Zuse Institute Berlin

Takustr. 7
14195 Berlin
Germany

M. KRONE, B. KOZLÍKOVÁ, N. LINDOW, M. BAADEN, D. BAUM, J. PARULEK, H.-C. HEGE, I. VIOLA

Visual Analysis of Biomolecular Cavities: State of the Art

Preprint version

ZIB Report 16-42 (August 2016)

Zuse Institute Berlin
Takustr. 7
D-14195 Berlin

Telefon: +49 30-84185-0
Telefax: +49 30-84185-125

e-mail: bibliothek@zib.de
URL: http://www.zib.de

ZIB-Report (Print) ISSN 1438-0064
ZIB-Report (Internet) ISSN 2192-7782

Visual Analysis of Biomolecular Cavities: State of the Art

M. Krone[1], B. Kozlíková[2], N. Lindow[3], M. Baaden[4], D. Baum[3], J. Parulek[5], H.-C. Hege[3], I. Viola[5,6]

[1]Visualization Research Center, University of Stuttgart, Germany

[2]Faculty of Informatics, Masaryk University, Czech Republic

[3]Department of Visual Data Analysis, Zuse Institute Berlin, Germany

[4]Laboratoire de Biochimie Théorique, UPR 9080 CNRS, France

[5]Department of Informatics, University of Bergen, Norway

[6]Institute of Computer Graphics and Algorithms, TU Wien, Austria

Abstract

In this report we review and structure the branch of molecular visualization that is concerned with the visual analysis of cavities in macromolecular protein structures. First the necessary background, the domain terminology, and the goals of analytical reasoning are introduced. Based on a comprehensive collection of relevant research works, we present a novel classification for cavity detection approaches and structure them into four distinct classes: grid-based, Voronoi-based, surface-based, and probe-based methods. The subclasses are then formed by their combinations. We match these approaches with corresponding visualization technologies starting with direct 3D visualization, followed with non-spatial visualization techniques that for example abstract the interactions between structures into a relational graph, straighten the cavity of interest to see its profile in one view, or aggregate the time sequence into a single contour plot. We also discuss the current state of methods for the visual analysis of cavities in dynamic data such as molecular dynamics simulations. Finally, we give an overview of the most common tools that are actively developed and used in the structural biology and biochemistry research. Our report is concluded by an outlook on future challenges in the field.

1 Introduction

The behavior of matter and the functioning of living systems is determined by molecular interactions. A molecule consists of atoms, each occupying a certain amount of space and contributing to the molecule's electron shell and force fields. For molecular interactions the spatial setting is of great importance, as most of the various forces are short-ranged and also are shielded by other parts of the molecule. In order to interact, molecules must come close together. The possibilities of molecules to move, to form bonds, and to arrange themselves to complexes are determined by *spatial* and *physicochemical* conditions. Molecular behavior thus is explained halfway by spatial aspects,

in particular by the shapes of the molecules.

Particularly important for understanding molecular processes are two related aspects of molecular shapes: First, the mutual *accessibility* of molecules, characterized by molecular surfaces, and, second, the *spatial volumes on the boundary or in the interior of a molecule, which can be occupied by other molecules or ions,* i.e., regions that are not filled with atoms of the molecule under consideration. Depending on the context, the spatial characteristics of these volumes, and the nomenclature used, these "empty" spatial volumes are called cavities, pockets, indentations, clefts, grooves, protrusions, voids, pores, channels or tunnels. In this article, we use the term *cavity* as generic term for all types of such spatial volumes.

While the generation, analysis, and visualization of *molecular surfaces* has been reviewed in our recent survey [66], the methods for detecting, analyzing, and visualizing *cavities* is the subject of the present article. Brezovský et al. report in [7] on available tools for cavity analysis and their functionality, without detailing the methodological background of the algorithms.

The importance of the cavities for the understanding of molecular phenomena can be seen from the following examples:

- A binding site is a region on a molecule to which other molecules and/or ions may bind, or even form a chemical bond. Binding sites exhibit chemical and spatial complementarity, often in form of a pocket or cleft. To understand the reactivity of a protein and to elucidate its function, binding site analysis, i.e., characterization of spatial and physico-chemical characteristics has to be performed.

- A particular example for a binding site is the active site of an enzyme, i.e., the region where substrate molecules bind to an enzyme and undergo a chemical reaction. Often this is the largest cavity on the surface of the enzyme. Characterization of known as well as detection of novel enzymes requires a detailed geometric and physico-chemical analysis of the cavity.

- In many biochemical processes molecular recognition plays a crucial role. The molecular specificity requires particularly pronounced geometrical and physicochemical complementarity, i.e., specifically shaped pockets and well-defined stereochemical arrangements.

- In pharmaceuticalmedicinal chemistry one aims at finding or constructing sites that bind drug-like molecules. Cavity analysis therefore is an essential part of the analysis of pharmaceutical agents, of rational drug design and of "druggability" prediction.

- Major biological processes are transport processes where molecules or ions are transported through protein complexes that belong to a biological membrane. Revealing potential paths for the molecules to be transported requires a spatiotemporal analysis of cavities in such protein complexes. As many transport processes involve long time scales that often are out of reach of molecular dynamics (MD) simulations, supporting geometrical cavity analyses are particularly useful.

Since the presence and the shape of cavities depend on the dynamics of the molecule or molecular complexes, cavity analysis often requires tracing on basis of MD trajectories; the development of such algorithms has started recently. When complex biochem-

ical phenomena are considered that are not (yet) accessible to full MD simulations one has to resort to geometrical analysis and simplified physical considerations.

Independent of whether full, partial, or only rudimentary MD analysis of the process of interest is possible, a geometrical and visually supported interactive analysis of cavities is necessary. This analysis can for example reveal if an active site within a cavity is accessible to the specific substrate molecule under certain boundary conditions. This procedure explains the fundamental need for fast algorithms for detection, analysis, and visualization of cavities. Because of the practical importance of visually supported cavity analysis, a large number of methodological approaches has emerged. To the best of our knowledge, the approaches altogether have never been compared and classified. With the present report we provide a formal definition of cavities, a methodological overview, a grouping of the different analysis methods according to methodological criteria, and finally an overview on available practical software tools. Such a survey seems to be necessary for further successful development of the analysis techniques—particularly since different research communities from separated disciplines are involved, who often are not fully aware of the progress in neighbored disciplines.

The report is structured as follows. In Section 2, a formal definition and a classification of various types of cavities is given. Section 3 provides additional aspects from the application side and mentions related areas. In Section 4, the various terms used for cavities, sometimes with slightly different meaning, are further clarified and the various analysis techniques are classified methodologically. In Section 5, algorithms for the extraction of cavities are presented, while Section 6 deals with their interactive visual analysis. In Section 7, the plethora of different methods is discussed, a brief overview on available tools is given, and directions on the comparison and verification of cavity extraction methods are provided. The final section provides conclusions and an outlook.

2 Definition & Classification of Cavities

Although there are many algorithms to compute molecular cavities, there is no clear formal definition for these structures. Often they are defined implicitly by the developed algorithms. In this section, we try to give a formal definition of molecular cavities based on the definition of paths. All algorithms to compute molecular paths and cavities that are presented in Section 5 are simplifications or restrictions of this formal definition. In general, the better these simplifications approximate the formal definition, the higher the accuracy of the cavities is.

Generally, a molecular path is a path of a small molecule or ion within a larger molecule. This could be, for example, a path of a substrate to its binding site in a receptor protein or the path of an ion through a channel of a membrane protein. Note that both molecules are dynamic structures, which makes a path time-dependent. Furthermore, a molecular cavity is defined as a continuous volumetric void space that can be accessed by the small molecule. Thus, each cavity is described by the space around connected molecular paths. Additionally, cavities require the definition of a volumetric boundary based on the large molecule. This boundary separates inside and outside. Without this boundary, all channels and pockets would belong to the same cavity because they are connected by paths outside the large molecule. In contrast to this formal, theoretical description of paths and cavities, it is quite difficult to define this boundary in practice, because it is not independent of the application.

2.1 Formal Definition

Let X be the current state of a molecule. It includes all properties to describe the molecule based on the underlying physical model. For example, for the classical physical model, the state includes the atom positions and electrostatic potentials as well as the bonding and non-bonding forces. If the molecule changes over time to another state Y, for the following definitions, it will be assumed that a continuous function exists that connects these two states.

Consider two molecules, a larger one, which could be a receptor protein, and a smaller one, which could be a substrate, ligand, solvent, or ion. First, we observe a static state $\hat{X}$ of the large molecule. Let $S_{\hat{X}}$ be the set of all states the smaller molecule can adopt under the influence of the large molecule in state $\hat{X}$. A *molecular path* of the small molecule is then defined as a continuous curve c in the space of $S_{\hat{X}}$. Furthermore, let $b_{\hat{X}}$ be a boundary function that evaluates if a state of the small molecule lies inside or outside of the region of the large molecule or if it lies on the boundary. The restriction $\tilde{S}_{\hat{X}} \subset S_{\hat{X}}$ is the set of all states $X \in S_{\hat{X}}$ for which $b_{\hat{X}}(X)$ evaluates the state as being inside or on the boundary. In the non-degenerate case, $\tilde{S}_{\hat{X}}$ consists of a network of paths with one or more connected components.

Now consider the spatial region $V_{\hat{X},X}$ in $\mathbb{R}^3$ representing the volume of the small molecule in state X under the influence of the large molecule in state $\hat{X}$. Note that a unique formal definition of $V_{\hat{X},X}$ is not available, again it depends on the underlying physical model. However, reasonable heuristics to approximate $V_{\hat{X},X}$ exist. For a state $\hat{X}$ of the large molecule, a *molecular cavity* is defined as the union of all volume sets whose corresponding states are connected by molecular paths in $\tilde{S}_{\hat{X}}$. Note that even if two cavities intersect each other, there still might not exist a molecular path between any two states of the two cavities.

Consider now the case where the large molecule is dynamic, i.e., $\hat{X}$ is a function of time $\hat{X}(t)$. Let $Y \in S_{\hat{X}(t_1)}$ and $Z \in S_{\hat{X}(t_2)}$ be two states of the small molecule for different times. A *dynamic molecular path* between Y and Z is defined as a time-dependent continuous function c, with

$$c : [t_1, t_2] \subset \mathbb{R} \to S_{\hat{X}(t)} \text{ , with } c(t_1) = Y, \; c(t_2) = Z.$$

Furthermore, a *dynamic molecular cavity* is defined as the union of all $V_{\hat{X}(t),X}$ that are connected either by a dynamic molecular path or by a molecular path. Thus, four possible topological events can be distinguished for the change of a molecular cavity over time. It can appear and disappear, or it can merge into another cavity or split into two or more cavities.

2.2 Simplification

Since the computation of all molecular paths is similar to an infinite number of physical simulations for all possible states of the small molecule, it is not practicable to directly apply this definition to the analysis of a potential receptor molecule (neither to a static receptor nor to the results of molecular simulations). To create a practical solution, the states of the molecules need to be simplified. For this reason, the formal definition serves as theoretical ground truth and allows to study the degree of simplification. An example of a typical simplification is given in the following.

Often, the states of the molecules are restricted to a space with pure geometrical properties. For the states of the large molecule, typically the hard sphere model is

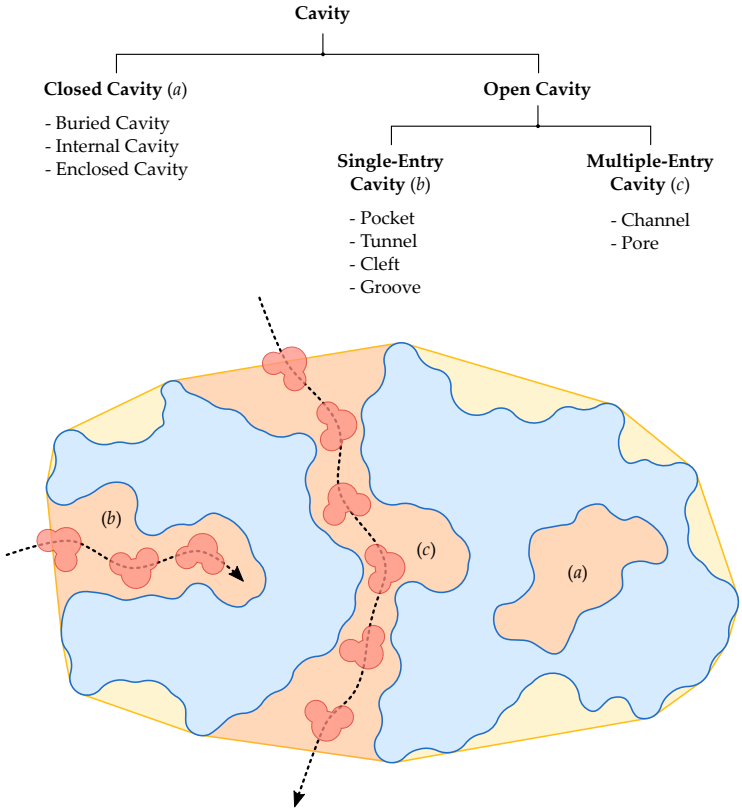

Figure 1: Definition and classification of molecular cavities. The boundary of the large molecule is shown by the yellow region and two molecular paths are depicted.

applied, which means only the atom positions and radii are used to create an imaginary hard boundary of the molecule. In addition, the small molecule is often approximated by a single probe sphere. Thus, for a state $\hat{X}$ of the large molecule, the set $S_{\hat{X}}$ includes all probe centers, where the probe does not intersect any atom sphere of the large molecule. With this restriction, a molecular path is a three-dimensional continuous curve of probe centers and a cavity is the union of all points inside all probe spheres that are connected by continuous curves in $\tilde{S}_{\hat{X}}$. Furthermore, a dynamic molecular path is a three-dimensional continuous curve of the probe over time and a dynamic cavity is the union of all cavities that are connected by dynamic molecular paths. For most grid-based algorithms, additionally, the shape of the molecules as well as their dynamics are discretized in $\mathbb{R}^3$ (see Section 5.1). As boundary $b_{\hat{X}}$ for the large molecule, the convex hull of the atom positions or atom spheres is often used. Other approaches use a distance threshold to the atom spheres or simply the axis-aligned bounding box of the atom spheres. An ambient occlusion threshold as boundary indicator seems to be quite promising and is also used in several approaches.

