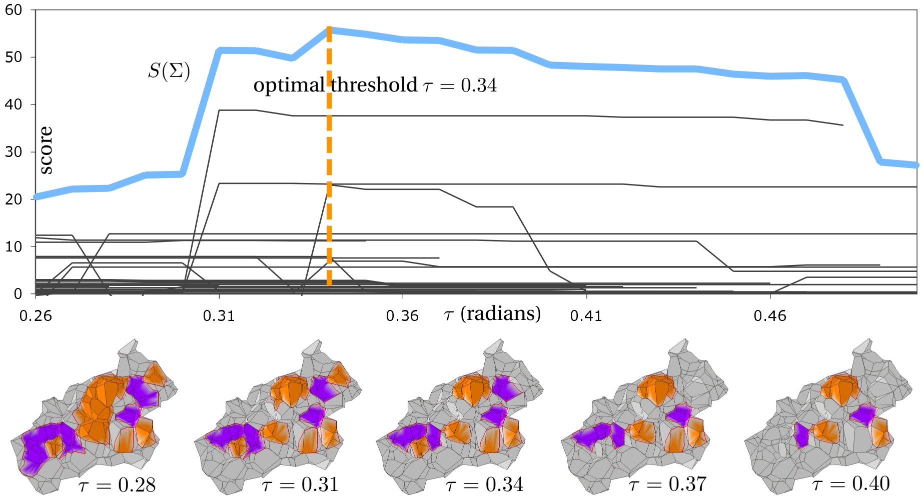

**Fig. 5.** Plot of $S(\Sigma)$ and $s(M)$ for all motifs on interface surface for 1nmbN{L,H}. (top) Interface surface of 1nmbN{L,H} at $\tau = \{0.27, 0.31, 0.34, 0.37, 0.40\}$. (bottom)

the structure of a protein beyond what could be extracted from the atomic coordinates and their ordering on the backbone alone, we envision interface motifs having the same contributions for understanding the structure of the protein-protein interface.

### 4.1 Visualizing the Motifs on the Interface Surface

To readily visualize these structures we have integrated the motifs into the MAPS site [2]. The MAPS software contains a database of over 150 protein-protein interfaces. It displays both a flattened version of the interface and a 3D version embedded within the protein structures, side-by-side. By flattening the interface, the 3D structure is removed so that the physical and chemical properties from one or both sides can be more easily visualized. Mapping the motifs into this flattened view reintroduces structural information and allows for facile comparisons between knobs and holes and physical and chemical properties.

Figure 6 shows a snapshot of the MAPS software displaying the interface surface for 1brsAD. On the left is the standard 3-dimensional view embedded in the structures of the protein. In the middle the flattened interface is colored by its motifs as seen from chain D. The two shades of motifs correspond to either chain A protruding into chain D or vice-versa. The thick black lines show the outline of the surface patches corresponding to different residues. Note how the large lightly shaded motifs are at the junction of many residue patches wherein the hole in chain D, made by the knob in chain A, is composed of multiple residues. On the other hand, the darkly shaded motifs are generally at the center of a single residue patch wherein the knob in chain D consists of only one residue.

This pattern is reversed if the other side of the interface is shown. On the right the flattened interface is colored by its amino acid composition as seen from chain D. A single motif is highlighted in all views. Note the large overlap of the highlighted motif with Asp35, one of the key hotspot residues in chain D of the 1brsAD complex.

