Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Original Paper

OXFORD

Structural bioinformatics

Protein Pocket Detection via Convex Hull Surface Evolution and Associated Reeb Graph

Rundong Zhao [1], Zixuan Cang [2], Yiying Tong [1]*, and Guo-Wei Wei [2]*

[1]Department of Computer Science and Engineering, Michigan State University, East Lansing, Michigan 48823, USA and [2]Department of Mathematics, Michigan State University, East Lansing, Michigan 48823, USA

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX;

Abstract

Motivation: Protein pocket information is invaluable for drug target identification, agonist design, virtual screening, and receptor-ligand binding analysis. A recent study indicates that about half holoproteins can simultaneously bind multiple interacting ligands in a large pocket containing structured subpockets. Although this hierarchical pocket and subpocket structure has a significant impact to multiligand synergistic interactions in the protein binding site, there is no method available for this analysis. This work introduces a computational tool based on differential geometry, algebraic topology, and physics-based simulation to address this pressing issue.

Results: We propose to detect protein pockets by evolving the convex hull surface inwards until it touches the protein surface everywhere. The governing partial differential equations (PDEs) include the mean curvature flow combined with the eikonal equation commonly used in the fast marching algorithm in the Eulerian representation. The surface evolution induced Morse function and Reeb graph are utilized to characterize the hierarchical pocket and subpocket structure in controllable detail. The proposed method is validated on PDBbind refined sets of 4,414 protein-ligand complexes. Extensive numerical tests indicate that the proposed method not only provides a unique description of pocket-subpocket relations, but also offers efficient estimations of pocket surface area, pocket volume, and pocket depth.

Availability: We will release the executable code upon acceptance.

Contact: ytong@msu.edu, wei@math.msu.edu

1 Introduction

The detection of pockets on protein surfaces is a prerequisite to various tasks in computational molecular biophysics and bioinformatics, such as the determination of the binding site when one attempts to dock a ligand to a protein target and the study of protein functional surfaces. Automatic procedures for potential pocket predictions have been evolving along with the advance in computational capability. Many methods have been designed for protein pocket determination and they can be classified as geometry-based, energy-based, sequence-based, or hybrid (Schmidtke *et al.*, 2011). We review several common categories of these geometry-based methods, namely, probe based methods, grid based methods, Voronoi diagram based methods, and marching surface methods that are relevant to our approach.

Based on the idea of rolling a probe to construct solvent excluded surfaces, many probe-based methods have been introduced to detect protein pockets. The pockets are captured by different behaviors with different probe radii. One type of such methods samples protein surfaces using many small probes, and then determines pockets according to surface depressions (Ruppert *et al.*, 1997; Del Carpio *et al.*, 1993; Brady and Stouten, 2000). Another type of such methods uses a large probe radius to create an envelope surface surrounding a protein surface, and then detect the hollow regions between the envelope and the protein surface(Yu *et al.*, 2009; Masuya and Doi, 1995; Nayal and Honig, 2006). There are also methods using combinations of both types of probes (Kawabata and Go, 2007).

The grid based methods, pioneered by Levitt and Banaszak (1992), place a protein inside a regular grid and then scan the grid in a specific order to mark grid points as inside pockets if certain criteria are satisfied (Hendlich *et al.*, 1997; Venkatachalam *et al.*, 2003; Weisel *et al.*, 2007; Hendlich *et al.*, 1997). For instance, grid points can be labeled as not

belonging to pockets either by a cube eraser Venkatachalam *et al.* (2003) or by a probe eraser Weisel *et al.* (2007). Kufareva *et al.* (2011) developed a grid potential to assist pocket extraction in grids. It is not only a geometry-based method but also an energy-based one.

Voronoi based methods, introduced by Liang *et al.* (1998), have been proposed to compare the differences between alpha shapes and the Delaunay triangulation (dual structure to Voronoi diagram) to find pockets, which are represented by the tetrahedra in Delaunay tessellation but not in alpha shapes. A new shape descriptor was later introduced to improve the overall efficiency of this approach (Xie and Bourne, 2007). Voronoi diagram was also used to detect depression regions Kim *et al.* (2008).

Marching surface methods, proposed by Kleywegt and Jones (1994), detect pockets as isolated cavities formed by offsetting a protein surface along outward normals at a uniform speed. Bock *et al.* (2007) proposed to trace points on surface along outward normal direction to check whether it has additional intersections with the protein surface, based on which protein surface regions are labeled as pocket or non-pocket.

There are multifunctional tools such as Castp:3.0 (Dundas *et al.*, 2006) and FPocket (Le Guilloux *et al.*, 2009) that additionally compute physicochemical properties and meta tools such as MetaPocket (Huang, 2009) that combine multiple approaches on top of the geometry. Owing to the advances in protein structural determination, databases about protein pockets and functional surfaces have been established, such as SitesBase database (Gold and Jackson, 2006). Structural databases of protein-ligand complexes (Wang *et al.*, 2004) can also be used to validate pocket detecting tools. Based on large annotated databases and efficient algorithms, web servers, such as PocketQuery (Koes and Camacho, 2012) and MSDmotif (Golovin and Henrick, 2008), have been developed for large scale pocket search.

However, many problems in protein pocket detection remain unsolved. New analysis based on different sequence identity thresholds of a non-redundant set of all holo structures in the PDB indicates that between 47 - 76% of holoproteins can simultaneously bind multiple, interacting ligands in the same pocket that may be comprised of several small but significant subpockets (Tonddast-Navaei *et al.*, 2017). The detailed understanding of protein-multiligand binding remains of profound importance on many fronts, not least of which includes drug discovery. The hierarchical structure between pockets and subpockets is a key to the understanding of the binding of multiple interacting ligands Tonddast-Navaei *et al.* (2017). Unfortunately, none of the aforementioned methods is designed to describe the hierarchical structure of protein pockets. Additionally, the analysis of protein-ligand binding and drug targets requires computational tools that are able to not only detect protein pockets but also provide more geometric details, including possible subpockets and pocket area, volume, and depth. Although grid based methods can provide a rough estimate for pocket volume, they typically suffer from ~~accuracy and~~ efficiency issues. These algorithms usually use the entire grid for the calculation, incurring extra memory consumption and computation time on grid cells far from the protein surface. Further process based on the whole grid will also introduce huge time complexity. Voronoi diagram based methods are efficient in providing area and volume estimates, but lack depth information. Finally, the performance of many current methods depends on many parameters that are not intuitive to tune for given specific pocket requirements. The objective of the present work is to address these difficulties by using geometric partial differential equation (PDE) and algebraic topology.

Inspired by a physical simulation used for surface coloring in 3D printing, in which air pockets are detected and treated (Zhang *et al.*, 2017), we start from a convex hull surface wrapping around a protein, and then press the surface inward until it is tightly in contact with the protein. The space between the convex hull surface and the protein surface is potential locations of pockets, and we use the time that the deforming surface passes through the point as a Morse function to build an evolving topological structure that helps define a pocket hierarchy with desired information.