Figure 1 shows a 2D illustration of a simplification. The states of both molecules are restricted to the atom positions and radii and the boundary $b_{\hat{X}}$ is the convex hull of the large molecule.

2.3 Classification

Besides the term cavity, many other denotations and classifications are used in the literature, for example, voids, hollows, pockets, tunnels, pores, or channels. In the following, we propose a classification based on the cavity definition above that clarifies the meaning of these terms in this work. While void can be seen as a synonym for cavity, all other terms usually describe a specification of a class of cavities. To do so, we distinguish between closed and open cavities. All molecular paths inside the closed cavities do not reach the boundary, given by $b_{\hat{X}}$. Open cavities are further separated into cavities with a single entry or with multiple entries. An entry is a set of states of the small molecule where all states are connected by paths that lie completely in the boundary. We define cavities with a single entry as pockets, tunnels, grooves, or clefts and cavities with multiple entries as channels or pores. An illustration of this classification is given in Figure 1. Note that subtle differences between these subcategories exist. For example, grooves are usually shallower than pockets, which are in turn not as deep as a tunnel. The term pores is typically used for straight channels. Since we are, however, mostly interested in the geometric properties of cavities, the simple categorization into single-entry cavities and multiple-entry cavities is sufficient.

3 Differentiation from Related Areas

This section introduces related areas of research that are not discussed in detail in this report. The purpose is to position the work discussed in our report within the greater subject of molecular visualization and analysis, and to provide directions for further reading.

A large field that cannot be covered in this report is the usage of MD simulations to investigate phenomena like binding affinities in cavities or transport processes. In this survey, we focus on fast geometry-based techniques to extract, analyze, and visualize potential paths and cavities (cf. Section 2.2). This can be seen either as pre-processing for simulations to detect, for example, potential binding sites in cavities, or as post-processing to analyze the dynamics of cavities. A typical application of MD simulation is the evaluation of the binding affinity of a certain ligand in a known cavity. These methods are out of scope of this survey since they are not designed to detect cavities in a molecule or to extract their geometrical properties. Readers interested in this field are referred to the following docking reviews [24, 96] and some recent works [33, 123, 135].

Void spaces or cavities are important for molecular interactions like enzymatic reactions, which are typically triggered by a ligand that docks to the active site of a protein. The traditional lock-key model of enzymatic catalysis proposed by Fischer [36] was designed for proteins with exposed active sites located on their surface. It follows the notion that the ligand fits the respective area on the protein geometrically like a key into a lock. For active sites that are buried inside the protein core, Prokop et al. [110] proposed the keyhole-lock-key model, where the keyhole is a path that leads to the active site. These models fit to most of the methods and algorithms discussed in this survey. However, it is noteworthy that this (keyhole-)lock-key metaphor does not apply to all proteins. For example, intrinsically disordered proteins [129] do not exhibit a stable conformation prior to the docking of a ligand. Their analysis requires specialized visualization methods like the one proposed by Heinrich et al. [45], which are out of scope of this survey.

A different approach to investigate molecular interactions are interactive systems where the user can manually dock the ligand to the protein surface. Typically, both molecules are treated as rigid bodies. The forces between them are computed interactively, which enables direct feedback, e.g., using a haptic input device. A recent example of such a system is *Haptimol_RD* by Iakovou et al. [53]. Maciejewski et al. [92] presented a molecular docking application based on volumes, which uses a haptic transfer function that makes soft and permable objects possible. Such interactive analysis tools are, however, diametral to the approaches that this survey focuses on, since they do not analyze the protein but rather offer users a way to explore possible paths for a ligand.

Void space analysis is also useful in protein-protein docking, which aims at predicting the preferred mutual orientation of two or more molecules binding together and creating a stable complex. However, this requires different search strategies compared to those for protein-ligand docking. We do not cover this topic in our survey, since a comprehensive review and evaluation of current methods was recently presented by Huang [51].

Contrary to the number of algorithms that analyze protein-protein interactions, there are only few specialized methods for their visualization. Existing approaches such as the one of Jin et al. [55] often combine simple 3D representation with 2D interaction maps for visual analysis. Another related area of ongoing research are protein interaction networks, which can be represented by network diagrams called *protein interaction maps*. For example, Edes et al. [32] proposed a tool for visualizing these maps using Kohn's Molecular Interaction Maps [64]. Another example is the Cytoscape platform for visualizing complex networks by Shannon et al. [118]. Since these visualizations do not focus on void spaces between the proteins but rather on chemical interactions, they are out of the focus of our survey.

In many biological processes, protein-RNA and protein-DNA interactions play a fundamental role. Although computational docking methods focusing on these protein-nucleic acid interactions are less frequently found in the literature than those solving protein-protein interactions, there are solutions like the *NPDock* web server [128]. Since these methods also do not explicitly deal with void spaces, they are not discussed in our survey.

To the best of our knowledge, only a few approaches explicitly extract intermolecular voids between molecules. For example, Intersurf [112] extracts an interface surface and creates a corresponding interaction map between proteins that can be colored by attributes such as distance to protein, in some way similar to the earlier MolSurfer approach [37]. In a similar spirit, the approach by Lee and Varshney [81] computes a plane between two docked molecules. Then, a double-height field is generated that shows the distances from the plane to the molecules. The surface of this double-height field is colored to show negative volumes, that is, intersecting parts of the molecules. The visualization of the resulting intermolecular voids can help to assess the fitness of the proposed docking. In contrast, Maeda et al. [93] use the Delaunay complex as a basis to measure the volume of intermolecular voids. In the first step, all atoms in the region of the interface are detected. Then, the Delaunay complex of all interface atoms is computed. All tetrahedra between the two molecules contribute to the interface volume. Finally, the volume is discretized on a grid to remove the volumes of the atoms from the intersecting tetrahedra.

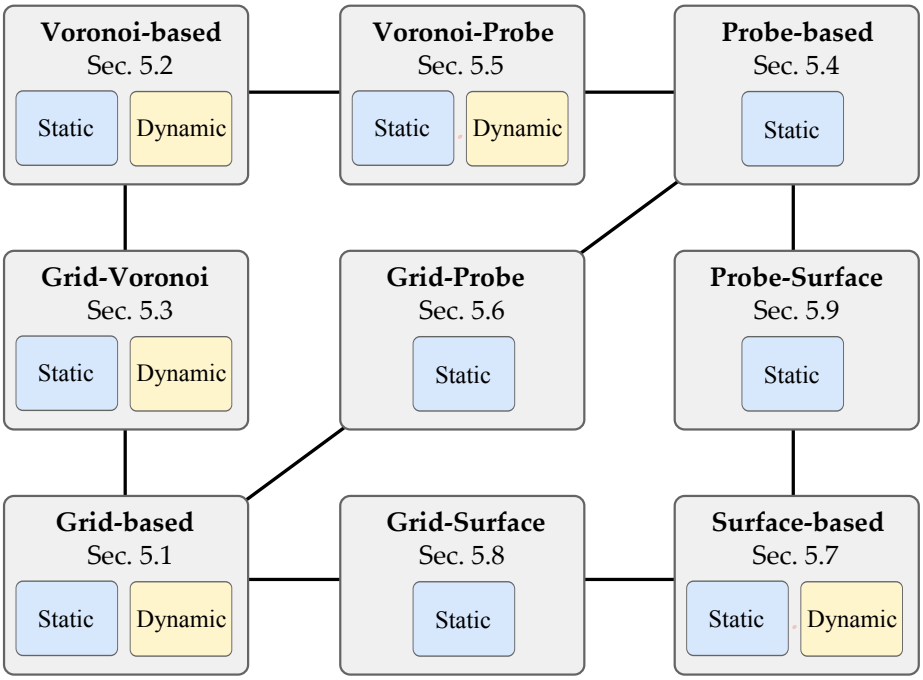

Figure 2: Classification of methods according to their algorithmic background. Grey rectangles represent individual categories, each of them contains methods for cavity computation on static molecules (blue rectangle), some are also able to handle dynamic molecular data (marked with ocher rectangle).

4 Classification of Methods to Analyze Cavities

As described in Section 2, different types of cavities can be distinguished based on their topological properties. Although these differences might be meaningful from the perspective of biochemistry and structural biology, from a technical point of view, they are often not very meaningful. In addition, the differences between sub-classes such as grooves and pockets are formally difficult to define. Almost all algorithms for the extraction of cavities simplify the molecule by the hard sphere model (Section 2.2). Hence, the extraction of cavities can be described in most cases as a geometry processing problem. Therefore, it is not only highly related to visualization of these structures but even the numerical results can highly depend on the employed computational method. For this reason, we propose a classification of these approaches according to the computational methods used for the cavity detection. All methods presented in this survey can be categorized into four main categories or a combination of two of these, as illustrated in Figure 2.

The four main categories are formed by methods based on Voronoi diagrams, grids, molecular surfaces, and (usually spherical) probes. These four categories form the corner nodes of the classification in Figure 2. The other five categories are combinations of these basic methods, which are shown as nodes between the four corner nodes. Furthermore, the categories can contain methods operating not only with static molecular structures but also with dynamic data, for example, the trajectories resulting from a molecular dynamics (MD) simulation. Our survey is structured according to this categorization. Therefore, the figure also contains references to the respective sections

describing the individual category and their corresponding methods.

An indispensable part of the cavity analysis is the visual representation of results. The output cavities detected by the algorithms from all categories can be visualized using different approaches which are in detail discussed in Section 6.

5 Algorithms for the Extraction of Cavities

Before discussing individual algorithms in detail, we briefly review the historical development. The first algorithms to detect cavities in molecules were based on grids due to simplicity. At that time, these solutions were suitable for small- to medium-size molecules (i.e., hundreds to a few thousands of atoms). Due to hardware limitations, they were not applicable to larger structures without decreasing the grid resolution substantially. One solution for this problem was the usage of Voronoi diagrams. Voronoi diagrams proved to be suitable for the detection of paths in molecules and were able to process large molecules as well. A limitation was their more complex implementation compared to grid-based algorithms. Other possibilities that were developed at about the same time were surface-based methods and probe-based ones. Molecular surfaces provide a natural way to detect cavities since they are defined as a border between a molecule and its environment. Probe-based methods have the advantage that the probe size approximates a ligand that can reach the extracted cavities. Nowadays, grid-based approaches are also popular again. This is due to the fact that current hardware allows to use fine grids even for very large structures and, thus, process even molecular dynamics simulations in a reasonable time. In combination with other approaches, the current solutions are very powerful and open new possibilities for the future development discussed in Section 8.

In the following, we will detail various methods for the extraction of cavities from molecular data. We will adhere to our cavity classification introduced in Section 2 for the naming of these intramolecular voids. That is, the terms used in this survey can sometimes differ from the ones used in the original papers for the sake of consistency and comprehensibility. The rest of this section is organized according to the algorithm-based classification of cavity extraction methods given in Section 4 (see Figure 2). Note that not all methods are able to extract all types of cavities. Therefore, Figure 3 shows an overview of all methods discussed in this section with respect to types of cavities the method can extract. The additional icons indicate important features and the algorithmic properties of the individual methods. This not only applies to the cavity extraction itself. Many methods and tools also introduce novel visualizations for cavities or allow for a comprehensive visual analysis of their properties. Thus, methods and tools that offer such capabilities are also highlighted by a dedicated *visual analysis* icon. The respective visualizations are discussed in detail in Section 6.

As mentioned in Section 3, the right branch of the tree in Figure 3—intermolecular voids—are out of scope of this survey and will not be discussed. We will focus only on methods for the extraction of intramolecular voids, that is, cavities as defined in Section 2. In accordance with most applications, we sometimes use the terms protein and ligand to denote the large and small molecule. However, most techniques are not restricted to the specific structure of a protein.

