Extraction of Robust Voids and Pockets in Proteins

Raghavendra Sridharamurthy[1] Harish Doraiswamy[2] Siddharth Patel[3] Raghavan Varadarajan[3]
Vijay Natarajan[1,4]

[1]Department of Computer Science and Automation, Indian Institute of Science, Bangalore
[2]Polytechnic Institute of New York University, USA
[3]Molecular Biophysics Unit, Indian Institute of Science, Bangalore
[4]Supercomputer Education and Research Centre, Indian Institute of Science, Bangalore

IISc-CSA-TR-2013-3
http://www.csa.iisc.ernet.in/TR/2013/3/

Computer Science and Automation
Indian Institute of Science, India

May 2013

This report is an extended version of the EuroVis 2013 short paper titled "Extraction of robust voids and pockets in proteins"

Extraction of Robust Voids and Pockets in Proteins

Raghavendra Sridharamurthy[*] Harish Doraiswamy[†] Siddharth Patel[‡]
Raghavan Varadarajan[§] Vijay Natarajan[¶]

Abstract

Voids and pockets in a protein refer to empty spaces that are enclosed by the protein molecule. Existing methods to compute, measure, and visualize the voids and pockets in a protein molecule are sensitive to inaccuracies in the empirically determined atomic radii. This paper presents a topological framework that enables robust computation and visualization of these structures. Given a fixed set of atoms, voids and pockets are represented as subsets of the weighted Delaunay triangulation of atom centers. A novel notion of (ε, π)-stable voids helps identify voids that are stable even after perturbing the atom radii by a small value. An efficient method is described to compute these stable voids for a given input pair of values (ε, π).

1 Introduction

Protein molecules have a well packed structure, yet they contain cavities. A cavity refers to both voids (without openings) and pockets (with openings). These cavities play a key role in determining the stability and function of proteins.

Several methods have been proposed to locate such cavities in protein molecules. We focus our attention on geometric methods. Edelsbrunner et al. [6, 8] and Liang et al. [16, 17] propose a definition that is based on the theory of alpha shapes and discrete flows in Delaunay triangulations. Kim et al. [11, 13] propose a definition of cavities based on an alternate representation of a set of atoms called beta shapes that faithfully captures proximity. Tools based on the above approach are available and widely used [2, 14, 12]. Till and Ullmann [24] employ a graph theoretic algorithm to identify cavities and compute their volume. Parulek et al. [21] use graph based methods on the implicit representation of molecular surfaces to identify pockets and potential binding sites. Varadarajan et al. [1] employ a Monte Carlo procedure to position water molecules together with a Voronoi region-based method to locate empty space. They discuss the importance of accurate identification of cavities for the study of protein structure and stability. Novel Voronoi diagram-based techniques for the extraction and visualization of cavities have also been developed from the viewpoint of studying and interactively exploring access paths to active sites [23, 22, 19, 18]. Krone et al. [15] present a visualization tool for interactive exploration of protein cavities in dynamic data.

1.1 Motivation and Problem Statement

The input used for calculations in previous work come from x-ray crystallography data or other lower resolution data. Previous cavity detection methods are sensitive to inaccuracies that are inherent in the crystallographic measurements. While the measurements may guarantee high resolution, it is important to note that even small inaccuracies may cause a significant difference in the reported number of cavities. The inaccuracies may also arise due to some fundamental limitations such as the notion of radii of atoms, which is determined empirically. For example, as illustrated in Figure 1, presence of such inaccuracies may result in a cavity detection method to report two distinct but large cavities in place of one or report very small volume cavities. Figure 2 illustrates the problem as it occurs in a

[*] Department of Computer Science and Automation, Indian Institute of Science, Bangalore. raghavendra@csa.iisc.ernet.in

[†] Polytechnic Institute of New York University, USA.

[‡] Molecular Biophysics Unit, Indian Institute of Science, Bangalore.

[§] Molecular Biophysics Unit, Indian Institute of Science, Bangalore.

[¶] Author for TR correspondence. Department of Computer Science and Automation, Supercomputer Education and Research Centre, Indian Institute of Science, Bangalore. vijayn@csa.iisc.ernet.in

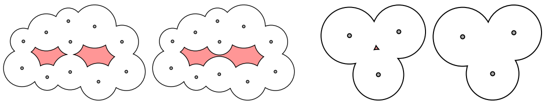


Figure 1: **Left:** Two cavities that are apparently very near to each other may be a single cavity. **Right:** A very small cavity may be reported whereas no such cavity may exist.

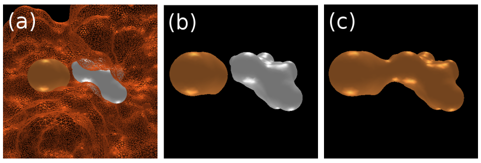


Figure 2: **(a)** Two voids that appear very near to each other in a lyzozyme protein (PDB ID: 2001). The solid surface represents the voids while the wireframe represents the molecule. **(b)** Zoomed-in view of the voids without the molecular skin surface. **(c)** The two voids may be a single void.

lyzosyme protein.