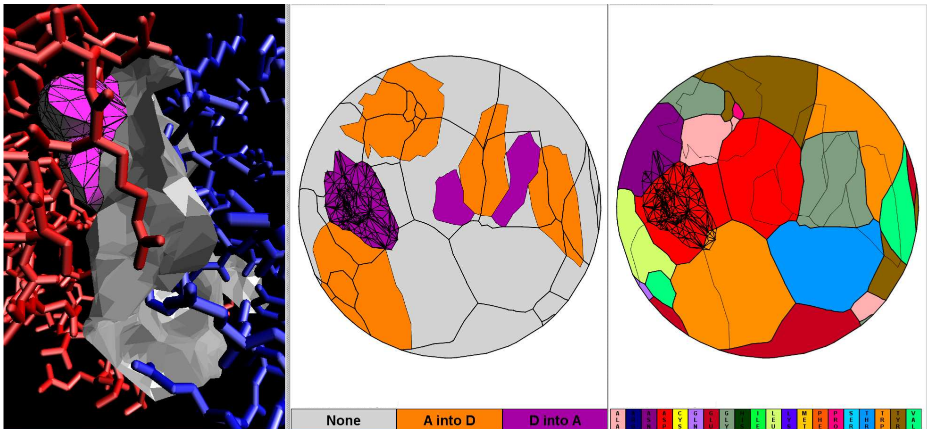

**Fig. 6.** Snapshot of MAPS software: Shows interface surface 1brsAD with protein complex drawn as a stick model (left), motifs on flattened interface (middle), and residue type on flattened interface. Motif corresponding to Asp35 is highlighted in all views.

### 4.2 Hotspot Residues and Motifs

Hotspot residues for a protein complex are those few residues (5-10%) at a protein interface whose mutation leads to a significant reduction in the binding energy ($\Delta\Delta G \geq 2$ kcal/mol) [4]. The identification of hotspots is not necessarily a matter of trivial inspection as essentially any type of residue can be a hotspot. Thus various computational methods have been developed to predict hotspots, relying either on traditional force fields [14, 9] or simpler physical energy functions parameterized on experimental data [8, 11], with reasonable to good results.

Intuitively their exists a potential relationship between motifs and hotspot residues. In particular, one might expect prominent knobs associated with a single residue to be likely hotspot residues. For illustration we consider the complex 1brsAD. For the 14 residues on the interface characterized by mutation [16] we compare each residue to the motif which has the largest area of overlap on the interface surface. That is, for each residue we compute the area of overlap and the percent of its area that is in this overlap region. In Figure 7 we plot these values versus the $\Delta\Delta G$s of the mutated residues. Three residues, including Asp35, are clearly identified as hotspots whereas three hotspot residues with $\Delta\Delta G > 4$

kcal/mol are not identified by this method. Satisfyingly, none of the non-hotspot residues are incorrectly labeled as such.

Future inquires into this relationship will incorporate a larger dataset of hotspot residues and the chemical properties of the potential hotspot residues.

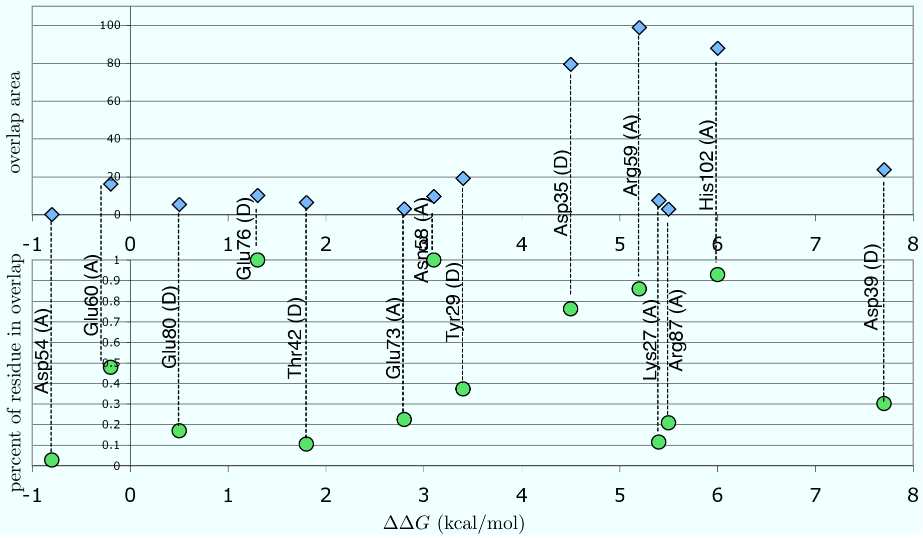

**Fig. 7.** Plots comparing hotspot residues to motifs on the interface surface 1brsAD. Top chart plots the area of the overlap between the hotspot residue and the motif best covering it on the interface surface versus $\Delta\Delta G$ of that residue. Bottom chart plots the percent of the residue area which is in the overlap region versus the $\Delta\Delta G$ of the residue.