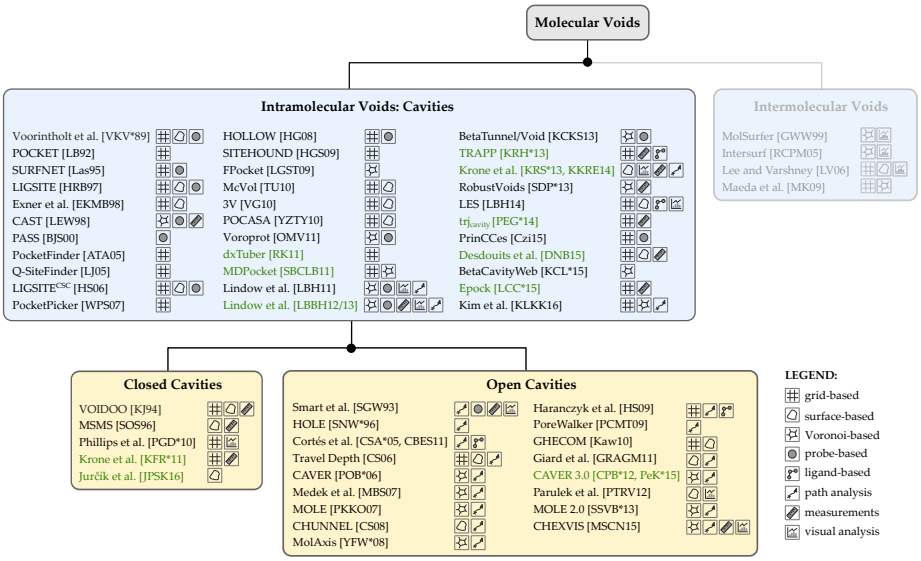

Figure 3: Classification of methods based on the cavity definition. Green color denotes tools and algorithms that are dealing not only with static molecules but are also able to process dynamic data like molecular dynamics trajectories. Note that Epock does not introduce a new method for cavity extraction but integrates other tools to enable the processing of molecular simulations (for details, see Section 6.)

5.1 Grid-based Methods

Many methods extract the cavities by simplifying the protein as hard sphere model and the possible ligand positions as discrete points, usually using a uniform cubical grid (Section 2.2). An advantage of such grid-based methods is that they usually require only simple data structures without numerical problems. For all following grid-based methods, the geometrical accuracy as well as the computation time and memory requirements depend greatly on the resolution of this grid.

One of the first approaches to compute and visualize cavities was POCKET [82] developed by Levitt and Banaszak in 1992. The algorithm creates a three-dimensional cubical grid with a user-defined cell width, which is typically 1 Å. For each grid point, the distance to the closest atom center is computed. If this distance is smaller than a predefined threshold (usually 3 Å), the grid point is marked as a protein contact point. Then, the neighboring grid points in the three main directions of each unmarked grid point are investigated. If such a point is bounded by protein contact points along both sides of at least one direction, the density of the point is set to 1. Note that the density is initialized with 0. Finally a modified Marching Cubes [91] algorithm is used to extract the surface of the cavities. Because of the small number of directions that are investigated for each grid point, the result depends a lot on the orientation of the molecule.

In contrast to this purely geometric technique, An et al. proposed a more physically-based technique in 2005, which is implemented in the tool PocketFinder [1]. They also use a grid for the cavity detection, but instead of geometrical properties, they compute the Lennard-Jones potential of a carbon probe atom at each grid position. In the next steps, they smooth the discrete potential field and compute a threshold using the average field value and the root mean square distance of all values. With this

threshold, the cavities are given by the isosurface of the discrete potential field. Finally, cavities with a volume smaller than 100 $Å^3$ are filtered out.

Q-SiteFinder by Laurie and Jackson [78] also does not rely purely on the geometry of the molecule. At each grid position, the non-bonded interaction energy is computed using the program Liggrid [54]. Using a threshold, all grid points with a high binding energy are marked and clustered. Finally volume calculations are performed. The authors compared their results with the ones of the Lennard-Jones-based PocketFinder [1] and stated that the Q-SiteFinder results have a higher success rate. SITEHOUND by Hernandez et al. [47] also uses non-bonded interaction energies to find potential ligand binding sites. Similar to Q-SiteFinder [78], it computes the binding affinity of either a carbon or a phosphate atom at each position in the grid. Subsequently, only grid points with a high binding affinity are considered and clustered to get potential cavities.

In 2006 appeared a grid-based solution for the detection of tunnels. The tool CAVER by Petřek et al. [108] searches for paths leading from a starting point located in the protein interior to its surface. Similar to POCKET [82], the cells of a uniform grid are clustered into two classes: those within an atom sphere (defined by the van der Waals radius of the corresponding element) and those containing an empty space. The convex hull is used to distinguish between the inner and outer space of the protein. The nodes on the boundary of this convex hull are potential stops of the grid-based path search algorithm, which aims at identifying the shortest low-cost path. The cost function is based on the length and curvature of the detected path: long and complicated paths are more ”expensive” than short and direct ones.

To overcome the limitations that are induced by investigating only few directions in a grid, Weisel et al. developed PocketPicker [132]. For each grid point that does not lie inside an atom sphere and whose minimal distance to any atom sphere is smaller than a user-defined threshold, a uniformly distributed set of 30 rays is cast. The rays are computed by subdividing an octahedron. For each ray, the surrounding atom positions are orthogonally projected onto the ray. If the distance between an original atom position and its projected position is smaller than 0.9 Å and the distance between the grid point and the projected position is smaller than 10 Å, the ray is marked as buried. For 16-26 buried rays, a grid point is defined as point inside a pocket. Grid points inside pockets are again clustered. Additionally, the shape of a pocket is described by evaluating the buried values and distances of all pairs of grid points in a single 420-dimensional vector. Figure 4 shows an example of a pocket detected by the PocketPicker.

Phillips et al. [109] proposed to cast parallel rays through the molecular structure in order to determine exact intersection points. From this information, one can analytically solve the line integral for each ray. All rays are then summed up to obtain an approximation of the molecular volume. The quality of this approximation can be controlled by the resolution of the plane from which the rays are cast. Naturally, all intersection points along the ray have to be found, since there can be internal cavities. That is, the ray-casting can be used for the extraction of cavities that are located behind the first surface intersection point. A flood-fill segmentation is used to distinguish between the empty space surrounding the molecule and the empty space within (i.e., the internal cavities). Furthermore, the surface area can be approximated using the information about the intersection locations by considering the area near each intersection point as a small quadrangle of the size of the pixel from which the ray was cast.

A method that takes the actual geometry of the substrate into account was proposed by Haranczyk and Sethian [44]. They use a 7-dimensional space that includes translational, rotational, and internal degrees of freedom of the substrate. For each sample in this space, it is checked if the substrate is in a valid state with respect to the static

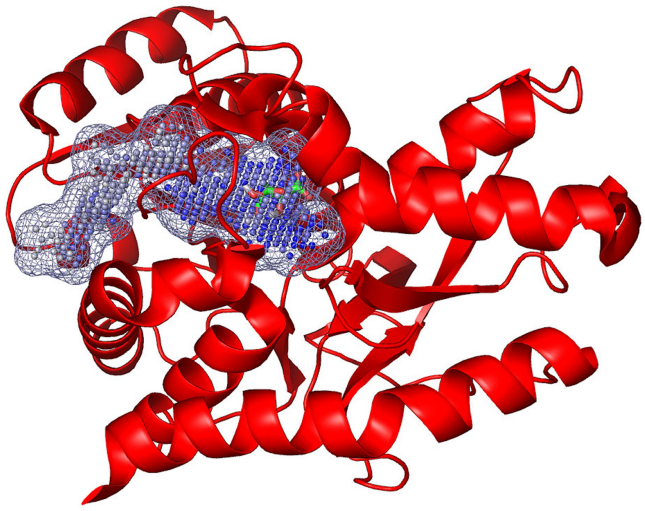

Figure 4: The largest pocket (blue spheres) of malate dehydrogenase (PDB ID 2CMD) detected by PocketPicker. Image source: [132].

receptor molecule. After sampling the space, the shortest path for the substrate is computed based on the valid samples. Due to the sampling of the 7-dimensional space, the approach is time-consuming and requires a lot of memory.

The abovementioned methods can only process static molecular structures. For the analysis of dynamics data, such as the results of a molecular dynamics simulation, these methods can only process the individual snapshots of the simulation but will not correlate the results between frames or compute temporal statistics. The tools and methods that are discussed in the following take this aspect into account.

In 2010, Raunest and Kandt [111] presented dxTuber, one of the first tools that investigates internal cavities based on the dynamics of the protein and water inside and around the protein. To achieve this, the protein dynamics are simulated inside a lipid membrane (in case of a membrane protein) and surrounded by water using the molecular dynamics (MD) simulation package GROMACS [2]. The positions of the water molecules yield the cavities of the protein. The authors found that short simulations of only 100 ps are sufficient to detect all cavities reachable by water. In order to compute the shape of the cavities, two 3D grids are used that store the number of water atoms and protein ones per grid cell. The cavities are detected and characterized by investigating cells along the three main directions for each grid cell. A grid cell is characterized as internal cavity if it is surrounded along all three axes in positive and negative direction by the protein. If only two of the three directions are surrounded by the protein, the grid cell is characterized as tunnel and in case of only one direction, the cell is defined to be inside a pocket. Finally, the grid cells will be clustered and the result is filtered to get the description of the cavities. Figure 5 clearly illustrates the workflow of the algorithm.

While the algorithm is suitable to detect cavities accessible by water, it cannot detect empty cavities. These cavities are often more flexible and their dynamics are often related to conformational changes in the protein. Furthermore, the algorithm results in a static representation of the cavities, which does not allow to study cavity dynamics. Hence, transport processes due to cavity changes that build, for example, a dynamic channel that is built by single-entry and closed cavities over time cannot be

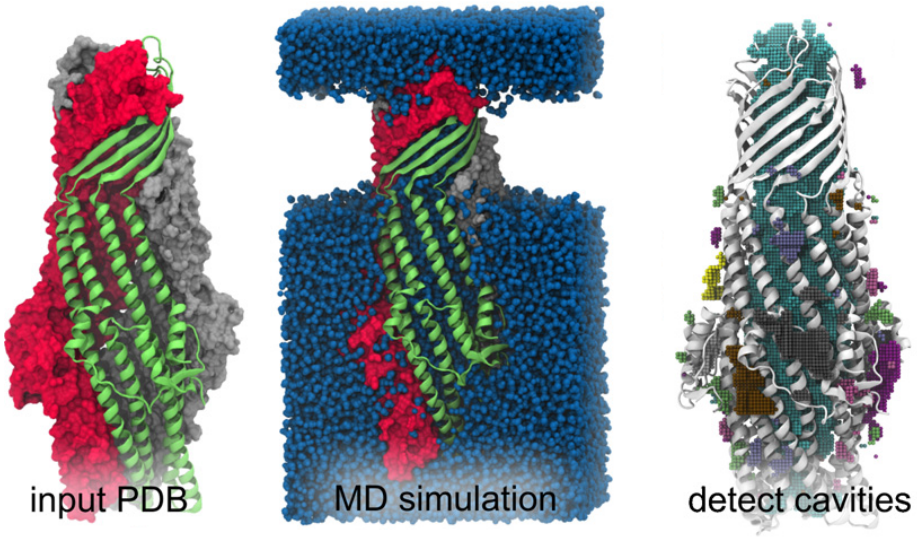

Figure 5: Cavity detection workflow proposed in dxTuber. Image source: [111].

investigated.

In contrast to the method by Raunest and Kandt [111], which uses a MD simulation to find channels and aggregates their dynamic behavior in one static snapshot, Krone et al. [69] developed a method to extract enclosed cavities directly from a molecular simulation trajectory. For each simulation frame, a Gaussian density grid is computed in real time from the atom positions using a GPU-accelerated algorithm. A semi-transparent molecular surface can be extracted from the grid using direct volume rendering. The user can interactively select an enclosed cavity by clicking on it. This cavity is extracted from the density grid via a 3D flood-fill segmentation. For subsequent time steps, the cavity found in the previous one is used as a seed point, which makes it possible to track the selected cavity over time. Additionally, the approximate cavity volume is computed by summing up the volumes of the grid cells (Figure 15 (a)).

TRAPP by Kokh et al. [65] is also tailored to trajectories or ensembles of structures and factors conformational changes into the cavity detection. Their grid-based algorithm determines the shape and physico-chemical characteristics of the voids. Two algorithms are implemented to take into account the motion: a Principal Component Analysis-(PCA-)based technique and an approach based on the average deviation from a reference structure. The method also offers the options to measure the fraction in which a particular cavity is open, to compare similarity of cavities between different structures, to trace the contribution of amino acids to a site of interest, and to measure spatial complementarity between void and ligand. More recently, PCA was used to follow cavity evolution throughout MD simulations and correlate it to functional motions in proteins [22].

Paramo et al. [100] presented a tool for MD trajectory analysis called trj_{cavity}, which detects and characterizes cavities. It analyzes the temporal evolution of cavity topology and provides different measurements (volume, occupancy, solvent or ligand statistics, cross-section, bottleneck identification). Similar to the work of Krone et al., time efficiency was a design focus. Thus, the method also uses an efficient grid-based region-growing algorithm that detects the type of cavity in terms of how many surrounding grid cells belong to the cavity or the protein. Trajectories of the cavities that show their

temporal evolution can be generated for visualization.