Lagrangian (mesh) representations are often used in surface deformation as in (Zhang *et al.*, 2017). We opted for an Eulerian (grid) representation, due to the complex surface geometry of the protein, large distortion and potential topological change, which are difficult to handle with a mesh. We encode the surface with an implicit function on a Cartesian grid. This type of methods was originally introduced in simulating two-phase flow by Sussman *et al.* (1994). The interface can be defined by the zero level set of an implicit function which has a good control flexibility (Peng *et al.*, 1999; Osher and Fedkiw, 2003). We simplify the procedure significantly for efficiency, by combining a simple surface offsetting and mean curvature flow to achieve our goal.

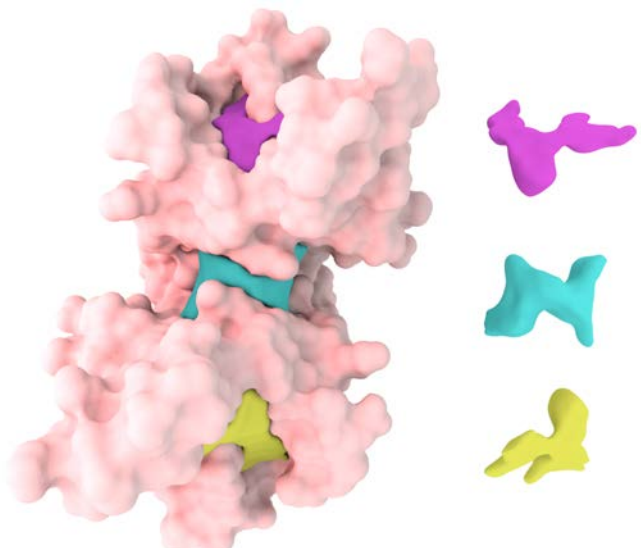


Fig. 1: Illustration of detected pockets of protein 1a4r showed by different colors.

To detect protein pocket hierarchies associate with geometric PDEs, we use persistent homology in the cubical setting. Persistent homology has flourished recently for analyzing geometry and topology of certain space. Early effort dealt with 0-th order topological persistence (Frosini and Landi, 1999), while high dimension topological persistence was formulated by Edelsbrunner *et al.* (2000). General mathematical theory of persistent homology has been developed by Zomorodian and Carlsson (2005). An efficient software for computing persistent homology on filterations of simplicial complexes and cubical complexes has been developed (Mischaikow and Nanda, 2013). While researchers keep enriching persistent homology theory, its practical applications in biomolecular analysis and landscape analysis have been developed (Xie and Bourne, 2007; Xia *et al.*, 2015). Differential geometry based persistent homology was proposed to proactively predict fullerene isomer curvature stability (Wang and Wei, 2016). Topological landscape tool was built to analyze real world terrain model (Harvey and Wang, 2010). In our approach, as the convex hull surface is deformed, we analyze the persistence of the 0-th dimensional topological invariant induced by the moving surface level set to detect potential pocket (equivalent and dual to membranes around cavities formed between the deforming surface and the protein surface). This approach enables us to analyze pocket area, volume, depth and hierarchical pocket-subpocket relation.

The rest of the paper is organized as follows. Section 2 discusses the preliminary mathematical background. Section 3 introduces the overall procedures. The implementation of our algorithms is given in Section 4.

Section 5 presents the results and applications of the proposed protein pocket detection method. This paper concludes in Section 6.

2 Math Background

2.1 Signed Distance Function

We consider a real-valued function ϕ defined on a regular Cartesian grid. An implicit surface is defined by the level set

$$\Gamma = \{\mathbf{r} \mid \phi(\mathbf{r}) = 0,\ \mathbf{r} \in \mathbb{R}^3\}, \tag{1}$$

which is our surface in the Eulerian form. It is possible to take a Lagrangian mesh as the input surface, since the conversion is a standard routine. During surface deformation, we rely on the Eulerian representation to handle the inevitable topological changes. Level set propagation is governed by a general level set equation

$$\frac{\partial \phi}{\partial t} + \mathbf{v} \cdot \nabla \phi = 0, \tag{2}$$

where $\mathbf{v}$ is the velocity of the flow. As tangential velocity does not change the shape, we can describe surface deformation by the normal component without loss of generality. Thus, one can rewrite the velocity field $\mathbf{v}$ as $v\mathbf{n}$, where $\mathbf{n} = \frac{\nabla \phi}{|\nabla \phi|}$, $|v|$ is the propagation speed and the sign of v indicates inward or outward motion. The level set equation can be rewritten as

$$\frac{\partial \phi}{\partial t} + v|\nabla \phi| = 0. \tag{3}$$

For uniform offset, we can set v to a constant c. A typical surface smoothing deformation is achieved by the mean curvature flow, which offsets each surface point at the speed given by the mean curvature, i.e., $v = -H = -\nabla \cdot \mathbf{n}$. The mean curvature flow level-set equation is given by Osher and Fedkiw (2003)

$$\frac{\partial \phi}{\partial t} - |\nabla \phi| \left(\nabla \cdot \frac{\nabla \phi}{|\nabla \phi|} \right) = 0. \tag{4}$$

We can simplify the above two flows if $|\nabla \phi| = 1$, which can be achieved by choosing ϕ to be the signed distance function, i.e., $|\phi(\mathbf{r})|$ stores the distance from $\mathbf{r}$ to the zero level set, with its sign being positive (negative) for outside (inside) locations. As such, the constant-speed normal flow is given by

$$\frac{\partial \phi}{\partial t} + c = 0, \tag{5}$$

and the mean curvature flow becomes

$$\frac{\partial \phi}{\partial t} - \Delta \phi = 0. \tag{6}$$

The use of the mean curvature flow for biomolecular surface generation was introduced by Bates *et al.* (2008). Our procedure will drive the surface inward, so the constant c is negative.

Before propagating the zero level set, we first initialize the signed distance function ϕ by the eikonal equation to transform the Lagrangian mesh Γ which is the boundary of a 3D domain Ω into an Eulerian grid embedded signed distance function,

$$|\nabla \phi(\mathbf{r})| = 1, \mathbf{r} \in \Omega \subset \mathbb{R}^3 \tag{7}$$

with boundary condition

$$\phi|_{\Gamma = \partial \Omega} = 0. \tag{8}$$

Fast marching method (FMM), which shares similar ideas from the Dijkstra algorithm, is commonly used to solve the eikonal equation on a regular grid (Sethian, 1996). Alternatively, fast sweeping method can be used (Zhao, 2005). When the regular grid is large, solving this problem in the whole grid is inefficient for both space and time. Typically, a narrow band is used to reduce the memory size. We specify a distance threshold w. Any voxel with a distance above the threshold w will not be used in the calculation. We use the typical choice of $w = 3$, which guarantees the accurate solution allowed by the resolution of the grid, since the gradient will be correctly calculated for the 0-th level set. Using any larger w will only slow down the calculation without changing the results.