We aim to develop an interactive method to compute robust cavities in proteins. We achieve this by enabling the user to reduce, if not completely eliminate, the inaccuracies mentioned earlier. We define what robustness would mean in this context, why such a notion is important and also demonstrate the robustness of the method. As an auxiliary task, we visualize the protein molecules and their cavities in an interactive manner and compute their properties.

From the biologist's point of view, obtaining a stable protein is the starting point of many applications, from in-vitro studies of binding and interactions, to using the protein as an antigen or vaccine. Whereas surface pockets often form part of the active site of enzymes or interacting sites for other proteins, internal voids are often relevant structurally as features that affect the overall stability of the protein. It is established that filling up internal voids improves the packing of the protein thus increasing stability. In this respect, detecting and visualizing structurally robust cavities inside the protein informs the biologist on which mutations to perform to improve internal packing and get a stable protein.

Our software while detecting robust cavities and calculating volumes, also provides an interactive framework that the biologist can use, along with their own knowledge and discretion, to decide which cavities affect their proteins and which mutations would be required for their specific application.

1.2 Results

The main results described in this paper are in the context of a novel definition and method for computing robust and stable voids in proteins. We employ a simple and succinct structure called the alpha complex to represent protein molecules. The alpha complex is a simplicial complex that can be stored as a filtration, a series of simplicial complexes K^i with $K^i \subset K^{i+1}$. With the aim of computing a set of voids that are stable with respect to small perturbations in the atom radii, we develop a method that modifies the radii of a select set of atoms symbolically by systematically processing and modifying the filtration. We show that this modification results in controlled changes in the number and properties of voids and does not violate key properties of the filtration. The method also supports the elimination of very small or insignificant voids as measured by the notion of topological persistence [9]. We also develop software to visualize the stable cavities together with the molecule, and to calculate cavity volume and surface areas.

2 Background

In this section, we briefly introduce the topological background required to define and represent the structure of biomolecules [20, 5, 4].

Simplicial Complex. A *k-simplex* σ is the convex hull of $k+1$ affinely independent points. A vertex, edge, triangle, and tetrahedron are k-simplices of dimension $0-3$. A simplex τ is a *face* of σ, $\tau \leq \sigma$, if it is the convex hull of a non-empty subset of the $k+1$ points. A *simplicial complex* K is used to represent a topological space and is a finite collection of simplices such that (a) $\sigma \in K$ and $\tau \leq \sigma$ implies $\tau \in K$, and (b) $\sigma_1, \sigma_2 \in K$ implies $\sigma_1 \bigcap \sigma_2$ is either empty or a face of both σ_1 and σ_2. A *subcomplex* of K is a simplicial complex $L \subseteq K$.

Voronoi diagram and Delaunay triangulation. Let $S \subseteq \mathbb{R}^d$ be a finite set of points. The *Voronoi cell* V_p, of a point $p \in S$, is the set of points in $\mathbb{R}^d$ whose Euclidean distance to p is smaller than or equal to any other point in S. The collection of Voronoi cells of all points in S partitions $\mathbb{R}^d$, and is called the *Voronoi diagram* (Figure 3(a)). The *Delaunay triangulation* D

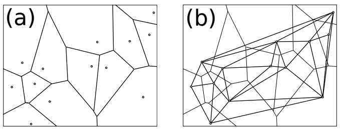


Figure 3: **(a)** Voronoi diagram of a point set in $\mathbb{R}^2$. **(b)** The Delaunay complex is the dual of the Voronoi diagram.

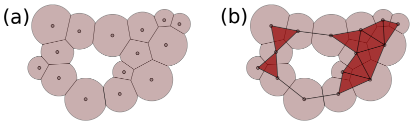


Figure 4: **(a)** Intersection of the weighted Voronoi diagram and the union of balls. **(b)** The dual complex is the dual of this partition of the union of balls that captures the incidence relationship.

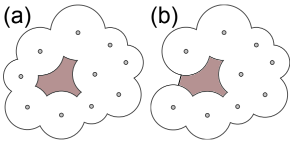


Figure 5: **(a)** Void and **(b)** Pocket in a collection of 2D balls.

of S is the dual of the Voronoi diagram and partitions the convex hull of S, see Figure 3(b). The above definitions can be extended to a set of balls or weighted points by choosing an appropriate measure of distance between a weighted point p and a point in $\mathbb{R}^d$. The *power distance* between p and a point $x \in \mathbb{R}^d$ is equal to $\pi_p(x) = \|x - p\|^2 - w_p$, where w_p is the weight of p.

Alpha Complex. Molecules are often represented using a space-filling model such as a union of balls. The weighted Voronoi diagram may be extended to represent the contribution from each atom to the union of balls. Consider an atom p. Define B_p as an open ball having the radius of the atom p. Let V_p be the weighted Voronoi cell corresponding to p, where the weight is equal to the square of the atom radius. The contribution from each atom p is equal to $B_p \cap V_p$, the intersection between the ball corresponding to the atom and the weighted Voronoi cell of p. The corresponding dual structure is a subcomplex of the weighted Delaunay triangulation and called the *dual complex*, see Figure 4.