Laurent et al. [77] developed the tool Epock, which measures properties of cavities within a predefined maximum encompassing (MER) region of the investigated molecular system. The algorithm is based on POVME [27, 26] with some improvements on accuracy. Epock was designed in the context of extensive MD simulations, where relevant cavities have been identified previously but need to be characterized efficiently over time to manage large, dynamic data sets. A particular issue that can be addressed by Epock is to separate close-by or interconnected networks of cavities (e.g., close-by ligand binding pockets) for analysis and comparison.

5.2 Voronoi-based Methods

Another group of algorithms to extract cavities is based on Voronoi diagrams. Among other advantages, this approach overcomes the basic limitation of the grid-based algorithms—the dependency of the accuracy and memory requirements on the resolution of the grid. The protein is often simplified by the atom positions or the hard sphere model. In contrast to grid based methods, the ligand positions are not restricted to discrete points (Section 2.2). Furthermore, the edges of Voronoi diagrams automatically provide geometrically optimal molecular paths based on the restriction.

There are two approaches that use Voronoi diagrams for detection of channels, tunnels, and pores, which were presented concurrently in 2007. Petřek et al. [107] published their MOLE algorithm, which computes the Voronoi diagram for centers of protein atoms. Its edges are assigned positive values representing the relative cost of taking this edge along a path. The cost function is derived from that defined by Petřek et al. in their previous grid-based CAVER 1.0 approach [108] but also takes the edge length into account. Then, Dijkstra's graph search algorithm is used to find the "cheapest" path leading from the starting point outwards. The boundary of the structure and its environment are determined by a convex hull. Figure 6 illustrates the procedure. When searching for more paths, a large positive "penalty" is added to the Voronoi edges that are parts of already detected paths. Dijkstra's algorithm then avoids these edges due to their high cost. Finally, the algorithm performs the clustering of detected paths that do not differ significantly.

At the same time, Medek et al. [95] presented their approach to the detection of tunnels in proteins. Their method computes tunnels using a Voronoi diagram and a Delaunay triangulation. First, it computes the Voronoi diagram for a set of points representing atom centers. The edges are evaluated according to their distance to the nearest atom. The authors claim that the computation is more convenient when using the dual structure of the Voronoi diagram, the Delaunay triangulation. As in MOLE, Dijkstra's algorithm is used for searching the path from the starting point to the outside solvent. The computed tunnel can then be represented by its centerline composed of Voronoi edges but also by a set of neighboring tetrahedra. The authors propose three modifications of the Voronoi diagram in order to detect more tunnels by changing the weights of Voronoi edges.

One year later, Yaffe et al. [133] presented their approach to the detection of channels in macromolecules, called MolAxis. This solution provides a shift in accuracy compared to the previous approaches. Since the Voronoi diagram of the atom positions does not take into account the different atom radii, the paths computed by MOLE and Medek et al. are geometrically not optimal. To increase the accuracy of the paths in MolAxis, the atoms are approximated by sets of spheres with constant radii. While, for example, a hydrogen atom can be approximated by a single sphere, a carbon atom

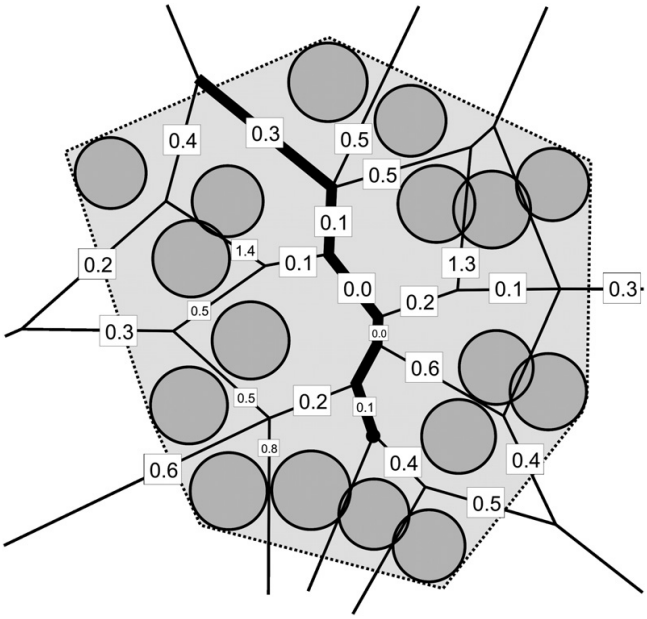

Figure 6: Voronoi diagram used by the MOLE algorithm [107]. Each Voronoi edge is evaluated by the cost function. The thick line represents the optimal path from a given point found by the Dijkstra's algorithm. Reprinted from Structure, 15(11), Petřek et al., *MOLE*, Pages 1357–1363, Copyright 2007, with permission from Elsevier.

is approximated by a cluster of several spheres of the same size. The Voronoi diagram is computed using the centers of these newly placed spheres. In the next step, all edges that correspond to spheres of the same cluster or that intersect the atom spheres are removed from the graph. The weighting and the path detection is then equal to the previous approaches. This algorithm accounts for the atom radii but substantially increases the number of spheres (i.e., their centers) that have to be inserted into the Voronoi diagram.

In 2013, Sehnal et al. [117] presented MOLE 2.0, an extension of the original MOLE algorithm by Petřek et al. [107]. The speed-up of this modified algorithm for computation of channels and pores on static molecules comes from several preprocessing steps. The implementation involves seven steps: computation of Voronoi diagram, construction of the molecular surface, identification of buried cavities, identification of possible channel start points (binding sites) as a subset of the buried cavities and similarly for end points, localization of channels, and filtering of the localized channels. Furthermore, the algorithm estimates physico-chemical properties of the identified channels, i.e., hydropathy, hydrophobicity, polarity, charge, and mutability. Most of the functionality of MOLE 2.0 is exposed in the MOLEonline 2.0 tool by Berka et al. [3]. MOLEonline 2.0 is a web-based tool using an embedded 3D graphical representation showing the detected path, its profile accompanied by a list of lining amino acids along with their basic physico-chemical properties.

All previously discussed methods were focusing on the analysis of static molecules. But similar to the grid-based methods discussed above, the Voronoi diagram approach can be used for processing of molecular dynamics trajectories as well. Thus, for the rest of this section, we will focus on algorithms that are able to deal also with dynamic

data.

In 2012, Chovancová et al. introduced the new version of the CAVER software first published in [108] which extends the solution presented by Medek et al. [95] based on Voronoi diagrams. The new extension, CAVER 3.0 [12], allows the user to analyze tunnels and channels in large ensembles of protein conformations. It detects individual paths in each time step and then clusters these paths in order to reveal their time evolution. The principles of the CAVER 3.0 algorithm were described in detail by Pavelka et al. [104]. In addition, the authors improved the clustering method for finding the correspondence between tunnels from different time steps of the molecular dynamics trajectory. They modified the average link hierarchical clustering approach used in CAVER 3.0. To avoid an expensive cluster-cluster similarity matrix, the distance between clusters is computed on the fly. For very large data sets, Pavelka et al. introduced two techniques to reduce the data size—subsampling and preclustering. These modifications enable processing larger sets of tunnels much faster. Both implementations are distributed as standalone command-line tools as well as PyMOL plugins. They were also integrated in the CAVER Analyst 1.0 tool [67].

Kingsley and Lill [62] focused on the combination of results by studying the variability of detected voids for different MD-based structural ensembles. Using the Cytochrome P450 enzyme family as example, CAVER 3.0 [12] and MolAxis [133] results on potential ligand paths were compared for a variety of structural ensembles derived from MD simulations. The ensembles comprised a collection of MD time steps, an RMSD-based clustering, a pairwise-distance clustering, and a hydrogen-bond network-based clustering. The main purpose was to provide guidelines on how the flexibility should be taken into account to be the most efficient, for example, how to generate a structural ensemble, how big it should be, and whether it should comprise apo- and/or holo-structural snapshots. The flexibility was revealed to be important to capture a maximum of paths and the authors proposed a general strategy to generate a representative ensemble of small size.

5.3 Grid-Voronoi Methods

Methods that combine Voronoi diagrams and grid-based extractions of cavities usually try to combine the benefits of both methods: the accuracy of the Voronoi diagram and the fast and simple handling of a grid.

Kim et al. [59] proposed a GPU-accelerated algorithm that extracts cavities using a grid-based Voronoi diagram. Their method first computes a voxelized approximate convex hull. Next, each convex hull voxel that is not within an atom is classified whether it belongs to a Voronoi diagram edge. This results in a discretized grid representation of the edges of the Voronoi diagram, which are then clustered and subsequently used to find paths.

Schmidtke et al. [115] presented the tool MDpocket, which uses fpocket [80] (see Section 5.5) to compute the Voronoi diagram of the atom positions for each time step of a MD trajectory. Then a grid is created on which a discrete density is computed based on the size of the α-sphere at the Voronoi vertices. By selecting only a section of the whole MD trajectory, the specific dynamic processes of the cavities can be analyzed. The cavities are visualized using an isosurface of the discrete density function. However, similar to dxTuber [111], it is difficult to analyze the detailed dynamic behavior of the cavities.

5.4 Probe-based Methods

In this section, we discuss methods to compute molecular paths and cavities that utilize the spatial extension of the ligand. For these methods and combinations with other techniques, the following simplifications are often applied to the ligand (Section 2.2). Most of the techniques approximate the ligand by a single hard sphere, called a *probe*. Few methods consider the full hard sphere model of the ligand, sometimes even with dynamics information. Furthermore, the positions of the ligand can be discretized. As for most of the previous techniques, the protein is often approximated by the atom positions or the hard sphere model. The probe-based methods presented in this section are only applicable to static data of the protein (e.g., individual snapshots of a simulation).

Smart et al. [121] presented a method that enables to characterize and display pores of ion channel proteins. This method was later included in the HOLE software [122]. The goal is to provide quantitative data to understand the biological ion permeation function of these channels by measuring properties relevant for ion conduction, pore dimensions, and constrictions. HOLE is one of the earliest tools to compute a possible molecular path. The tool computes a path from a user-defined start point inside a cavity to the outside of the molecule. The path direction is steered by a given direction vector $\vec{v}$ of the cavity. The start point is moved to the position where the distance to the atom spheres becomes locally maximal using a Monte Carlo simulated annealing approach. During this process, the point stays in the plane that includes the original start point and is orthogonal to $\vec{v}$. Afterwards, the point and the plane move a step into the direction of $\vec{v}$ and the simulated annealing approach starts again. This is repeated until the outside of the molecule is reached. Note that the approach cannot guarantee to detect the optimal point with the local maximal distance to the atom spheres. Furthermore, the algorithm fails to detect paths in cavities where the medial axis is more complex (i.e., it cannot be described by a single direction). Smart et al. [120] later presented an extension that predicts the conductance of an ion channel from its three-dimensional structure. The method combines the pore dimensions of the channel as measured in the HOLE program with an Ohmic model of conductance.

Laskowski presented the tool SURFNET [76], which fills cavities in a molecule with *gap spheres* that do not penetrate the atom spheres. In more detail, between each pair of atoms a gap sphere is placed in the middle, touching the two atom spheres. The radius of the gap sphere is reduced in case of a penetration with another atom sphere. If the radius falls below a user-defined threshold, the sphere is completely rejected. Finally all gap spheres are sampled into a three-dimensional grid using Gaussian density kernels. From this grid a surface of the cavities can be easily generated. The main shortcomings of this method are the time complexity, which is cubic and the geometric accuracy, which is not optimal due to the fixed position of the gap sphere.