We evolve an initial surface inward without creating sharp corners, so we iteratively update the sign distance function via Eqs. (5) and (6). The normal flow guarantees that the zero level set moves inward while the mean curvature flow offers a smooth surface representation. The property of $|\nabla \phi| = 1$ is fundamental in simplifying our updating equations. However, the mean curvature flow makes ϕ deviate from a signed distance function. As typically done in level set methods, we reinitialize the signed distance function by solving the eikonal equation with the zero level set as the boundary every few iterations.

2.2 Persistent Homology

Another technique we employ in our algorithm is persistent homology, a widely applied algebraic topology tool for data analysis, especially in the field of computational biology and chemistry. It significantly reduces geometric complexity by representing essential geometric properties in terms of a sequence of topological invariants parameterized by a geometric function.

2.2.1 Homology Group

For a topological space $\mathcal{X}$, we define a series of complexes $\mathcal{C}_i(\mathcal{X}), i = 0, 1, 2...$ describing different dimensional information of the topological space. Each complex is an Abelian group. The complexes are linked by the boundary maps, which include the homeomorphisms $\partial_i : \mathcal{C}_i \to \mathcal{C}_{i-1}$ satisfying the condition

$$\partial_{i-1} \circ \partial_i = 0,\ i \in \mathbb{Z}, i > 0. \tag{9}$$

The algebraic construction by connecting the complexes by the maps is called a chain complex,

$$\cdots \xrightarrow{\partial_{i+1}} \mathcal{C}_i \xrightarrow{\partial_i} \mathcal{C}_{i-1} \xrightarrow{\partial_{i-1}} \cdots \xrightarrow{\partial_2} \mathcal{C}_1 \xrightarrow{\partial_1} \mathcal{C}_0 \xrightarrow{\partial_0} 0 \tag{10}$$

The i-th homology is constructed based on two subsets of complex $\mathcal{C}_i$, the boundary $\mathrm{Im}(\partial_{i+1})$, the image of map ∂_{i+1}, and the cycle group $\mathrm{Ker}(\partial_i)$, the kernel of map ∂_i. The property in Eq. (9) implies that

$$\mathrm{Im}(\partial_{i+1}) \subseteq \mathrm{Ker}(\partial_i) \tag{11}$$

More precisely, the homology group is defined as the quotient group

$$\mathcal{H}_i(\mathcal{X}) = \frac{\mathrm{Ker}(\partial_i)}{\mathrm{Im}(\partial_{i+1})}. \tag{12}$$

When $\mathcal{C}_i$ are generated by i-dimensional cells of a tessellation of $\mathcal{X}$, homology provides topological information of $\mathcal{X}$. Intuitively, $\mathcal{H}_i(\mathcal{X})$ contains independent i-dimensional (i-D) holes in $\mathcal{X}$.

For instance, the quotient group $\mathcal{H}_1$ of a torus describes holes on it. It is constructed from $\mathrm{Im}(\partial_2)$, the group of 1D curves that are boundaries of certain 2D subspaces of $\mathcal{X}$, and $\mathrm{Ker}(\partial_1)$, the group of all closed 1D curves. There are 2 independent types of closed 1D curves that are not a boundary curve of $\mathcal{X}$, which are the generators of the homology. This, in fact, shows the 1D topological features, a loop around the tunnel and another around the handle of the torus.

2.2.2 Persistent homology

In order to provide relevant geometric information, a geometric parameter can be introduced to provide a dynamic homology analysis for a topology space through filtration, which is a series of subspace $\mathcal{X}_i$ of $\mathcal{X}$,

$$\emptyset = \mathcal{X}_0 \subseteq \mathcal{X}_1 \subseteq \mathcal{X}_2 \subseteq \cdots \subseteq \mathcal{X}_m = \mathcal{X}. \tag{13}$$

For our evolving surface, the index is related to the time parameter. A homology class $\delta \in \mathcal{H}_k(\mathcal{X}_i)$ is referred to be born at time i if it is not an image from the inclusion map from $\mathcal{X}_{i-1}$ to $\mathcal{X}_i$, and to die at time j if it is no longer in the image of the inclusion map from $\mathcal{X}_i$ to $\mathcal{X}_j$. The time interval $j-i$ is called the persistence. See, e.g., (Edelsbrunner *et al.*, 2000; Wang and Wei, 2016) for additional details. A major topological feature will have a long persistence. Thus, geometric PDEs can induce persistence to provide a robust description of protein pocket topological features.

3 Algorithm

The relatively simple pockets and their areas, volumes, depths and pocket-subpocket relations, can be characterized by the persistence of only homology group $\mathcal{H}_0$, which in fact describes the connected components for the topological space. Describing ring-like pockets can be performed by homology group $\mathcal{H}_1$, but detecting protein cavities requires a different set of the geometric PDEs and would be beyond the scope of the present work.

As we use regular Cartesian grid, cubical complexes and persistent homology at the cubical setting are employed. The associated filtration can be created by a Morse function $T(\mathbf{r})$ stored on the 3D grid, with subspaces

$$\mathcal{X}_i = \{\mathbf{r} \mid T(\mathbf{r}) \leq t_i = ih\}, \tag{14}$$

where h is the time step size.

For a deforming surface, we can define the Morse function through $T(\mathbf{r}) = \inf\{t \mid \phi(t, \mathbf{r}) = 0\}$, i.e., the time when the surface first sweeps through the location $\mathbf{r}$.

One option to evolve the surface is to start from the protein surface and move outward, but the PDEs involved are less stable than those for moving the convex hull inward. Moreover, the time $T(\mathbf{r})$ for the inward motion with unit speed also provides a better depth estimate. We prevent the evolving surface from entering the protein surface since we are looking for pockets outside the protein. In this case, the total space $\mathcal{X}$ is the space between the protein and its convex hull. As the surface moves inward, $\mathcal{X} \backslash \mathcal{X}_i$ is shrinking, and it will be separated by protein surface, forming connected components (pieces of the hollow space between the protein surface and the deforming surface at time t_i).

We define these pieces with long persistence as potential protein pockets. This procedure can be equivalently, and more efficiently described by a Reeb graph, describing the splitting and merging of the connected components of level sets of $T(\mathbf{r})$. More precisely, the Reeb graph contains nodes, each of which represents a connected component of $\{\mathbf{r} \mid T(\mathbf{r}) = t_i\}$ for certain time t_i, and edges connecting nodes at t_i and t_{i+1} if they are connected through $\{\mathbf{r} \mid t_i \leq T(\mathbf{r}) \leq t_{i+1}\}$. For our purpuse, we only need to construct the Reeb graph to infer potential protein pockets and subpockets.