Edelsbrunner et al. [7, 10, 3] consider a growth model where the ball radii grow, and track the changes in the dual complex. The growth parameter, α, corresponds to a radius $\sqrt{r_p^2 + \alpha^2}$ for a ball centered at p with radius r_p. The weight of the point $w(p)$ increases to $w(p)+\alpha^2$. Note that $\alpha = 0$ corresponds to no growth. The dual complex corresponding to a set of balls after they are grown by α is called the *alpha complex*.

Given a simplicial complex K, a finite sequence, $\emptyset = K^0, K^1, \ldots, K^m = K$, of subcomplexes of K is a *filtration* if $K^0 \subset K^1 \subset \cdots \subset K^m$. The *rank* of a subcomplex refers to its position in the filtration. The set of alpha complexes obtained by varying α from $-\infty$ to ∞ is a filtration of the Delaunay triangulation. In particular, we consider the filtration that is generated by inserting the simplices one at a time and ties broken based on the dimension of the simplex.

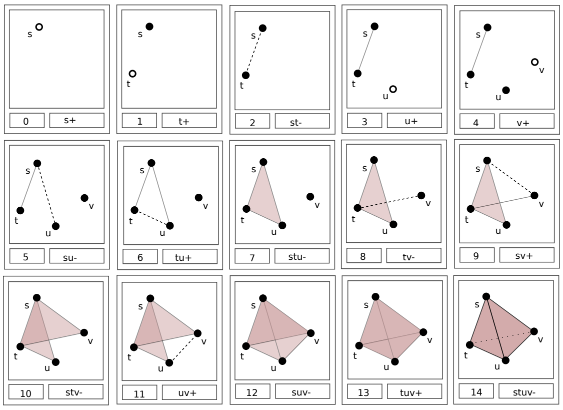


Figure 6: A filtration generated by inserting simplices in a particular order. The arrival time and the behavior of the simplices are also shown in the boxes.

Voids and Pockets. Let the alpha complex K represent a molecule at a given value α and D be the Delaunay complex of the weighted point set. A *cavity* is a maximally connected component of the complement $D - K$. *Voids* and *pockets* are cavities that are, respectively, bounded and not bounded by the union of balls. Figure 5 illustrates a void and a pocket in 2D.

Topological persistence. The alpha complex helps represent and track voids via the growth process. A void is represented by a 2-*cycle*, which is a collection of 2-simplices whose boundary is empty. A 2-cycle is created when the last triangle in the collection is inserted and it is destroyed by the insertion of a tetrahedron. *Topological persistence* of a k-cycle measures its lifetime ($k \geq 0$) in a filtration [9]. It is equal to the difference between the ranks of alpha complexes in the filtration where the cycle is created and destroyed. If a 2-cycle doesn't have a destroyer then its persistence is equal to ∞. Given a filtration, the persistence of cycles can be computed efficiently [9]. The insertion of every simplex either creates a cycle or destroys a lower dimensional cycle. The persistence value associated with a simplex is equal to the topological persistence of the corresponding cycle.

Figure 6 illustrates creation and destruction of cycles in a filtration of a small simplicial complex. The number in each box denotes the timestamp. The simplex that is added is denoted in the adjacent box along with a positive or a negative sign which respectively signifies creation or deletion of a cycle.

3 Robust Voids and their Computation

We introduce a notion of robust voids based on two parameters, one local and another global. The local parameter is referred to as stability and the global parameter is specified by topological persistence. In order to simplify the description, we assume that the voids are computed for the α-complex corresponding to $\alpha = 0$. However, the definitions, methods, and subsequent analysis are valid for all values of α.

3.1 ε-stable and π-persistent voids

Consider the interval $[-\varepsilon, \varepsilon]$ of α values, where $\varepsilon \geq 0$. A void is called an ε-*stable* void if it remains a single connected void within all α-complexes for α values in the range $[-\varepsilon, \varepsilon]$. In other words, using the lifetime terminology, the void is born, possibly split into multiple components, and destroyed at α-values that lie strictly outside of this interval. A void is π-*persistent* if its topological persistence is greater than π *i.e.*, the void size measured in terms of its lifetime is greater than π. Combining the two notions of robustness, we call a void to be (ε, π)-*stable* if it is both ε-stable and π-persistent.

The above definitions help measure the stability of the voids when the radii are perturbed by a small value. The local parameter considers perturbation within a small interval centered at the α-value of interest whereas the global parameter measures the size of the void in terms of its lifetime in the filtration. Voids of interest may often not be stable with respect to both notions. For example, a large sized void (π-persistent for some large π) may be born within the interval $[-\varepsilon, \varepsilon]$. However, note that a small perturbation in the radii of atoms that line the surface of the void could result in an earlier birth time, hence making the void to be ε-stable. We aim to extract all voids that are either stable as is or can be made stable via a small perturbation.

3.2 Computing (ε, π)-voids

The location of the atoms that constitute a protein molecule together with their van der Waals radii is obtained from the protein data bank in pdb format. Given ε and π, we compute the set of (ε, π)-stable voids as follows.