## 5 Future Work

In addition to a possible correlation with hotspot residues, we envision that our structural annotation tool could aid other biological applications. For instance, statistical analysis may reveal preferred knob residues whose identification may be useful for prediction of protein docking. Moreover, overlaying motifs from different interfaces may lead to classifications by size or shape that can be useful for describing recurring interfacial motifs. Also, by observing higher order relationships of motif arrangements on the protein interface (tertiary structures), we envision new ways of comparing and classifying protein-protein complexes.

### Acknowledgments

We thank Tammy Bailey and Herbert Edelsbrunner for observations and suggestions and Jeffrey Headd for help integrating the motifs into the MAPS software.

## References

1. P. K. Agarwal, H. Edelsbrunner, J. Harer, and Y. Wang. Extreme elevation on 2-manifold. *Proceedings 20th Annual Symposium on Computational Geometry*, 2004.
2. Y.-E. A. Ban, P. L. Brown, H. Edelsbrunner, J. J. Headd, and J. Rudolph. MAPS: Protein docking interfaces. http://biogeometry.cs.duke.edu/research/docking/index.html, May 2006.
3. Y.-E. A. Ban, H. Edelsbrunner, and J. Rudolph. Interface Surface for Protein-Protein Complexes. *in press, Journal of the Association of Computer Machinery.*
4. A. A. Bogan and K. S. Thorn. Anatomy of hot spots in protein interfaces. *Journal of Molecular Biology*, 280:1–9, 1998.
5. C. S. Chua and R. Jarvis. Point signatures: A new representation for 3d object recognition. *International Journal of Computer Vision*, 25(1):63–85, 1997.
6. M. L. Connolly. Shape complementarity at the hemoglobin $\alpha_1\beta_1$ subunit interface. *Biopolymers*, 25(7):1229–1247, 1986.
7. M. P. do Carmo. *Differential Geometry of Curves and Surfaces.* Prentice-Hall, Upper Saddle River, NJ, 1976.
8. R. Guerois, J. E. Nielsen, and L. Serrano. Predicting changes in the stability of proteins and protein complexes: A study of more than 1000 mutations. *Journal of Molecular Biology*, 320:369–387, 2002.
9. S. Huo, I. Massova, and P. A. Kollman. Computational alanine scanning of the 1:1 human growth hormone-receptor complex. *Journal of Computational Chemistry*, 23:15–27, 2002.
10. S. Jones and J. M. Thornton. Analysis of protein-protein interaction sites using surface patches. *Journal of Molecular Biology*, 272:121–132, 1997.
11. T. Kortemme and D. Baker. A simple physical model for binding energy hot spots in protein-protein complexes. *Proceedings National Academy of Science USA*, 99:14116–14121, 2002.
12. L. Lo Conte, C. Chothia, and J. Janin. The atomic structure of protein-protein recognition sites. *Journal of Molecular Biology*, 285:2177–2198, 1999.
13. D. G. Lowe. Object recognition and local scale-invariant features. *Proceedings Seventh IEEE International Conference on Computer Vision*, 2:1150–1157, 1999.
14. I. Massova and P. A. Kollman. Computational alanine scanning to probe protein-protein interactions: A novel approach to evaluate binding free energies. *Journal of the American Chemical Society*, 121:8133–8143, 1999.
15. M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. *Visualization and Mathematics III*, 2003.
16. G. Schreiber and A. R. Fersht. Energetics of protein-protein interactions: Analysis of the barnase-barstar interface by single mutations and double mutation cycles. *Journal of Molecular Biology*, 248:478–486, 1995.
17. L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, 1991.
18. Y. Wang, P. K. Agarwal, P. Brown, H. Edelsbrunner, and J. Rudolph. Coarse and reliable geometric alignment for protein docking. *10th Annual Pacific Symposium on Biocomputing*, pages 64–75, 2005.
19. Y. Wang, B. S. Peterson, and L. H. Staib. 3d brain surface matching based on geodesics and local geometry. *Computer Vision and Image Understanding*, 89:252–271, 2003.
20. D. Xu, C.-J. Tsai, and R. Nussinov. Hydrogen bonds and salt bridges across protein-protein interfaces. *Protein Engineering*, 10(9):999–1012, 1997.