For our Morse function $T(\mathbf{r})$, the Reeb graph is simply a tree. Starting from a single root, the tree will bifurcate whenever there is a splitting of the connected components. Finally, all connected components will disappear when the surface has deformed to the protein surface.

With persistent homology, we can actually capture all potential pockets regardless of their sizes, and the tree provides us with a hierarchy among the pocket candidates. Then, we can use arbitrary geometric or physical pocket dimensions to eliminate those with short persistence as "noise". We elaborate on capturing pockets with high probability by further examining the geometry in the next section.

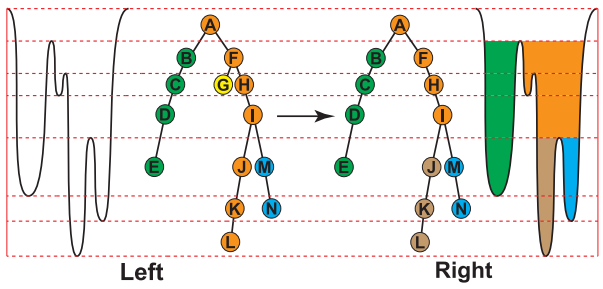


Fig. 2: **Left:** Illustration of Reeb graph. The dashed line represents critical times for the filtration when there will be components newly born or killed. We have extracted 4 components labeled by different colors. **Right:** Illustration of a trimmed Reeb graph. The component (i.e., the yellow leaf) that lives for a short period is eliminated. Note that orange path is divided into 2 components (orange and brown), due to pocket hierarchical relation. Persistent objects are then marked by green, orange, brown and blue regions extracted from nodes B, F, J and M respectively. Brown and blue are sub-pockets of orange.

4 Implementation

Algorithm 1 Pocket Detection Algorithm

```
1: function PocketDetection(model, atoms)
2:     BuildConvexHull()
3:     BuildSignedDistanceFunction()
4:     Initialize()
5:     while NotAllSurfaceBlocked() do          ▷ Fig. 4
6:         ReinitializeSDFIfNeeded()            ▷ Sec. 2.1
7:         EvolveSurface()
8:         ExtractConnectedComponents()
9:     BuildReebGraph()
10:    ExtractMajorPersistencePath()
11:    ExtractPotentialPockets()
```

Proper implementation is mandatory for efficiency of Eulerian methods. To reduce memory space usage, we perform a two-pass algorithm to avoid storing the Morse function explicitly in the 3D grid. In the first pass, we record only the necessary information to build the Reeb graph and extract the major component paths. We then collect the geometric information for the long persistent pockets by evolving the surface with a second pass.

4.1 Input and Output

Our algorithm is independent of the type of input surface, e.g.,~~van der Waals surface, solvent accessible surface or~~ solvent excluded surface (SES). A triangulated SES can be computed by software provided by (Liu *et al.*, 2017). We also use a standard molecule description file, containing the locations and radii of all the atoms for future atom query.

The output provides information on protein pocket candidates, including the depth, area, volume and adjacent atoms for downstream applications. The first three geometry properties are obtained by analyzing the

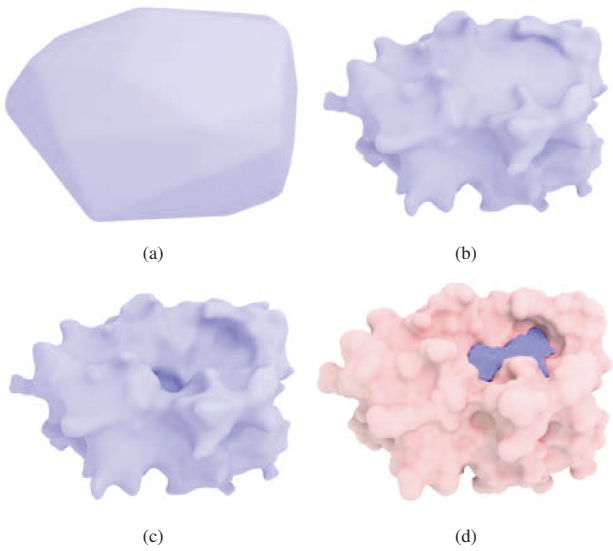


Fig. 3: Illustration of the convex hull surface evolution on protein 3kgp. The surface moves inward from the convex hull and finally reaches the protein surface in (a), (b), and (c). (d) shows the detected pockets.

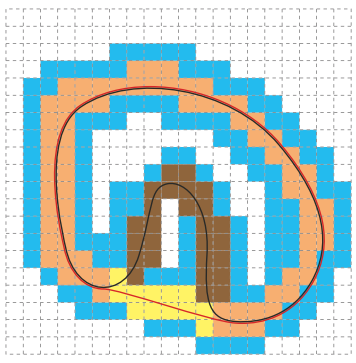


Fig. 4: Illustration of basic algorithmic concepts. All colored voxels are active, and the rest inactive. Orange voxels are blocked surface voxels, while Yellow voxels are free surface voxels of the deforming surface (indicated by red curve). The black curve indicates the protein surface. The brown voxels represent the currently untouched region of protein surface, which is a pocket in this case.

space bounded by the deforming surface and the protein surface. We build a kd-tree for fast access of nearest atom.

4.2 Initialization

Open source software packages exist for convex hull surface generation and solving the eikonal equation. We resort to the Computational Geometry Algorithms Library (CGAL) (Fabri and Pion, 2009) for building the convex hull surface from a triangulated SES (Liu *et al.*, 2017) and OpenVDB (Museth, 2013) for the data structures and subroutines of surface deformation. A surface mesh can be converted into a signed distance function by using an OpenVDB procedure. OpenVDB uses a hierarchical tree structure to achieve narrow band storage, which contributes to the overall efficiency of our implementation.

4.3 Evolving Surface

With the narrow band representation of the signed distance function (SDF), moving the surface only amounts to update ϕ in each active voxel. We mark each active deforming surface voxel as either blocked or free depending on whether the deforming surface is touching the protein surface at that voxel, which can be determined by comparing the signed distance functions for the deforming surface and for the protein surface. We update ϕ for the moving surface only in free voxels and change the signed distance function monotonically in time to prevent moving the surface backwards. The monotonicity prevents the mean curvature flow from overpowering the normal flow motion, while preventing sharp corners from developing near contact regions of the two surfaces. As mentioned before, reinitialization for every few update steps is necessary, since otherwise the level set function will deviate from an SDF.

4.4 Connected Component

As mentioned above, the connected components of $\mathcal{X}$ in the filtration is memory-intensive to compute. Thus, we opt for the equivalent calculation based on surface voxels, which are the active voxels containing a piece of the current zero level set. We then compute the connected components of surface voxels that are not blocked by the protein surface yet.