1. Compute the weighted Delaunay triangulation of the input [14]. The atom centers form the set of points that are weighted using their van der Waals radii.
2. Build the alpha shape spectrum [3], which is a filtration of the weighted Delaunay triangulation.
3. **Modify the filtration based on the value of ε.**
4. Compute the set of (ε, π)-stable voids by identifying all voids [16] of the modified filtration at $\alpha = 0$, and retaining only those voids that have persistence at least π.

Modifying the filtration. The filtration of the weighted Delaunay triangulation as defined by the α-values provides an explicit representation of the birth/death times of each void and the evolution during its lifetime. We propose to alter the birth/death times of voids by modifying the filtration instead of directly modifying radii of atoms that line the surface of the void. While the latter approach follows directly from the definition, it is cumbersome and computationally inefficient. For example, varying the radii without explicit control may lead to changes in the triangulation and the alpha complex. These changes need to be explicitly tracked, else they may lead to inconsistencies between the alpha complex that represents the molecule and the space-fill model. Resolving such inconsistencies would necessitate the re-computation of all representations. On the other hand, the former approach is simpler and computationally efficient.

Delayed simplex insertion. One or more simplices are inserted to obtain a rank $i+1$ simplicial complex from a rank i simplicial complex in the filtration. Higher ranks correspond to higher values of α. The topology of voids may change when the simplices are inserted. In particular, the insertion of a triangle may either not affect any void, split a void into two, or create a new void. On the other hand, the insertion of a tetrahedron always destroys a void. These topology changes may therefore be avoided by delaying the insertion of the simplices that change the topology of voids. Let K^j be the alpha complex corresponding to $\alpha = \varepsilon$. Consider the set of simplices, Σ, inserted into the filtration for values of α in the range $[-\varepsilon, \varepsilon]$. Let $\Sigma_t \subset \Sigma$ be the set of the triangles that split a void, and $\Sigma_T \subset \Sigma$ be the set of tetrahedra. We delay the insertion of simplices σ_i in Σ_t and Σ_T such that $\sigma_i \notin K^j$ but $\sigma_i \in K^l$, where $K^j \subset K^l \subset D$. Simplicial complexes in the filtration of the weighted Delaunay triangulation and the order of simplices that are inserted to generate the filtration satisfy several containment and incidence properties. These properties should be satisfied for the modified filtration as well. Towards this, we propose a conservative but computationally efficient approach to modify the filtration:

1. Move all tetrahedra in Σ_T to the end of the filtration. All such tetrahedra are present in D but not in any $K^i \subset D$.
2. For each triangle in Σ_t, find its incident tetrahedra τ_1, τ_2.
3. Delay the insertion of the triangle and the two tetrahedra, τ_1 and τ_2, to the end of the filtration.

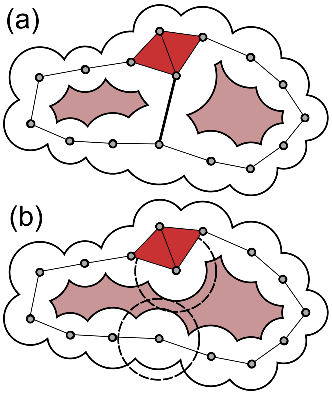


Figure 7: 2D illustration of simplex insertion causing a void to split. **(a)**. Voids occur near to each other and the edge that splits the single void into two. **(b)**. The two voids merge into one if the simplex insertion is delayed.

To illustrate the modification in 2D let us consider a void split into two as shown in Figure 7(a). Assume that the highlighted edge (triangle in 3D) becomes part of the alpha complex in the interval $[-\varepsilon, \varepsilon]$. Further it also satisfies the criterion that it bounds two different tunnels (voids in 3D). So it becomes a candidate for delayed insertion. We move the edge to the end of the filtration, which means that it does not belong to the alpha complex as shown in Figure 7(b). Also the radii of atoms centered at the end points of the edge (triangle in 3D) are changed accordingly in relation to the ε value. Selective modification of the radii of a specific set of atoms is hence achieved in a controlled manner.

Implication of the modified filtration. Consider the delayed insertion of a triangle (edge in 2D) to avoid the split of a void into two as illustrated in Figure 7. The delay corresponds to shrinking the atoms centered at the vertices of the triangle. However, note that the radii are not yet modified. We optionally modify the radii later for further analysis of the void. A triangle that creates a void is left untouched and the corresponding void is also declared to be ε-stable. The triangle insertion may be advanced to ensure that the void is created outside the interval. This corresponds to a small increase in the radii of the atoms centered at the vertices of the triangle. We choose not to explicitly advance the triangle insertion because it does not affect the results for small values of ε. After the filtration is modified as described above, we recompute the voids from the alpha complex.

We compute the persistence of the ε-stable voids obtained and retain only those having persistence greater than π. This pruned set of voids are (ε, π)-stable. Note that the persistence is computed with respect to the original filtration. The above notion of stability can be extended to pockets as well.