One of the few methods that take the geometry and dynamics of a ligand into account was developed by Cortés et al. [18]. They use rapidly-exploring random trees (RRTs) [79] to compute a possible molecular path of a specific ligand to a binding site. That is, the method does not use a spherical probe but the actual geometry of the ligand molecule to probe the protein for possible paths. RRTs were originally developed for fast path planning in robotics. A tree is incrementally constructed by adding random valid robot configurations as tree nodes until a node reaches a point or area of interest (see Figure 7). For molecular path detection, the ligand is considered as the robot and the protein is the labyrinth for which a path should be detected from a user-defined start position to the outside of the protein. The start position is the root of the tree and a valid configuration is a position and orientation of the ligand such that it does not pen-

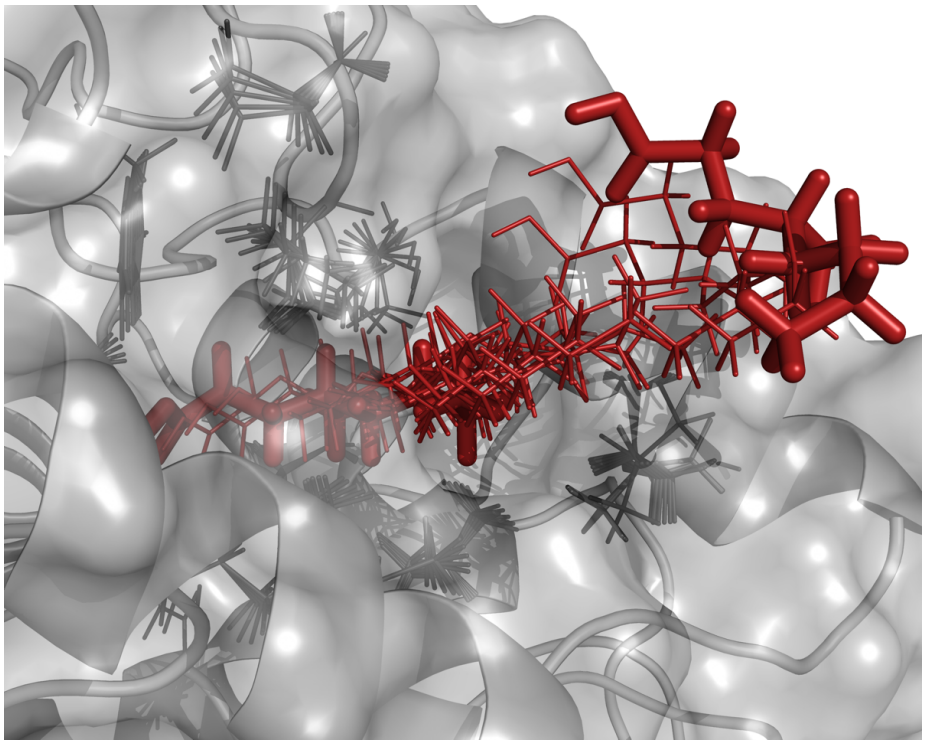

Figure 7: Trajectory of a ligand aiming to access the active site [18].

etrate the protein. Furthermore, the configuration has to be reachable from the closest node in the current tree. This means that the ligand must be close enough to an existing node in the tree such that it is guaranteed to move the ligand from the tree node to the new configuration without penetrating the protein. Depending on the number of free variables for the configuration of the ligand, the algorithm can be slow. Furthermore, it is difficult to setup a stop criterion for the tree construction. The authors further extended their approach to better visualize and analyze the results of the RRT [19]. To do so, they generate a 3D volume on which the RRT is mapped. The three variables of the volume can encode any user-selected ligand property of interest such as three selected bond torsions. The mapping algorithm is straightforward and simple visualization techniques for 3D volumes are applied. The implementation of the approach is part of the BioMove3D software package.

5.5 Voronoi-Probe Methods

This section covers approaches combining the Voronoi diagram method with the usage of a probe. Edelsbrunner and Liang presented a series of papers dealing with cavity detection and cavity analysis based on α-shapes and α-complexes. Ultimately, these works led to the development of a tool called CAST [85]. The algorithm calculates the Voronoi diagram consisting of Voronoi cells (Figure 8 left). The Voronoi diagram is mathematically eqivalent to the Delaunay triangulation of the complex hull drawn around the protein atom centers. The α-complex is then defined as a subset of the Delaunay complex (Figure 8 right). Each Delaunay element whose dual Voronoi element has a closer minimal distance to the atom positions than $\alpha \in \mathbb{R}$ is also an element of

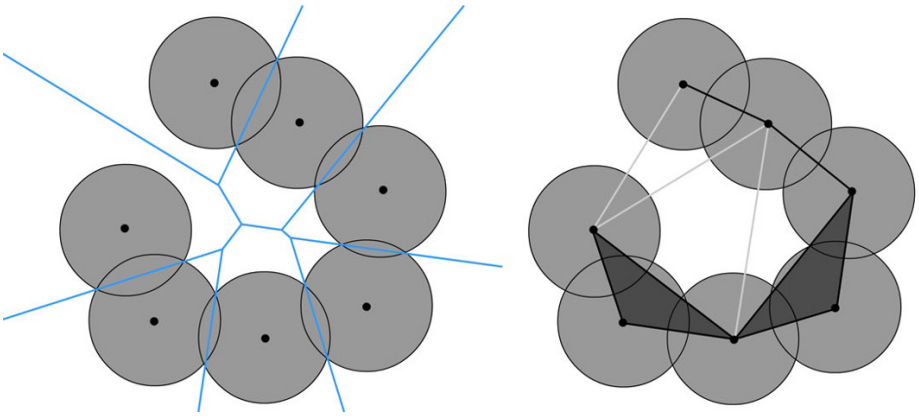

Figure 8: Illustration of the α-shape used in CAST. Left: Voronoi diagram of atoms of the same radii. Right: convex hull triangulated into Delaunay triangles, the dual complex is defined by the shaded triangles and the black lines. Image source: [132].

the α-complex. The probe radius is included in the α value.

In 1995, Edelsbrunner et al. [29] described the detection of internal cavities and the analytical computation of their volumes. These cavities can be easily extracted from the α-shape. In a subsequent work, they extended the cavity computation to pocket detection [30]. The approach uses the discrete flow of the Delaunay triangles to define and identify the cavities. Later, they presented the VOLBL tool to measure properties such as volume and area for internal cavities and pockets [83, 84]. As mentioned above, they integrated the detection and measurements into the tool CAST [85]. The minor drawbacks of the approach are the limited cavity visualizations and the problem that shallow pockets cannot be detected by the algorithm.

Similar to the CAST algorithm, fpocket by Le Guilloux et al. [80] first computes the Voronoi diagram of the atom positions and assigns a maximal α-sphere that does not intersect the atom spheres to each Voronoi vertex. In the next step, all α-spheres with a radius smaller than a minimal threshold or larger than a maximal threshold are removed. Afterwards, the remaining spheres are labeled as apolar or polar, depending on the neighboring atoms. A three-step clustering method is applied to the spheres. In the first step, α-spheres are clustered if they are connected by a Voronoi edge and if their distance is smaller than a given threshold. In the second step, clusters are aggregated based on the distance of their centers of mass. Finally, the pairwise distances between α-spheres of clusters are investigated. If a certain number of distances is smaller than the threshold, the two clusters are aggregated. After clustering, small and hydrophobic cavities are removed and the remaining cavities are ranked.

A combination of a Voronoi diagram and probes was also used by Olechnovič et al. [98] in 2011 when they presented Voroprot, which is an interactive tool for the analysis and visualization of cavities. Voroprot was one of the first tools using the additively weighted Voronoi diagram—also called Apollonius diagram—of the atom spheres instead of the atom positions (similar to Lindow et al. [88]). It computes the diagram to analyze interatomic contact surfaces but also to study cavities using the Voronoi vertices. For each vertex, an empty sphere that is tangent to four atom spheres exists. Such a sphere corresponds to an internal cavity if it is larger than a given probe sphere and if it is not accessible by the probe sphere from outside the molecule. Large probe sizes can be used to detect pockets. However, the authors do neither give a clear

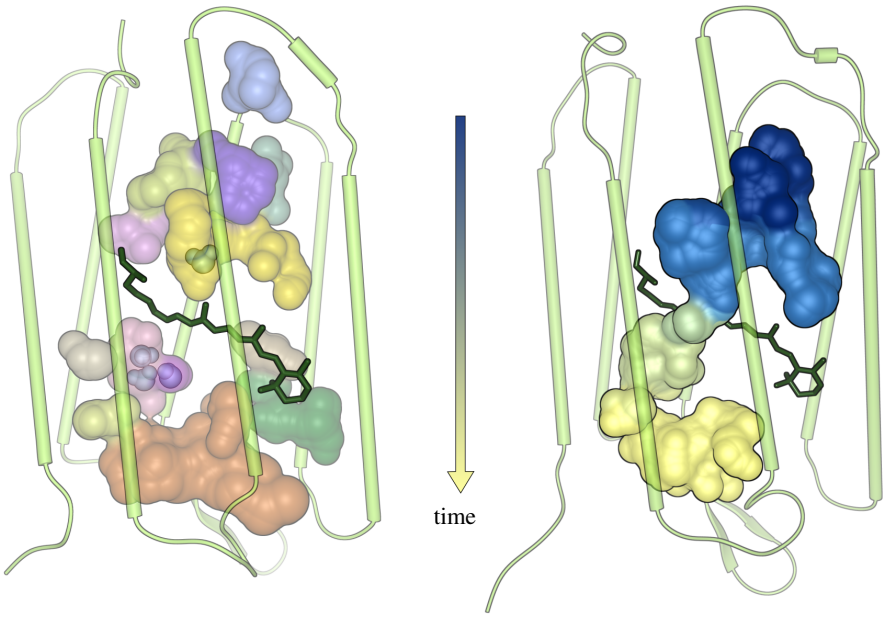

Figure 9: Cavities computed by the algorithm by Lindow et al. Left: cavities in one time step of the MD trajectory. Right: 3D shape of the dynamic cavity traced from the selected cavity (purple, left) and colored according to time [86].

definition nor a visualization concept for molecular cavities.

At the same time, Lindow et al. [88] presented another approch based on the Voronoi diagram of the atom spheres. It computes and visualizes molecular paths and cavities. By considering the different atom radii, the paths are geometrically optimal for probe spheres in contrast to previous approaches. Additionally, the paths can be filtered to get an overview of the most significant paths of the whole molecule. Besides path computation, the corresponding cavities can be extracted and visualized as a skin surface [28, 90]. Furthermore, the paper describes several possibilities to visualize the paths and cavities in combination with the surrounding molecule (see Figure 16, refer to Section 6 for more details). The temporal evolution of cavities was also studied by Lindow et al. [87]. In this extended version of their previous paper [86], they describe a visualization tool to analyze molecular dynamics trajectories. To do so, the paths and cavities are precomputed for each time step (see Figure 9 left) and correlated over time to keep track of their evolution. Afterwards, the user can interactively trace them over time while topological events like splits and merges of cavities are illustrated in plots. The tracing is computed by approximating the overlap of cavities of consecutive time steps. For an in-depth visual analysis, the cavities and the molecular structure are visualized accordingly. Furthermore, cavities related to each other over several time steps can be aggregated to visualize the 3D shape of dynamic channels or pockets in a static 3D visualization (see Figure 9 right). Additionally, the volume of cavities can be computed and the probability of cavities for the whole trajectory can be visualized in a single image.

Sridharamurthy et al. [124] also used the α-complex to identify robust voids and pockets, which are stable with respect to small perturbations in the atomic radii. The notion of robust voids is based on the stability and the topological persistence with

respect to the α-complex. First, the weighted Delaunay triangulation is computed on the set of atom centers. Second, the α-shape spectrum is constructed, which represents the filtration of the weighted Delaunay triangulation. Via the modification of filtration a set of stable voids is acquired. The implementation of this algorithm, *RobustVoids*, allows to visualize the results for different values of α.

In 2013, Kim et al. [60] presented a generalization of the CAST algorithm [85]. The algorithm, focusing on the detection of tunnels and voids via Voronoi diagrams and β-complexes, considers the correct atom radii by using the β-shape. The idea of the algorithm is to compute the Voronoi complement that corresponds to the skeleton of the molecular complement. Then, the tunnels and voids are recognized by analyzing the Voronoi complement. The algorithm was integrated to the software tools BetaTunnel and BetaVoid.

One of the most recent tools for automated characterization of voids is ChExVis, presented by Masood et al. [94]. The paper introduces their α-complex based method and a webserver, treating a large range of biological use cases with a focus on transmembrane channels. The method stores the occupied volume and centerlines of identified voids and can handle multiple objects simultaneously. The visualization and visual analysis is quite feature-rich, also mixing in physico-chemical descriptors such as hydrophobicity and conservation in a representation the authors call channel profiles (see Figure 18 (c) and Section 6). Handling of transmembrane pores seems to be a strong point of their approach.

5.6 Grid-Probe Methods

Another category of methods detecting cavities discussed in this survey combines the grid-based approach with the usage of a probe. A pioneering work that belongs to this category was presented by Voorintholt et al. [130], who developed a fast grid-based visualization of the Solvent Accessible Surface (SAS). In detail, each grid point inside the van der Waals sphere is assigned the value 100. Grid points with a larger distance than the van der Waals radius plus the radius of the probe are assigned the value 0. All grid points in between are assigned the value $(100 \cdot ((R_v + R_p)^2 - d^2)/((R_v + R_p)^2 - R_v^2)$, where R_v is the radius of the closest atom, R_p is the radius of the probe, and d is the distance to the closest atom position. Although this method does not explicitly extract cavities but only visualizes them as contours of the SAS derived from the grid data, it can be seen as a precursor for subsequent methods such as LIGSITE by Hendlich et al. [46].

LIGSITE is a tool for direction-based cavity detection that also maps the SAS into a three-dimensional grid. Afterwards, for each grid point outside the SAS, all neighboring grid points along the three main axes and the four cubic diagonal axes are investigated within 12 Å. If a grid point lies inside the SAS in both directions on one axis, this axis is marked as *protein, solvent, protein* (PSP). All grid points with at least two PSP directions are marked as cavity grid points and will be clustered. The surface of the cavities is obtained by sampling the solvent probe sphere at each cavity grid point. An improved version of LIGSITE, called LIGSITE^{CSC}, was published by Huang and Schroeder [50]. Additionally, the conservation of the neighboring residues of the three main pockets is analyzed to rate the availability of the pockets. The authors also compared the results with other tools—LIGSITE [46], PASS [6], SURFNET [76], and CAST [85].