The idea is illustrated in Fig. 4 in 2-dimension, a snapshot of active voxels during the surface deformation. The black curve represents the protein surface, and the red curve represents the deforming surface. Note that for stable implementation, we start the deforming surface from a surface slightly offset outward from the convex hull. Both the deforming surface and the protein surface are stored as zero-level sets of the corresponding signed distance functions. All colored voxels are active. Orange and yellow voxels are surface voxels of the deforming surface, and brown voxels are surface voxels of the protein surface. Orange voxels are blocked by the protein surface, but yellow voxels are still free to move. We further allow the deforming surface to move within the protein surface by a short distance, again for robustness. The voxels between brown voxels and yellow voxels belong to a potential pocket. The free moving piece of the deforming surface will continue evolving inward until it becomes blocked the protein surface.

4.5 Reeb Graph

We construct the Reeb graph, based on connected components. The persistence of branches in the Reeb graph indicates how likely it corresponds to a real protein pocket. As explained in Sec. 3, nodes of the Reeb graph corresponds to connected components, and edges show their connection through temporal evolution of the surface. As we use a nearly uniform unit speed to evolve the surface along the normal directions, except for small deviations introduced by the mean curvature flow, the persistence well captures the depth information.

Each node is labeled with a persistence computed as the graph distance from the deepest leaf node among its descents. Branches with a small persistence can be trimmed. This does not prevent deep but narrow candidate pockets from being detected. However, the estimated free moving surface area associated with the component can be used as an additional criterion to eliminate those candidates. So both the depth and width thresholds can be easily specified and applied. Finally, we just need to run the second pass to extract the desirable pocket information.

4.6 Geometric feature

Our surface deformation procedure can easily produce geometric features for detected pockets, as each pocket is represented by space bounded by protein surface patches and deforming surface patches, rendering the pocket volume and pocket surface area. We can also extract the opening

area by the area of the deforming surface patch, which indicates the pocket width. Pocket depth is naturally defined by the persistence of a certain pocket. More precisely, the depth of a pocket is defined by the persistence measuring the difference between birth and death times multiplied by the surface evolution speed, which is 0.5 times the grid spacing in our implementation.

Such volume and area calculation for level sets is well established. Here we offer a highly efficient estimation. We simply count the number of voxels that are bounded by the two surfaces as an estimate for volume. The pocket area and horizontal span are estimated by the corresponding surface voxel counts on protein surface patches and deforming surface patches, respectively. We only provide a rough estimate of the surface area, but more accurate results can be calculated as efficiently by weighting different types of surface voxels as in (Mullikin and Verbeek, 1993). Since the voxel count times the volume of voxel provides the volume of a thin shell of about 2 grid spacing, we estimate the area by dividing this volume by this approximate thickness of the thin shell.

All our thresholds, the minimum required depth, the minimum required horizontal span, and the minimum required volume, are all intuitive parameters, that can be either user-specified or application-determined. The final detected pockets will thus not be too shallow, too narrow or too small.

5 Results and Discussion

We validate our method with pocket detection performed on the PDBbind database (Wang *et al.*, 2004) which contains high quality crystal structures of diverse protein-ligand complexes. A residue or a ligand can be represented as sets of atoms, $R = \{a_i\}_i$ or $L = \{b_j\}_j$. A protein can then be represented as a set of residues $P = \{R_i\}_i$. All protein atoms are considered. Then we define a set of confirmed pocket residues within a distance d from the surface as

$$\mathrm{POC}(P, L, d) = \left\{ R_i \in P \;\middle|\; \min_{a \in R_i, b \in L} \|a - b\| \le d \right\}. \quad (15)$$

Let $\mathrm{POC}_{\mathrm{comp}}(P)$ be the set of residues in P that are identified as pockets by the program. We say the pocket detection succeeds for a protein if

$$R(P, d) = |\mathrm{POC}_{\mathrm{comp}}(P) \cap \mathrm{POC}(P, L, d)| / |\mathrm{POC}(P, L, d)| \ge r, \quad (16)$$

where r is a ratio (required recall rate). The success rate $S(\mathcal{P}, d, r) = |\{P \in \mathcal{P} : R(P, d) \ge r\}|/|\mathcal{P}|$ is the percentage of proteins that our method succeeded to detect the pockets.

One set of proteins and its two subsets are used for validation. The first one containing 4,414 entries is the union of all proteins from the PDBbind refined sets v2007, v2013, v2015, and v2016, and is denoted $\mathcal{P}_{\mathrm{all}}$. The second set containing 2,430 entries is the subset of $\mathcal{P}_{\mathrm{all}}$ containing all single chain proteins denoted $\mathcal{P}_{\mathrm{sc}}$. The third set containing 290 entries is the PDBbind 2016 core set denoted $\mathcal{P}_{\mathrm{cr16}}$. The atomic radii are first generated by PDB2PQR software (version 2.1.0) (Dolinsky *et al.*, 2007) with CHARMM force field. The pockets are computed for the chain closest to the ligand if a protein contains multiple chains. The performance of the proposed method on the three sets is shown in Table 1.

Our method successfully captures the majority of the real binding pockets in Table 1. We found that there are three cases where our method cannot detect the provided ligand binding references. 1) The ligand is bound at a rather shallow place. 2) The ligand is bound at pockets which are formed by more than a single chain. 3) The ligand is bound at closed cavities, which is beyond the cases that our current method handles. Note that the success rate may appear to drop with increasing d in some cases because the denominator $|\mathrm{POC}(P, L, d)|$ may increase.

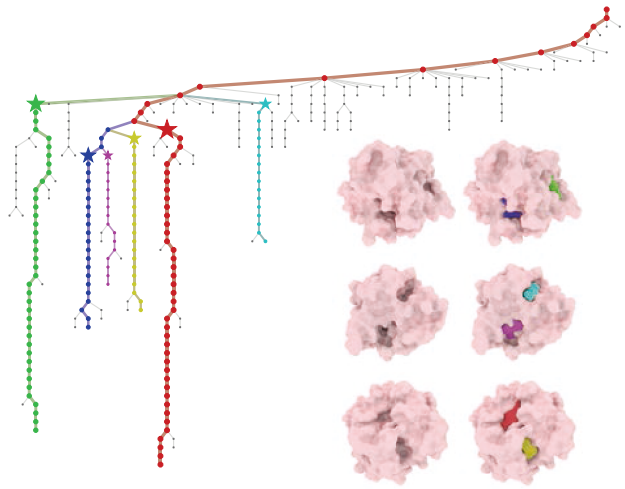


Fig. 5: Visualization of detected pockets of protein 3ao4 with the corresponding Reeb graph.