3.3 Implementation details

The filtration obtained from the alpha shape spectrum (Step 2) is stored as a list. The index of each simplex in this list represents the rank of that simplex. For each triangle, we additionally store the indices of its coface tetrahedra. For a given ε, we first compute the ranks k_1 and k_2 of the alpha shape at $\alpha = -\varepsilon$ and $\alpha = \varepsilon$, respectively, and the set of voids present in the interval, $\alpha \in [-\varepsilon, \varepsilon]$. If the addition of a triangle σ_k, $k_1 \leq k \leq k_2$, splits a void, σ_k is added to the set Σ_t. The coface tetrahedra τ_1 and τ_2, of σ_k is added to Σ_T. All tetrahedra having their rank in the range $[k_1, k_2]$ are added to Σ_T. The set of triangles and tetrahedra in Σ_t and Σ_T, respectively, are sorted in the increasing order of their ranks. Instead of explicitly moving these simplices to the end of the filtration, we perform an implicit move. These simplicies are marked as invalid in the list containing the original filtration. The new filtration is obtained by traversing the original filtration, ignoring the invalid simplicies, followed by traversing Σ_t and Σ_T. An advantage of using this approach is that, when the value of ε is changed, it is easy to revert to the original filtration and recompute the new filtration.

3.4 Analysis

Let m be the number of simplices in the Delaunay triangulation of the input protein having n atoms, $m = O(n^2)$. Computing the set of voids takes $O(m\alpha(m))$ time using the union-find data structure. Here, α is the inverse Ackermann function. Given ε, Σ_t and Σ_T is computed in $O(m)$ time using a sequential search over the filtration. Identifying the set of tetrahedra incident on triangles in Σ_t, and moving all the simplices to the end of the filtration takes $O(m)$ time. Thus the time required to modify the filtration is $O(m\alpha(m))$.

3.5 Stability of pockets

The notion of stability defined for voids can be extended to pockets as well. A pocket may be destroyed in the filtration by the addition of a triangle. By destroying the pocket, this triangle creates a void, or a void-pocket pair. In our current implementation, the voids that are created by the triangle are given preference over pockets that are destroyed in the $(-\varepsilon, \varepsilon)$ range. We choose to do this because a pocket with a very narrow opening (size less than ε) will not allow any ion to pass into it, and hence functions more as a void. Therefore, the only stable pockets that are identified correspond to those that are destroyed or split into a void-pocket pair outside of the $(-\varepsilon, \varepsilon)$ range. However, in case a pocket is preferred over a stable void, the algorithm can be easily modified to delay the insertion of the triangle that destroys the pocket, along with its coface tetrahedra.

4 Results

Given the input protein, our software *RobustVoids* first computes the alpha complex of the input. The user can visualize the protein either as a skin surface, or as a volume mesh for different α-values. The set of (ε, π)-stable cavities are computed using the values of ε and π specified by the user. The software also reports the cavity volume and surface area.

Visualization of stable cavities. Figure 8(a) shows protein 2CI2, which has 3 voids. Using a value of $\varepsilon = 0.3$ and $\pi = 0.01$ results in two $(0.3, 0.01)$-stable voids, see Figure 8(b). Note that a value of $\varepsilon = 0.3$ is equivalent to an increase / decrease of the radius of an atom by at most 0.33Å, which is within the tolerated 0.5Å used by the biologists. Modifying the filtration and computing the stable voids for this protein takes 0.1 s.

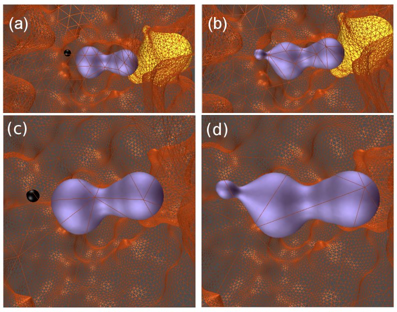
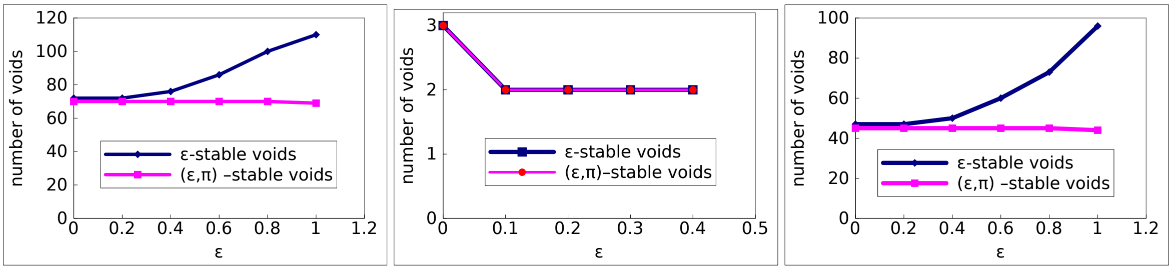


Figure 8: Visualization of voids of the protein 2CI2. The values $\varepsilon = 0.3$ and $\pi = 0.01$ was used to compute the set of stable voids. **(a)** Voids in the volume computed at $\alpha = 0$. Number of voids = 3. **(b)** The set of $(0.3, 0.01)$-stable voids. Number of $(0.3, 0.01)$-stable voids = 2. **(c)** Two nearby voids in the protein rendered in solid and the skin surface of the molecule rendered in wireframe mode. **(d)** The merged stable void shown along with the skin surface of the molecule.