Another approach is implemented in the tool HOLLOW by Ho and Gruswitz [48]. Instead of placing a sphere between each pair of atoms as in SURFNET [76], they place

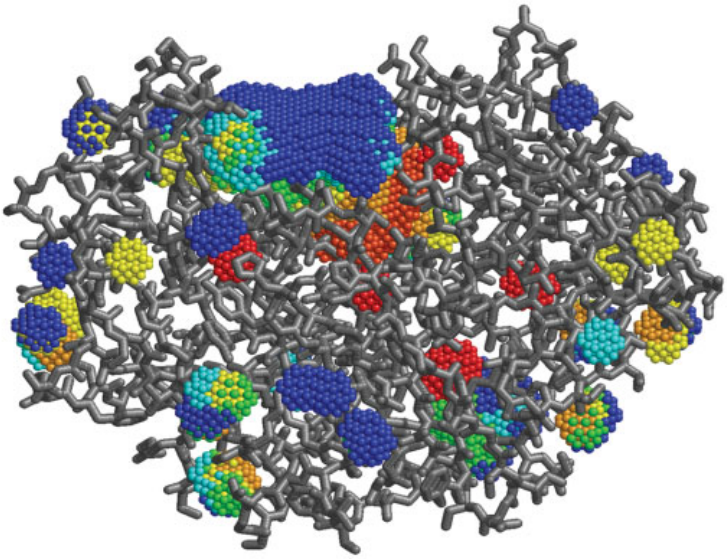

Figure 10: Multiscale pockets detected by the approach by Kawabata. Image source: [58].

them directly on a grid with a fixed sphere size. Afterwards, all spheres that penetrate the atom spheres or that lie outside the envelope of the molecule are removed from the grid. The remaining spheres of the grid are used as dummy atoms whose molecular surface represents the surface of the cavities.

In 2010, Kawabata proposed an algorithm to detect shallow and deep multiscale protein pockets [58]. The algorithm exploits a 3D grid representation and morphological operators. The grid is defined as an approximation of spheres representing the atoms. Afterwards, morphological operators are applied, where the structural element is defined by probe spheres of different sizes. The final formula, enabling to extract the protein pockets, is defined by a combination of opening and closing operators (see Figure 10). For each pocket, a measure of its shallowness is computed as the minimum inaccessible radius. The algorithm was implemented in a tool called GHECOM.

Recently, a technique for extracting cavities called PrinCCes was presented by Czirják [20]. It is essentially a multistage grid-based algorithm that uses flood-fill to iteratively mark the space with particular numerical encoding. Initially, two levels of probes—a large probe and a small probe—are placed to the center of each atom and sampled to a grid. The large probe separates the entire structure from the outer space. Then the space between the atoms and within the large probe space is defined as cavity candidate and finally the exact radii of the cavity extents at each grid point are calculated. The final void space is then segmented into separate structures based on the connectivity and intersection of the void space spheres. The technique was showcased on several examples from single proteins up to large protein complexes such as virus capsids.

5.7 Surface-based Methods

In contrast to most of the previous approaches, the protein is not purely restricted to the hard sphere model, but to a molecular surface model (Section 2.2). Molecular surfaces

define an interface of the molecule and its environment. Therefore, they can be used to define cavities as well. Surfaces like the Solvent Excluded Surface (SES), the Solvent Accessible Surface (SAS), or the Ligand Excluded Surface (LES) have the additional benefit that they define the interface with respect to a specific solvent or ligand. That is, cavities derived from these surfaces are also accessible by a solvent or ligand molecule of this size. For more information about molecular surfaces, please refer to the survey of Kozlíková et al. [66].

Sanner et al. [113] introduced the Reduced Surface, which is similar to α-shapes [31]. Based on the Reduced Surface of a molecule, its Solvent Excluded Surface can be computed. Sanner et al. also decribe how to compute the SES for internal voids, that is, enclosed cavities. The surface area and enclosed volume of the SES can be computed analytically for further analyses. The same idea—constructing the SES of internal voids to detect closed cavities—was recently also applied by Jurčík et al. [57]. Their method is an extension of the GPU-accelerated SES computation by Krone et al. [70], which is based on the Contour-buildup algorithm [127]. In contrast to Sanner et al., Jurčík et al. use an approximation of the surface area to describe the cavities.

A technique to compute all channels in a protein was developed by Coleman and Sharp [17]. Their tool CHUNNEL uses a triangulation of the SES provided by the GRASP tool as an input. Afterwards, all topological loops on the surface are detected as triangle strips. These strips characterize the channels in the molecule. In the final step, the topological paths through the channels and the corresponding loops are computed such that their distance to the surface becomes maximal. While the approach is among the first that automatically detect all channels, the algorithm is very slow and geometrically invalid channels can be detected due to circular singularities of the SES.

Parulek et al. [101, 102] introduced an implicit distance function that can be used to extract the SES. This distance function can also be used to detect the cavities of a protein. The approach involves a sampling strategy, where random but uniformly distributed samples are placed around the molecular surface. For each sample that is within a certain threshold, a ray is cast in the direction of the gradient of the distance field. If the ray hits the molecular surface, the sample is within a cavity (see Figure 11). For all samples that are within a cavity, a minimum spanning tree is computed, which can be used for substrate path analysis. Additionally, properties related to amino acids surrounding the cavity are computed to improve the parameter set describing each cavity. Although their method only considers one individual time step of a simulation trajectory, they propose the use of a scatterplot of the results for all time steps to assist users with the visual analysis of dynamic data.

Krone et al. [72] developed a method that extracts all types of cavities in real-time on the GPU. For each frame, a Gaussian density surface mesh is computed [73], which approximates the SES. For each triangle of this mesh, the Ambient Occlusion (AO) factor is computed (using the particle-based AO method of Grottel et al. [40]). If the AO is higher than a certain threshold, this part of the surface is classified as belonging to a cavity. Adjacent cavity triangles are collected into sub-meshes that represent the individual cavities. The evolution of the cavities is tracked over time by matching their centroids and additional properties such as the surface area of the cavities are computed. Krone et al. later extended their work to compute additional metrics of the cavities and to classify them into channels, pockets, and enclosed cavities [71]. Additionally, the length and width of a channel or pocket is computed based on the centerline.

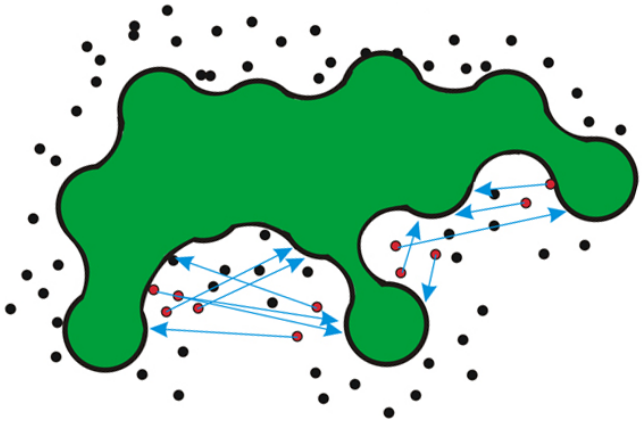

Figure 11: Illustration of the approach to the detection of cavities used by Parulek et al. Image source: [102].

5.8 Grid-Surface Methods

Methods combining surface-based and grid-based methods are also commonly used for cavity detection. This section reviews methods that fall into this category.

In 1994, Kleywegt and Jones [63] developed the tool VOIDOO to detect closed cavities in molecules. VOIDOO computes the Solvent Accessible Surface for a given probe on a discrete grid. Afterwards, all grid points that can be reached from the boundary of the grid are removed. Consequently, all remaining grid points outside the SAS are points inside closed cavities. These points can be used to create a surface of the cavities or to measure their volumes. The procedure is repeated several times with increasing scaling values for the atomic radii. The scale factor that creates the most cavities is finally used for further analyzes. However, the detection of this factor is not trivial and small variations can change the results a lot. Due to the nature of the algorithm, only closed cavities can be detected but not channels or pockets.

Similar to LIGSITE [46] presented in Section 5.6, Exner et al. [35] proposed a method that maps the SES into a discrete grid representation. For each grid point outside the SES, the grid points in the three main directions are investigated within a given neighborhood radius. If at least two directions contain grid points that lie inside the molecular surface in either direction, the investigated grid point is marked as belonging to a cavity. All cavity grid points are combined in clusters on which contraction and expansion operations are performed. The final clusters represent the cavities. The main shortcoming of this method is the limited detection directions. Depending on the neighborhood radius, this can lead to missing cavities whose medial shape axis is aligned diagonal to the main directions.

Yu et al. [134] presented the *Roll* algorithm for cavity detection, which is also based on the SES. Here, the volume of the cavities is defined as the difference of the volume enclosed by the SES and the volume enclosed by the van der Waals surface. To compute the difference efficiently, they sample the van der Waals surface to a 3D grid. Then, the SES is sampled by rolling the probe sphere along the grid without intersecting the atom spheres. The grid points between the SES and the van der Waals surface lie inside cavities. All cavities that are completely surrounded by the van der Waals surface are specified as closed cavities. On the other hand, cavities that are partially surrounded by the SES are denoted as pockets. By computing the volume depths of the cavities, a

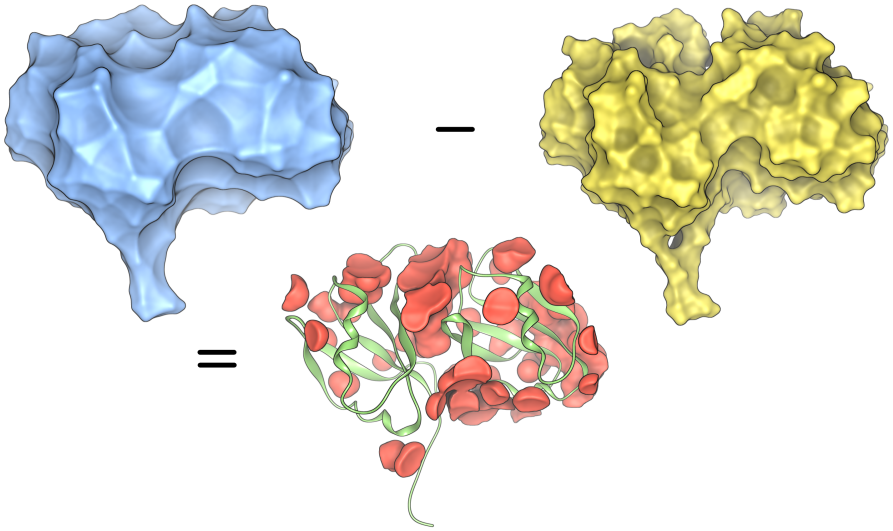

Figure 12: Result of a reimplementation of the tool 3V [131]. The difference between two solvent excluded surfaces with different probe radii results in the cavity structure in 3V.

ranking is achieved to easily detect potential binding sites. The Roll algorithm is used by the tool POCASA.

Till and Ullmann [126] presented another approach to extract cavities, called McVol. The approach computes the SAS of the protein as a discrete set of points using the method by Eisenhaber et al. [34]. Internal cavities are detected by connecting neighboring points of the SAS, followed by a connected components search on the resulting graph. Typically, the largest connected component represents the outer part of the SAS, while the other components represent the internal cavities. In addition, a second possibility to extract the internal cavities is proposed. To do so, further points are sampled inside the bounding box of the protein. If a point lies inside the SES it is marked as protein point otherwise it is marked as solvent point. Then, a grid is constructed, where each cell is marked as a solvent cell if at least one sample point in the cell is a solvent point, otherwise the cell is defined as a protein cell. Neighboring solvent cells are connected and again all connected components are detected, which results in the exterior of the protein as well as all internal cavities. Since this method does not detect pockets, the authors proposed a modification to extract them in a separate pass. For each solvent cell, all surrounding cells within a given cube are investigated. If the ratio of protein cells and solvent cells is larger than a user-defined threshold, the cell is marked as a pocket cell. Note that the accuracy of the algorithm depends on the number and quality of the point samplings. Furthermore, the definition of internal cavities and pockets is rather heuristic.

The 3V tool was introduced by Voss and Gerstein [131] in 2010 as a generalization of the Roll algorithm [134]. 3V computes the Solvent Excluded Surface for two probe spheres with different radius. The first probe approximates the substrate of interest. The second probe is larger and is used to extract the so-called shell surface that closes all outer pockets of the molecule. The volume of all cavities is defined as the difference of the volume enclosed by the shell and the volume enclosed by the SES of the substrate

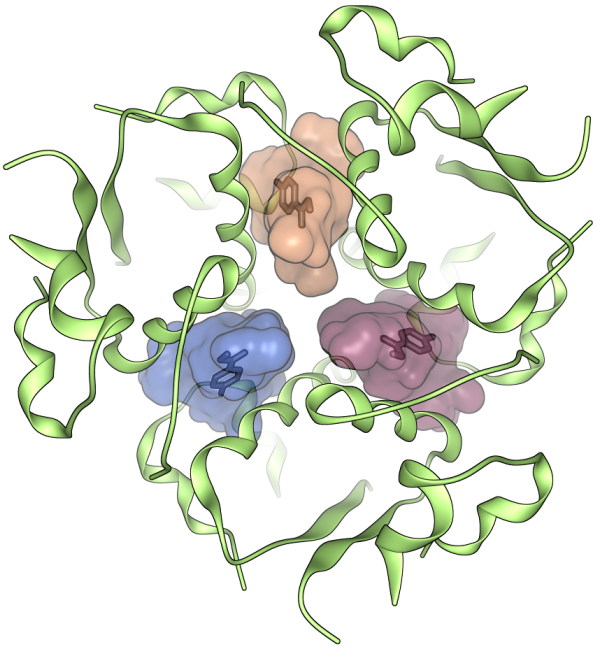

Figure 13: Three main cavities of hexameric insulin (PDB ID 3MTH) computed with the LES algorithm [89].