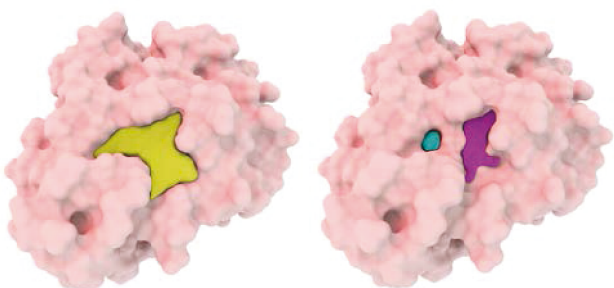


Fig. 6: Visualization of ligand interaction suggestions for multi-ligand binding on protein 1tok. (a) a large detected pocket (yellow). (b) two subpockets (cyan and purple) that bifurcate from this large pocket. (c) the corresponding branches in the Reeb graph. Yellow branch bifurcates into two subbranches (cyan and purple). (d) our suggestion for multi-ligand (red and green) binding with ligand interactions.

In addition to the known pockets, we are able to provide many other pocket candidates with detailed geometric information. For example, in Fig. 5, in addition to the binding site of protein 3ao4 confirmed by PDBbind database marked purple, our method also provides other potential candidates.

	$\mathcal{P}_{\text{all}}$ (4414)			$\mathcal{P}_{\text{sc}}$ (2430)			$\mathcal{P}_{\text{cr16}}$ (290)		
d \ r	0.25	0.5	0.75	0.25	0.5	0.75	0.25	0.5	0.75
3Å	0.91	0.86	0.78	0.94	0.89	0.83	0.95	0.89	0.81
4Å	0.91	0.86	0.76	0.94	0.89	0.80	0.95	0.89	0.77
5Å	0.91	0.86	0.68	0.94	0.89	0.71	0.94	0.90	0.71

Table 1. Performance measured by $S(\cdot, d, r)$ on the three sets with different distance thresholds (d) and ratio cutoffs (r)

Pocket	Volume(Å^3)	Area(Å^2)	Depth(Å)
1a4r top	964	475	4
1a4r mid	1227	558	5
1a4r bottom	935	463	4
3kgp	973	436	8
3ao4 blue	569	326	9
3ao4 green	521	293	5
3ao4 cyan	508	266	7
3ao4 purple	672	373	9
3ao4 red	828	409	7
3ao4 yellow	447	243	5
1tok yellow	1252	600	9
1tok cyan	533	272	7
1tok purple	173	90	6

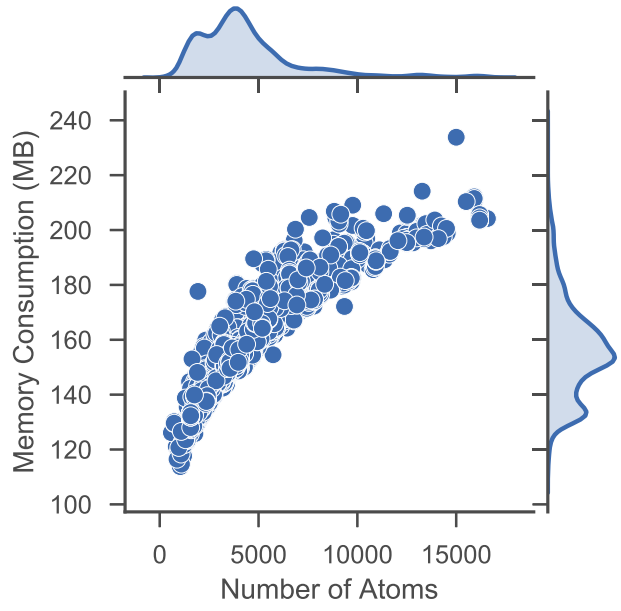
Table 2. Geometric properties of all detected pockets in figures.

Fig. 5 shows a specific example of the detected pockets for protein 3ao4. The colored branches in the Reeb graph are among the major persistent candidates, whereas gray paths are eliminated as noise. The color of the major component path is consistent with that for pockets. The pockets are extracted at the stage marked by a star. It can be noticed that pockets detected are highly reliable and resistant to noise. Figure 6 shows that our hierarchical detection procedure finds two subpockets (cyan and purple) from a large ancestor pocket (yellow), from which multi-ligand binding with ligand interactions may be suggested (red and green).

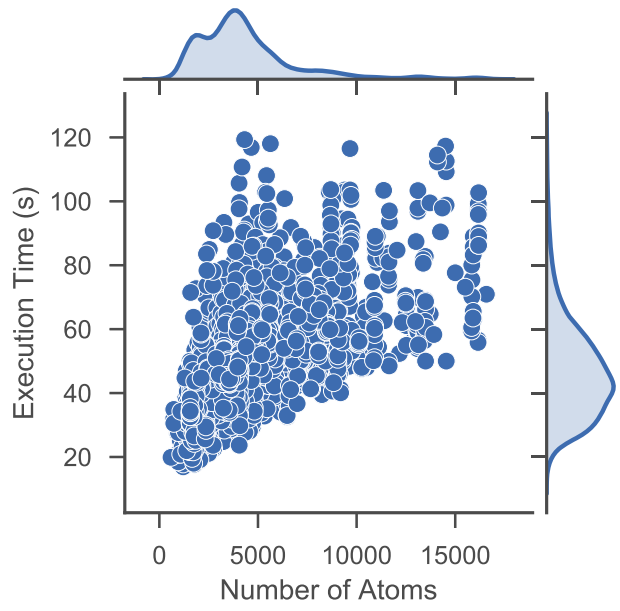
Table 2 provides details of geometric properties for all pockets in figures. We also provide statistics for all the test cases. Fig. 7(a) shows memory consumption distribution, which is roughly proportional to $O(\sqrt{n})$, where n is the number of atoms. Fig. 7(b) shows execution time distribution, which is within a reasonable amount of time, no more than 120 seconds.

6 Conclusion

This work introduces the geometric partial differential equation (PDE) based convex hull surface evolution and associated topological persistence for accurate, efficient and robust detection of protein pockets. The level set function is governed by the unit speed normal flow to measure the pocket surface area, volume, and depth. The mean curvature flow is incorporated to ensure a smooth surface representation of protein pockets. These equations are iteratively integrated in the Eulerian representation to allow potential topological changes. The transformation from Lagrangian mesh to the Cartesian grid is accomplished via the eikonal equation. The convex hull surface evolution naturally induces a Morse function and topological persistence. The resulting Reeb graph is utilized to describe the hierarchical relation between protein pockets and subpockets, a crucial information for protein-multiligand interactions that is not available ever before. Topological persistence also enables the classification and visualization of significant and insignificant pockets and subpockets.



(a)



(b)

Fig. 7: Statistics for all 4,414 test cases.