The protein 4HHB has a total of of 72 voids, shown in Figure 9(a). Using a value of $\varepsilon = 1.0$ and further removal of voids having persistence $\pi < 0.01$ results in a total of 70 $(1.0, 0.01)$-stable voids, see Figure 9(b). Figure 9(c) shows two nearby voids in the protein which merge to form a single stable void, see Figure 9(d). Note that a value of $\varepsilon = 1.0$ is equivalent to an increase / decrease of the radius of an atom by at most 0.33Å, which is within the tolerated 0.5Å used by the biologists. Modifying the filtration and computing the stable voids for this protein takes 62 secs. Figure 10 shows the visualization of the (ε, π)-stable voids for the protein 4B87. Modifying the filtration and computing the stable voids for this protein takes 33 secs.

The implementation for now uses a naive approach of explicitly computing the voids/pockets while constructing the set of simplices which needs to be moved and thus the running time is large. This can be reduced by tracking the creators of voids/pockets as described in [9].

Properties of stable cavities. Figures 11 and 12 plots the number and volume of (ε, π)-stable voids for various values of ε. Note that increasing the value of ε implies that voids from a larger α range are considered. This could potentially increase the number of ε-stable voids. However, such voids usually have low persistence and are therefore not (ε, π)-stable. We use

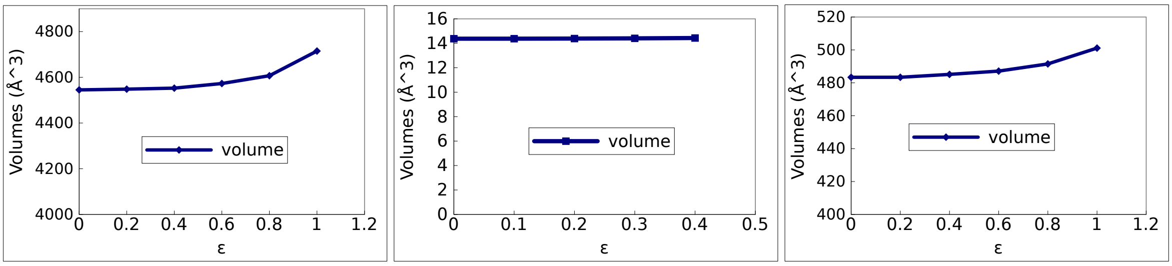


(a) Protein 4HHB

(b) Protein 2CI2

(c) Protein 4B87

Figure 11: Graphs showing the variation of the number of voids with varying ε. Note that there is an increase in ε-stable voids as we consider a larger interval but (ε, π) voids are less than or equal to original number of voids.



(a) Protein 4HHB

(b) Protein 2CI2

(c) Protein 4B87

Figure 12: Graphs showing the variation of the total volume of voids with varying ε.

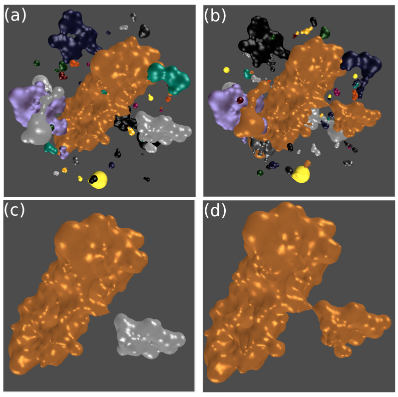


Figure 9: Visualization of voids of the protein 4HHB. The values $\varepsilon = 1.0$ and $\pi = 0.01$ was used to compute the set of stable voids. **(a)** Voids in the protein computed at $\alpha = 0$. Number of voids = 72. **(b)** The set of $(1, 0.01)$-stable voids. Number of $(1.0, 0.01)$-stable voids = 70. **(c)** Two nearby voids in the protein. **(d)** These two voids merge together resulting in a single stable void.

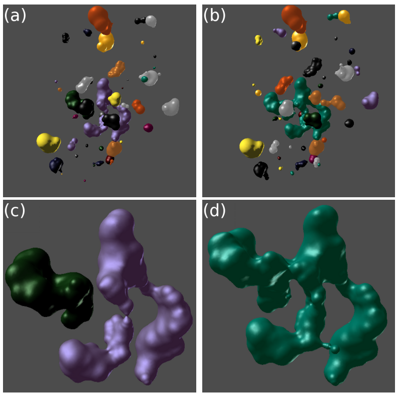


Figure 10: Visualization of voids of the protein 4B87. The values $\varepsilon = 1.0$ and $\pi = 0.01$ was used to compute the set of stable voids. **(a)** Voids in the protein computed at $\alpha = 0$. Number of voids = 47. **(b)** The set of $(1.0, 0.01)$-stable voids. Number of $(1.0, 0.01)$-stable voids = 44. **(c)** Two of the nearby voids in the protein. **(d)** These voids merge together resulting in a single stable void.