(Figur 12). The SES for both probe spheres are computed using a discrete grid. In 2014, the same method was proposed again by Oliveira et al. [99] in their tool KVFinder. In addition, Desdouits et al. [22] extract cavities in the same manner to study their evolution throughout MD simulations.

In 2014, Lindow et al. [89] proposed a method enabling the detection of all cavities based on the actual ligand geometry (Figure 13). It is based on an extension of the SES called Ligand Excluded Surface, which shows the accessibility for a specific ligand instead of an approximating single probe sphere. The grid-based algorithm to compute the surface computes intersection tests of the ligand with the receptor for each grid point, for a number of discrete ligand orientations and conformations. Grid points where the ligand can be placed are clustered according to the valid orientations and conformations that results in cores of cavities. An ambient occlusion technique is used to decide whether a grid point is inside or outside the boundary of the receptor. Only grid points inside the boundary are clustered. From these cores, the surface of the cavities is computed by sampling all valid ligand orientations and conformations into a discrete scalar field, which is visualized using Marching Cubes or direct volume ray casting.

5.9 Probe-Surface Methods

The last category includes combinations of surface-based and probe-based approaches. The only method that falls into this category is the PASS tool by Brady and Stouten [6]. PASS (Putative Active Sites with Spheres) enables the detection and measurement of

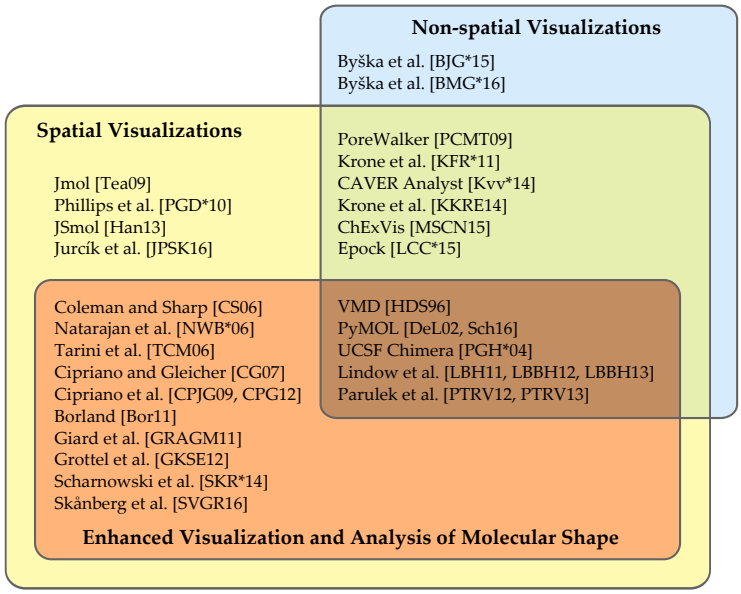

Figure 14: Three main categories of methods focusing on visualization and visual analysis of molecular cavities.

buried cavities inside the proteins, which also helps to identify the active site of the molecule. The algorithm uses layers of probe spheres to fill the cavities. Only exposed spheres that are surrounded by at least a certain number of atoms are kept. The first layer of spheres are placed tangent to the atom spheres by looping over all unique triplets of atoms. The next layers are placed in the same way, tangent to the previous layers. These spheres are then utilized to evaluate the size, shape, the extent of such cavities and the prediction of active sites. For a sphere, the active site is estimated by a number of neighboring spheres and the parameter describing the extent to which it is buried. Additionally, the system allows to visualize the residues close to the cavities as well as the cavities themselves.

6 Interactive Visual Analysis of Molecular Cavities

This section reviews the state of the art concerning the visualization of results produced by the previously described methods to extract cavities. The appropriate presentation of these results is an essential part for their analysis. This aspect is also visible in Figure 3, where cavity analysis methods that also introduced special methods for the visual analysis of these cavities are marked. The visual analysis of cavities is closely related to general methods for the visualization and visual analysis of molecular shape. This allows expert users a comprehensive visual analysis of cavities, binding sites, and related phenomena.

There are basically three main categories, spatial and non-spatial visualizations and enhanced visualization and analysis of molecular shape. The organization of papers in this section is illustrated in Figure 14, which shows their categorization. The image clearly shows high overlaps between these categories, which denotes that many of the papers are conveying the information about cavities by combining visualization techniques of more than one category.

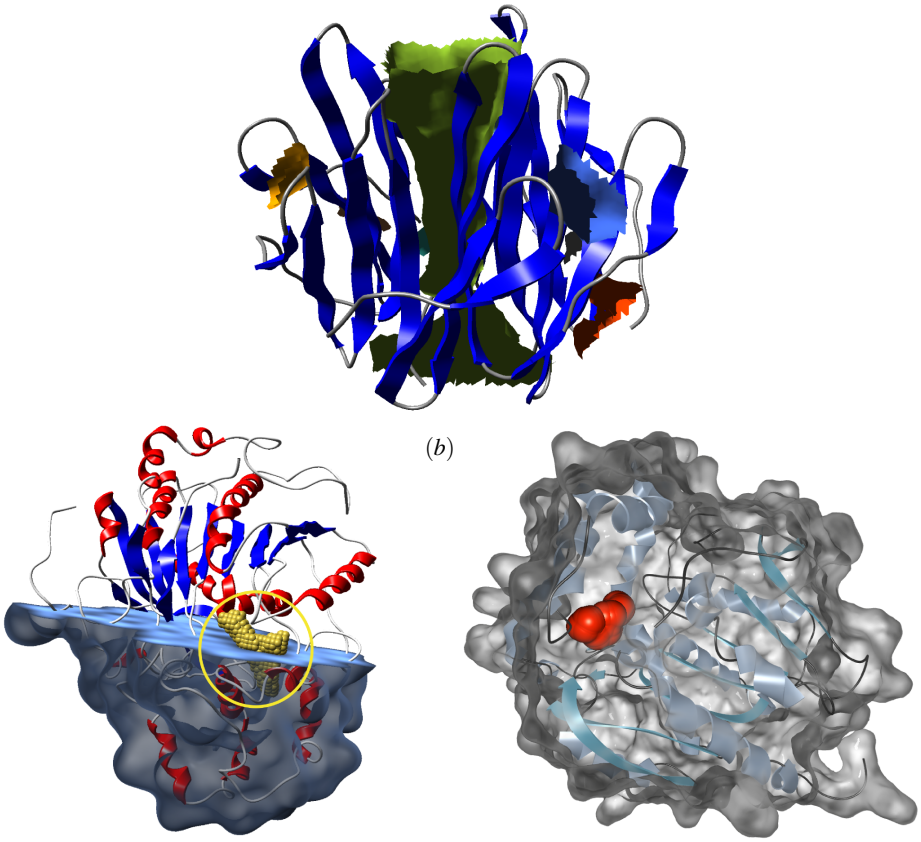

Figure 15: Visualizations of cavities in combination with the secondary structure and molecular surface. (a) Illustration of cavities as spheres on a uniform grid by Krone et al. [69], (b) Visualization of cavities as surface segments (combined with Cartoon rendering) in MegaMol [71, 39], (c) Visualization of cavities as spheres of Voronoi vertices in CAVER Analyst [67].

6.1 Spatial Visualizations

In many cases, the methods described in Section 5 do not introduce specific visualizations. Typically, well-established and commonly used molecular visualization tools like VMD [52], PyMOL [21, 116], or UCSF Chimera [106] are used to present the results. In these cases, the methods are either implemented as a plugin for these tools or the results are stored to a file that can be read by the respective tool (e.g., in the PDB file format [4] or as a PyMOL script file). While these molecular visualization tools offer a wide range of general-purpose visualizations for molecular data, they are not tailored to the visualization of cavities.

Most commonly, cavities are simply rendered as a set of spheres (Figure 15 (a), (c)), or can be represented by a molecular surface based on these spheres. For grid-based methods, isosurfaces derived from the grid are commonly used as a surface-based representation of the cavities. Many tools use polygonal isosurface extraction (e.g., Marching Cubes). In contrast, Krone et al. [69] used isosurfaces obtained by direct volume rendering, which can be beneficial in term of image quality, especially when rendering semi-transparent surfaces. Phillips et al. [109] also applied direct volume

rendering to visualize internal cavities. Another option that is often used to highlight a cavity is to color the atoms or amino acids of the protein that surround this cavity. For methods that extract a possible path through a pore or a tunnel, the tools will usually depict this path by a simple line strip. Paths can also be visualized as a set of spheres positioned on each node of the path. Here, the radius of each sphere is typically equal to the maximal radius of a hypothetical spherical probe that touches the surrounding atoms. In this case, the cavity extraction method has to provide the positions and radii of the spheres. Recent examples for such visualizations can be found in the works of Czirják [20] who used only spheres, or Kim et al. [59] who used spheres as well as centerlines. Especially in combination with cutting planes or transparency, these simple sphere-, surface-, or line-based representations already convey a lot of information about the cavities to the user.

Many methods described in Section 5 are accessible as a web service (see Section 7.2 for more details). Most of these web services use web-based molecular visualizations based on Jmol [56] or JSmol [43] to offer simple cavity visualizations similar to the ones described above (e.g., fpocket [80] or ChExVis [94]). An exception is PoreWalker [105], which presents only still images of the results that are pre-rendered on the server. The visualizations in these images are, however, similar to the simple visualizations described above (e.g., spheres that represent the path through the extracted pore).

Epock offers visualizations of cavity computation results through a plugin for the molecular visualization tool VMD [52]. It also includes Python scripts for plotting the results, for example, the evolution of cavity volume or the pore profile. The focus is on the visualization and analysis of the time evolution. Trajectories of the cavities can also be generated for visual analysis.

A stand-alone molecular visualization tool that focuses on cavities is CAVER Analyst [67], which uses the CAVER method [12]. Besides the most common representations for proteins like ball-and-stick, cartoon, and surfaces, CAVER Analyst also offers sphere- or surface-based representations of tunnels that show their path and width. Clipping planes and transparency further help users to see the interior of the proteins. The visual analysis tool for dynamic cavities presented by Krone et al. [71] is integrated in the molecular visualization framework MegaMol [39]. To convey the results of their cavity extraction, they either render only the surfaces of the cavities or use semi-transparent molecular surfaces for the exterior parts of the molecule to provide the context, whereas the parts of the molecular surface that demarcate cavities are rendered opaque (Figure 15 (b)). Similarly, Jurčík et al. [57] improved the visualization of transparent Solvent Excluded Surface to enable users to see the internal cavities in the context of the molecular surface without the need to slice through it. All these tools offer the usual coloring schemes that show physico-chemical properties of the proteins to support the analysis (e.g., by amino acid or hydrophobicity).

Lindow et al. [88, 86, 87] proposed a set of methods to highlight the paths in proteins that are extracted by their Voronoi diagram-based cavity extraction method. They filter the paths to show only the most relevant ones. Afterwards, they place many point light sources along the paths. Consequently, the molecular surface around the path is brightly lit. The paths themselves are rendered as tubes that follow a NURBS curve. Screen Space Ambient Occlusion is used to illustrate the general shape of the protein. Furthermore, they offer a view-dependent clipping to remove parts of the exterior surface that occlude user-selected cavities. Examples of the resulting visualization can be seen in Figures 16 and 17.