Three intuitive parameters describing geometric features are designed for user interaction and control. Efficient algorithms are carefully implemented to avoid potentially excessive memory consumption or execution time pitfalls. On a regular CPU (Intel Xeon 3.77GHz), the user can obtain results in about a minute without the need to worry about computational resource limitation. Our method has a high locality, which means that the efficiency can be further improved significantly by parallel computing techniques either with GPU such as CUDA, or CPU such as TBB. The resulting implementation of our method exhibits high accuracy in pocket detection in our tests. One limitation of our method is that we do not incrementally handle deforming flexible proteins, but we can treat them

frame by frame and establish the correspondence by mapping the pockets to atoms.

References

Bates, P., Wei, G.-W., and Zhao, S. (2008). Minimal molecular surfaces and their applications. *Journal of Computational Chemistry*, **29**(3), 380–391.

Bock, M., Garutti, C., and Guerra, C. (2007). Effective labeling of molecular surface points for cavity detection and location of putative binding sites. volume 6, pages 263–274.

Brady, G. P. and Stouten, P. F. (2000). Fast prediction and visualization of protein binding pockets with pass. *Journal of computer-aided molecular design*, **14**(4), 383–401.

Del Carpio, C. A., Takahashi, Y., and Sasaki, S.-i. (1993). A new approach to the automatic identification of candidates for ligand receptor sites in proteins:(i) search for pocket regions. *Journal of molecular graphics*, **11**(1), 23–29.

Dolinsky, T. J., Czodrowski, P., Li, H., Nielsen, J. E., Jensen, J. H., Klebe, G., and Baker, N. A. (2007). Pdb2pqr: expanding and upgrading automated preparation of biomolecular structures for molecular simulations. *Nucleic acids research*, **35**(suppl_2), W522–W525.

Dundas, J., Ouyang, Z., Tseng, J., Binkowski, A., Turpaz, Y., and Liang, J. (2006). Castp: computed atlas of surface topography of proteins with structural and topographical mapping of functionally annotated residues. *Nucleic acids research*, **34**(suppl_2), W116–W118.

Edelsbrunner, H., Letscher, D., and Zomorodian, A. (2000). Topological persistence and simplification. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 454–463. IEEE.

Fabri, A. and Pion, S. (2009). Cgal: The computational geometry algorithms library. In *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*, pages 538–539. ACM.

Frosini, P. and Landi, C. (1999). Size theory as a topological tool for computer vision. volume 9, pages 596–603.

Gold, N. D. and Jackson, R. M. (2006). Sitesbase: a database for structure-based protein–ligand binding site comparisons. *Nucleic acids research*, **34**(suppl_1), D231–D234.

Golovin, A. and Henrick, K. (2008). Msdmotif: exploring protein sites and motifs. *BMC bioinformatics*, **9**(1), 312.

Harvey, W. and Wang, Y. (2010). Topological landscape ensembles for visualization of scalar-valued functions. In *Computer Graphics Forum*, volume 29, pages 993–1002. Wiley Online Library.

Hendlich, M., Rippmann, F., and Barnickel, G. (1997). Ligsite: automatic and efficient detection of potential small molecule-binding sites in proteins. *Journal of Molecular Graphics and Modelling*, **15**(6), 359–363.

Huang, B. (2009). Metapocket: a meta approach to improve protein ligand binding site prediction. *OMICS A Journal of Integrative Biology*, **13**(4), 325–330.

Kawabata, T. and Go, N. (2007). Detection of pockets on protein surfaces using small and large probe spheres to find putative ligand binding sites. *Proteins: Structure, Function, and Bioinformatics*, **68**(2), 516–529.

Kim, D., Cho, C.-H., Cho, Y., Ryu, J., Bhak, J., and Kim, D.-S. (2008). Pocket extraction on proteins via the voronoi diagram of spheres. *Journal of Molecular Graphics and Modelling*, **26**(7), 1104–1112.

Kleywegt, G. J. and Jones, T. A. (1994). Detection, delineation, measurement and display of cavities in macromolecular structures. *Acta Crystallographica Section D: Biological Crystallography*, **50**(2), 178–185.

Koes, D. R. and Camacho, C. J. (2012). Pocketquery: protein–protein interaction inhibitor starting points from protein–protein interaction structure. *Nucleic acids research*, **40**(W1), W387–W392.

Kufareva, I., Ilatovskiy, A. V., and Abagyan, R. (2011). Pocketome: an encyclopedia of small-molecule binding sites in 4d. *Nucleic acids research*, **40**(D1), D535–D540.

Le Guilloux, V., Schmidtke, P., and Tuffery, P. (2009). Fpocket: an open source platform for ligand pocket detection. *BMC bioinformatics*, **10**(1), 168.

Levitt, D. G. and Banaszak, L. J. (1992). Pocket: a computer graphies method for identifying and displaying protein cavities and their surrounding amino acids. *Journal of molecular graphics*, **10**(4), 229–234.

Liang, J., Woodward, C., and Edelsbrunner, H. (1998). Anatomy of protein pockets and cavities: measurement of binding site geometry and implications for ligand design. *Protein science*, **7**(9), 1884–1897.

Liu, B., Wang, B., Zhao, R., Tong, Y., and Wei, G.-W. (2017). Eses: Software for eulerian solvent excluded surface. *Journal of Computational Chemistry*, **38**(7), 446–466.

Masuya, M. and Doi, J. (1995). Detection and geometric modeling of molecular surfaces and cavities using digital mathematical morphological operations. *Journal of molecular graphics*, **13**(6), 331–336.

Mischaikow, K. and Nanda, V. (2013). Morse theory for filtrations and efficient computation of persistent homology. *Discrete & Computational Geometry*, **50**(2), 330–353.

Mullikin, J. C. and Verbeek, P. W. (1993). Surface area estimation of digitized planes. *Bioimaging*, **1**(1), 6–16.

Museth, K. (2013). Vdb: High-resolution sparse volumes with dynamic topology. *ACM Trans. Graph.*, **32**(3), 27:1–27:22.

Nayal, M. and Honig, B. (2006). On the nature of cavities on protein surfaces: application to the identification of drug-binding sites. *Proteins: Structure, Function, and Bioinformatics*, **63**(4), 892–906.

Osher, S. and Fedkiw, R. (2003). Implicit functions. In *Level Set Methods and Dynamic Implicit Surfaces*, pages 3–16. Springer.

Peng, D., Merriman, B., Osher, S., Zhao, H., and Kang, M. (1999). A pde-based fast local level set method. *Journal of computational physics*, **155**(2), 410–438.

Ruppert, J., Welch, W., and Jain, A. N. (1997). Automatic identification and representation of protein binding sites for molecular docking. *Protein Science*, **6**(3), 524–533.

Schmidtke, P., Bidon-Chanal, A., Luque, F. J., and Barril, X. (2011). Mdpocket: open-source cavity detection and characterization on molecular dynamics trajectories. *Bioinformatics*, **27**(23), 3276–3285.

Sethian, J. A. (1996). A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, **93**(4), 1591–1595.