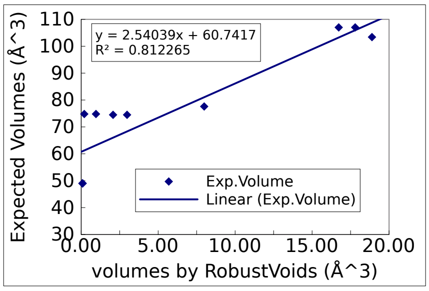


Figure 14: Plot of the computed and actual volumes of the artificial voids that were generated using mutant models.

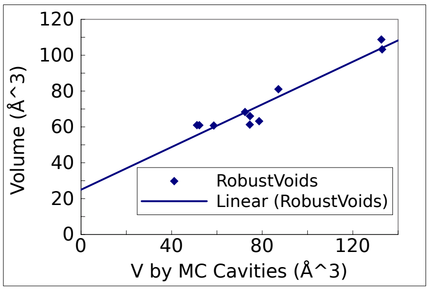


Figure 15: Comparison of normalised volumes computed using RobustVoids and MC Cavity.

a constant value of $\pi = 0.01$ in all experiments. The total volume of all stable voids increases marginally ($< 1\%$) with increasing ε. The merging of two nearby voids into a single stable void does not effect the total volume. However, volumes of individual voids could change drastically. The volume of the stable void in Figure 8(b) is approximately equal to sum of the volume of the unmerged voids. The volumes are verified against known results [1].

Robustness of (ε, π)-stable cavities. Figure 13 plots the number of (ε, π)-stable voids and π-persistent voids for various values of α for $\varepsilon = 1.0$ in case of 4HHB and 4B87 and $\varepsilon = 0.3$ in case of 2CI2. The number of (ε, π)-stable voids remain constant over an interval as the method considers all the creation/destruction events happening in the interval $[-\varepsilon, \varepsilon]$ for the computation. But the number of π-persistent voids vary as they are sensitive to creation/destruction of voids at a particular α value.

Validation of computed cavity volumes. There is usually a variation in the volumes of cavities computed using various methods [1]. This variation may arise due to the different models used for computing the volumes. Therefore, we perform an additional normalization of the computed volumes using model mutants [1] to eliminate such variations. We used 13 different model mutants to create a set of artificial voids. We use the resulting volumes to compute the linear normalization function, see Figure 14. The volumes computed using our computation is normalized as follows:

$$Volume = 2.54 \times ComputedVolume + 60.77.$$

In order to verify the correctness of the volumes computed by our software *RobustVoids*, we compare them to the volumes computed using *MCCavity* [1], see Figure 15. Note that we use normalised volumes in this graph.

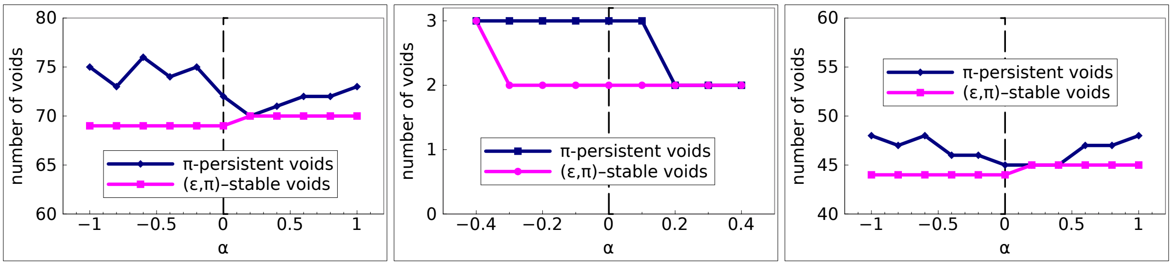
5 Conclusions

We have defined a novel notion of robust cavities that is insensitive to the perturbation of the atomic radii. Robust cavities are computed via a controlled modification of the filtration that represents the molecule and its cavities. Identifying robust cavities is important so that the biologist only targets these cavities in tedious mutation-based experiments.

The method addresses the inaccuracies in the measurements of the radii by selectively varying the radii for a specific set of atoms. But the positional uncertainties which arise due to the motion of the molecules is not addressed. We have been able to present some visual evidence that slight perturbations in the radii results in a larger cavity instead of smaller cavities. The value of ε is lower than the typical experimental error in crystallographic measurements. We plan to further investigate the relationship between the perturbation in the atom radii corresponding to the delayed simplex insertion and the structural and functional properties of the protein. Future work also includes generalizing the framework to use empirically determined intervals of radii for each atom type and addressing the issue of biological implications of the method.

Acknowledgements

This work was supported by a grant from Department of Science and Technology, India (SR/S3/EECE/0086/2012).



(a) Protein 4HHB

(b) Protein 2CI2

(c) Protein 4B87

Figure 13: Graphs showing the variation of the total volume of voids for constant ε and varying α.