Sussman, M., Smereka, P., and Osher, S. (1994). A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics*, **114**(1), 146 – 159.

Tonddast-Navaei, S., Srinivasan, B., and Skolnick, J. (2017). On the importance of composite protein multiple ligand interactions in protein pockets. *Journal of computational chemistry*, **38**(15), 1252–1259.

Venkatachalam, C. M., Jiang, X., Oldfield, T., and Waldman, M. (2003). Ligandfit: a novel method for the shape-directed rapid docking of ligands to protein active sites. *Journal of Molecular Graphics and Modelling*, **21**(4), 289–307.

Wang, B. and Wei, G.-W. (2016). Object-oriented persistent homology. *Journal of computational physics*, **305**, 276–299.

Wang, R., Fang, X., Lu, Y., and Wang, S. (2004). The pdbbind database: Collection of binding affinities for protein- ligand complexes with known three-dimensional structures. *Journal of medicinal chemistry*, **47**(12), 2977–2980.

Weisel, M., Proschak, E., and Schneider, G. (2007). Pocketpicker: analysis of ligand binding-sites with shape descriptors. *Chemistry Central Journal*, **1**(1), 7.

Xia, K., Feng, X., Tong, Y., and Wei, G. W. (2015). Persistent homology for the quantitative prediction of fullerene stability. *Journal of computational chemistry*, **36**(6), 408–422.

Xie, L. and Bourne, P. E. (2007). A robust and efficient algorithm for the shape description of protein structures and its application in predicting ligand binding sites. In *BMC bioinformatics*, volume 8, page S9. BioMed Central.

Yu, J., Zhou, Y., Tanaka, I., and Yao, M. (2009). Roll: a new algorithm for the detection of protein pockets and cavities with a rolling probe sphere. *Bioinformatics*, **26**(1), 46–52.

Zhang, Y., Tong, Y., and Zhou, K. (2017). Coloring 3d printed surfaces by thermoforming. *IEEE transactions on visualization and computer graphics*, **23**(8), 1924–1935.

Zhao, H. (2005). A fast sweeping method for eikonal equations. *Mathematics of computation*, **74**(250), 603–627.

Zomorodian, A. and Carlsson, G. (2005). Computing persistent homology. *Discrete & Computational Geometry*, **33**(2), 249–274.

Cover Letter

Dear reviewers,

We greatly appreciate your valuable comments and suggestions on our paper. We tried out best to modify the paper accordingly. We highlighted the changed text in red color, except for minor typos. In the following, we provide the details on how we addressed the suggestions.

1. **Review 1:** *The paper is well written and explained, although an additional effort should be made to convey the gist of the results to readers with a non-mathematical background.*
   Thank you for the suggestion. We will not be able to make it accessible to reader without sufficient mathematical background without going over the page limit. However, our presenter at the conference will make every effort to illustrate our results to those without mathematical background among the audience.

2. **Review 1:** *Although the authors present cases of proteins, where subpockets are detected, it would be interesting to see an example of real drugs/ligands which bind in several subpockets of the same pocket.*
   Our current paper focuses on the algorithmic aspect of detecting such hierarchies in protein pockets. Currently, we have not experimented with finding such real cases, but we will investigate the subpockets in a more realistic setting as future work.

3. **Review 2:** *In equation 15, the parameter "c" is not explicit or too late.*
   We added a sentence at Line 383 to explain the parameter "c". To avoid confusion with another parameter denoting the deforming speed, we also replace it by "d".

4. **Review 2:** *In the results of Table 1, the authors would have to justify why the success rate decreases, when we estimate pockets by greater proximity (with a c increasing), which seems counterintuitive.*
   We added a sentence at Line 404 to explain this behavior.

5. **Review 2:** *It would be necessary to detail more the choices of the settings and to discuss more some limits and perspective of the approach (flexibility of proteins).*
   We added some explanations to the choices of the settings, and added a sentence at Line 452 to mention our limitation with flexibility of proteins, and also to offer one viable solution for such cases.

6. **Review 2:** *Page 2, lines 73-74: "grid based method ... typically suffer from accuracy and efficiency issues", while a Catesian grid is retained, lines 93-94. So, explain why the current method does not suffer from these drawbacks.*
   We added a sentence at Line 74 to clarify why the previous grid based methods suffer from efficiency.

7. **Review 2:** *Page 3, lines 165-166: the parameter w has a crucial impact on the results. How to select its value? (should have been discussed in section 5).*
   We added a sentence at Line 169 to address this issue. In short, w is a parameter for the data structure. It only changes memory usage and computational cost, without influencing the results.

8. **Review 2:** *Page 4, Algorithm 1: pls. define the functions NotAllSurfaceBlocked() and ReinitializeSDFIfNeeded(), and/or explain to the readers what they do.*
   We added comments to these two functions pointing to the figure and the subsection, where the details are provided.

9. **Review 2:** *Page 4, section 4.1, just curious: how do you compute the convex hull of a vdW surface, an SAS or an SES? Since this hull should contain portions of spheres, a discretization procedure should be defined. The user is left with this decision (see line 359): without help, he is assumed to master the math background presented in the manuscript. I am not sure that most biologists can do that.*
   We clarified this issue by adding explanations at Lines 278 and 291. Basically, we triangulated SES first using a piece of software in the reference.

10. **Review 2:** *Page 5, section 4.6, lines 349-356: I can understand that counting voxels is a mean to estimate the volumes of the pockets. But what about surface area: how is it computed? Does the generation of*

a sequence of embedded cubic grids with decreasing edge lengths would show a convergence of the associated computed surface area to a suitable limit, meaningful for the biologist? How is the parameter "depth" computed?

We added sentences at Line 356 to further explain how to compute depth. And we added sentences at Line 365 to explain how we calculate area fast but roughly, and included a reference to a more accurate calculation with convergence proof, which uses a weighted sum of relevant voxels.

11. **Review 2:** *Page 6, Results and discussion: in this section, it is unclear whether or not all protein heavy atoms are considered, or only th C-alphas representing the residues.*

We added a sentence at Line 381 to explain that we use all protein atoms including hydrogen.

12. **Review 2:** *Page 6, eq. (16), even after having fixed the parameter r, it is unclear what means a "success rate" (something like a ratio predicted minus experimental to experimental), since the reader does not know how are computed the reference values (experimental or else).*

We added sentences at Line 386 to define exactly what is the "success rate" we use. It is the percentage of proteins that our method succeeded to detect the pockets.

13. **Review 2:** *Page 6, Table 1: the method is evaluated for several c and r values, but ω is not recalled.*

According to [Sethian 1996] and in our tests, w has no influence on final results as long as it is large enough (3 in our case). It only changes the memory consumption. So including that parameter would only produce identical numbers.

Thank you again for all the great suggestions.

Sincerely,
The authors.