References

[1] Chakravarty, S., Bhinge, A., and Varadarajan, R. A procedure for detection and quantitation of cavity volumes in proteins. *Journal of Biological Chemistry 277*, 35 (2002), 31345–31353.

[2] Dundas, J., Ouyang, Z., Tseng, J., Binkowski, A., Turpaz, Y., and Liang, J. CASTp: computed atlas of surface topography of proteins with structural and topographical mapping of functionally annotated residues. *Nucleic acids research 34*, 2 (2006), W116–W118.

[3] Edelsbrunner, H. *Weighted alpha shapes*. University of Illinois at Urbana-Champaign, Department of Computer Science, 1992.

[4] Edelsbrunner, H. Biological applications of computational topology. In *Handbook of Discrete and Computational Geometry*, J. E. Goodman and J. O'Rourke, Eds. CRC Press, 2004, pp. 1395–1412.

[5] Edelsbrunner, H. *Computational Topology. An Introduction*. Amer. Math. Soc., 2010.

[6] Edelsbrunner, H., and Fu, P. Measuring space filling diagrams and voids. Tech. rep., UIUC-BI-MB-94-01, Beckman Inst., Univ. Illinois, Urbana, Illinois, 1994.

[7] Edelsbrunner, H., Kirkpatrick, D., and Seidel, R. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory 29*, 4 (1983), 551–559.

[8] Edelsbrunner, H., and Koehl, P. The geometry of biomolecular solvation. *Combinatorial & Computational Geometry 52* (2005), 243–275.

[9] Edelsbrunner, H., Letscher, D., and Zomorodian, A. Topological persistence and simplification. *Discrete & Computational Geometry 28*, 4 (2002), 511–533.

[10] Edelsbrunner, H., and Mücke, E. Three-dimensional alpha shapes. *ACM Transactions on Graphics (TOG) 13*, 1 (1994), 43–72.

[11] Kim, D.-S., Cho, Y., Sugihara, K., Ryu, J., and Kim, D. Three-dimensional beta-shapes and beta-complexes via quasi-triangulation. *Computer-Aided Design 42*, 10 (2010), 911–929.

[12] Kim, D.-S., Ryu, J., Shin, H., and Cho, Y. Beta-decomposition for the volume and area of the union of three-dimensional balls and their offsets. *Journal of Computational Chemistry* (2012).

[13] Kim, D.-S., and Sugihara, K. Tunnels and voids in molecules via voronoi diagram. In *Proc. Symp. Voronoi Diagrams in Science and Engineering (ISVD)* (2012), pp. 138–143.

[14] Koehl, P., Levitt, M., and Edelsbrunner, H. Proshape: understanding the shape of protein structures. *Software at biogeometry.duke.edu/software/proshape* (2004).

[15] Krone, M., Falk, M., Rehm, S., Pleiss, J., and Ertl, T. Interactive exploration of protein cavities. In *Computer Graphics Forum* (2011), vol. 30, pp. 673–682.

[16] LIANG, J., EDELSBRUNNER, H., FU, P., SUDHAKAR, P., AND SUBRAMANIAM, S. Analytical shape computation of macromolecules: II. inaccessible cavities in proteins. *Proteins Structure Function and Genetics 33*, 1 (1998), 18–29.

[17] LIANG, J., EDELSBRUNNER, H., AND WOODWARD, C. Anatomy of protein pockets and cavities. *Protein Science 7*, 9 (1998), 1884–1897.

[18] LINDOW, N., BAUM, D., BONDAR, A., AND HEGE, H. Dynamic channels in biomolecular systems: Path analysis and visualization. In *Proc. IEEE Symposium on Biological Data Visualization (BioVis)* (2012), pp. 99–106.

[19] LINDOW, N., BAUM, D., AND HEGE, H. Voronoi-based extraction and visualization of molecular paths. *Visualization and Computer Graphics, IEEE Transactions on 17*, 12 (2011), 2025–2034.

[20] MUNKRES, J. *Elements of Algebraic Topology*, vol. 2. Addison-Wesley Menlo Park, CA, 1984.

[21] PARULEK, J., TURKAY, C., REUTER, N., AND VIOLA, I. Implicit surfaces for interactive graph based cavity analysis of molecular simulations. In *Biological Data Visualization (BioVis), 2012 IEEE Symposium on* (2012), pp. 115–122.

[22] PETŘEK, M., KOŠINOVÁ, P., KOČA, J., AND OTYEPKA, M. MOLE: A Voronoi diagram-based explorer of molecular channels, pores, and tunnels. *Structure 15*, 11 (2007), 1357–1363.

[23] PETŘEK, M., OTYEPKA, M., BANÁŠ, P., KOŠINOVÁ, P., KOČA, J., AND DAMBORSKỲ, J. Caver: a new tool to explore routes from protein clefts, pockets and cavities. *BMC bioinformatics 7*, 1 (2006), 316.

[24] TILL, M. S., AND ULLMANN, G. M. Mcvol-a program for calculating protein volumes and identifying cavities by a monte carlo algorithm. *Journal of molecular modeling 16*, 3 (2010), 419–429.