Parulek et al. [101] presented an interactive visual analysis approach to explore the

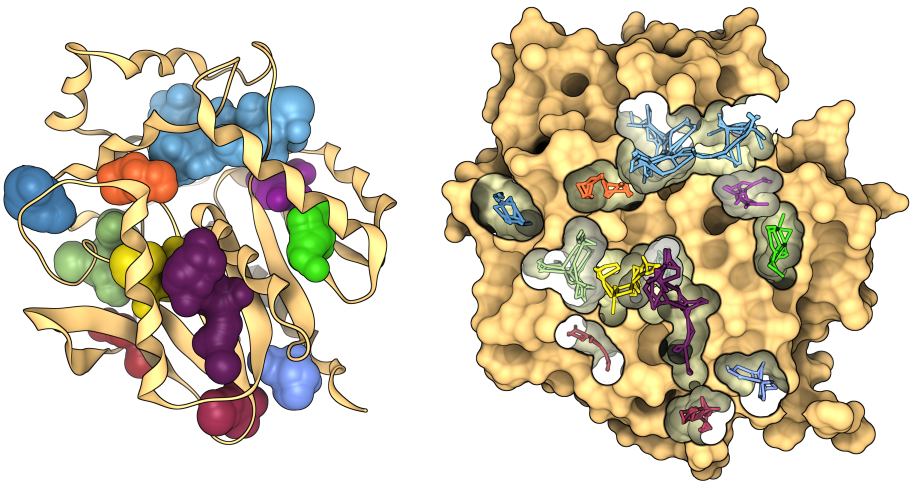

Figure 16: Visualization of Voronoi-based cavities [88]. Left: Cavities as skin surface in combination with the secondary structure. Right: Solvent Excluded Surface clipped by the cavities to show the corresponding paths.

space of cavities, i.e., their parameters, by means of a system linked views. This technique is a typical representative of combining spatial visualizations with non-spatial ones, which are discussed in the subsequent part. Parulek et al. use several types of scatterplots which allow users to interactively select the desired cavity parameters. Each point in the scatterplot represents a single cavity instance, where the user can opt between displaying two different parameters against each other or a single parameter over time. By brushing points in the scatterplot, all the linked views are automatically updated. In the accompanying 3D view, a focus-and-context visualization is utilized, where the molecular surface becomes more saturated around the cavity than the regions further away. Additionally, the user has the possibility to slice through the molecule while the visualization preserves the focus-and-context visualization style. In the follow-up study, Parulek et al. [102] enhanced the cavity parameter set by properties of the amino acids. The user can select cavities by specification of amino acids names in addition to their geometric characteristics. Chemical properties of the amino acids are color-coded near the selected cavities in the 3D visualization. Moreover, the user is provided with a dedicated linked view that shows evolution of chemical properties of selected cavities over time.

6.2 Non-spatial Visualizations

Another possibility is to present cavities and their properties using non-spatial visualizations. These methods can convey additional information and statistics about the cavities that are not easily discernible when using typical three-dimensional representations. Consequently, non-spatial visualizations are often used in concert with spatial ones to provide complementary information. This is especially helpful when analyzing dynamic data, where a spatial visualization would require either an animation or a temporal aggregation.

Conformational changes of a protein during a simulation can lead to interaction of cavities. For example, a cleft can merge with an internal cavity so that a pocket is formed. Lindow et al. [87] used a relational graph to show the evolution and interaction

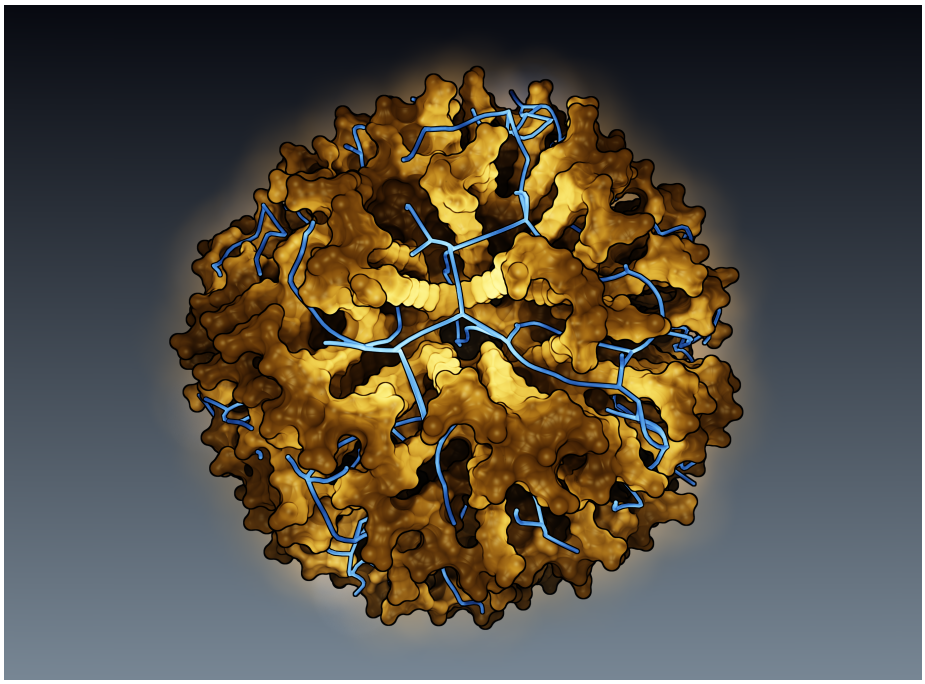

Figure 17: Illumination of cavities by placing many small point lights along potential molecular paths [88].

of cavities over time (e.g., splits or merges between cavities). A second graph shows the evolution of the cavities—that is, their position and spatial extent—over time (Figure 18 (a)). Krone et al. [72, 71] also used relational graph to show cavity evolution. Additional properties like the channel width or surface can be encoded in this graph.

In addition to the relational graph, Krone et al. employed several 2D line plots to show the profiles of the extracted cavity and to illustrate the temporal evolution of cavity properties like channel diameter or surface area in molecular simulation trajectories. This for example allows users to detect the narrowest sites of a tunnel (i.e., its bottlenecks). Similar 2D line plots that show the properties for dynamic data over time were used by Byška et al. [9] (see Figure 18 (b)). For each time step, they plot one line that shows the tunnel profile, which reveals the most stable and unstable parts of the tunnel over time. Below this profile plot, the amino acids that surround the tunnel are plotted. Each amino acid is represented by a colored strip consisting of individual lines. The number of these lines corresponds to the total number of time steps and their length depicts the extent of influence of the tunnel by this amino acid. Using this representation, users can detect amino acids that have a substantial contribution to a bottleneck, which might be candidates for protein mutations that influence the protein reactivity. The coloring can be changed according to different physico-chemical properties of the amino acids. A combination of non-spatial visualizations and spatial ones can, however, also be useful to provide additional information for static data. An example for this is ChExVis [94], which uses JSmol to show the geometry of cavities, as mentioned above. An additional 2D plot is used to show the profile of the currently selected channel, that is, its length and width in Ångströms, as well as the hydrophobicity profile along the channel (see Figure 18 (c)). Similar to the work of Byška et al., the visualization also shows the amino acids that are in contact with the channel.

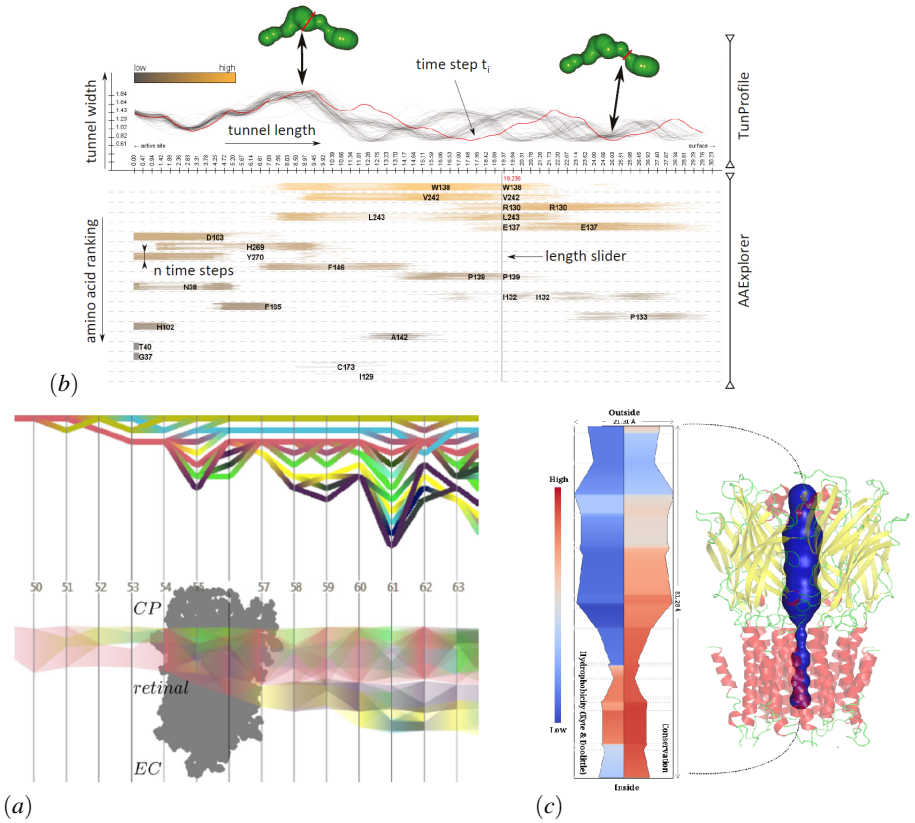

Figure 18: (a) Relational graph (top) showing splits and merges of cavites and evolution graph (bottom) illustrating the position and spatial extent of the cavities over time. (b) TunProfile (top) depicts the tunnel width and length for each time step. AAExplorer (bottom) shows the amino acids that surround the tunnel. (c) Combination of channel profile plot (left) and the classical three-dimensional representation of a channel (right) in the tool CHEXVIS. Image source: [87, 9, 94]

In another work, Byška et al. [8] proposed further methods to explore the shape as well as the properties of a selected tunnel in dynamics molecular data using 2D visualizations and plots. Heat map plots are either used to show the width of all tunnels or the evolution over time for one specific tunnel. The temporal evolution of the bottleneck of a single tunnel can also be explored in detail in one static image called the *MoleCollar* representation (see Figure 19). This view is enriched with abstract depictions of different physico-chemical properties of the amino acids surrounding the bottleneck. The idea of these methods is to show the most biochemically relevant tunnels and evaluate their throughput without tedious observation of all steps of the simulation trajectory.

6.3 Enhanced Visualization and Analysis of Molecular Shape

Rendering methods that highlight the shape of a molecule can be beneficial for the visual analysis of cavities. Tarini et al. [125] proposed a set of techniques to enhance the perception of molecular data, including contour lines and ambient occlusion. Borland [5] proposed Ambient Occlusion Opacity Mapping, which modulates the transparency of the molecular surface based on the ambient occlusion in order to emphasize

the internal structure—that is, the cavities—of a molecule (see Figure 20). Grottel et al. [40] presented an interactive ambient occlusion method for large, dynamic molecular data. Most recently, Skånberg et al. [119] proposed a combination of ambient occlusion and diffuse interreflections to highlight the entrance of the ligand to the active site located on the molecular surface. They use this technique also for the visualization of the interaction strength between a ligand and a receptor molecule. A recent review of methods for molecular visualization was given by Kozlíková et al. [66].

Besides methods that focus mainly on the visual appearance of the shape of a molecular surface, algorithms aiming to analyze the shape have been proposed as well. Visualizing the results of such shape descriptors cannot only assist the visual analysis of cavities but also provides a means to analyze and compare the shape of cavities. Cipriano et al.[14] proposed a multi-scale shape descriptor and applied it to molecular surface analysis. The descriptor estimates the degree of non-planarity and the anisotropy in a circular area around a given surface point. They showcased descriptor properties on a set of proteins while varying the value of the descriptor radius giving it a biological relevance. In a follow-up work [15], they showed how their shape descriptor can also be applied to surface matching, that is, to finding similar surface points among different proteins, which can be used to identify potential ligand binding sites. In their visualizations, Cipriano et al. use color to show the results of their surface descriptor together with ambient occlusion, which emphasizes the general shape of the molecule (see Figure 21 left).

Coleman and Sharp [16] introduced a shape descriptor called *travel depth*, which is defined as the shortest path for a small molecule from the convex hull to the molecular surface (see Figure 21 right). The small molecule can be represented by a probe sphere. This shape descriptor can be used to illustrate cavities on protein surfaces. Coleman and Sharp proposed a grid-based implementation to compute the travel depth. Subse-

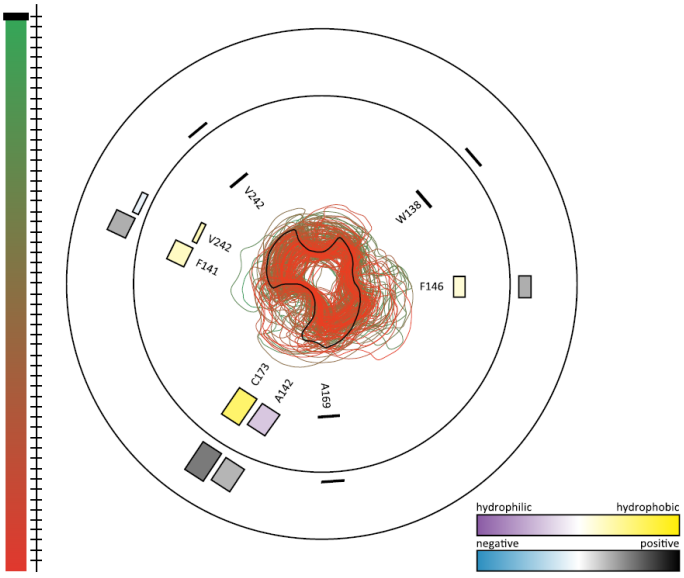

Figure 19: The MoleCollar representation by Byška et al. shows the bottleneck of a channel or tunnel over time. All time steps are superimposed and the surrounding glyphs denote the amino acids that are bodering the bottleneck. Image source: [8].

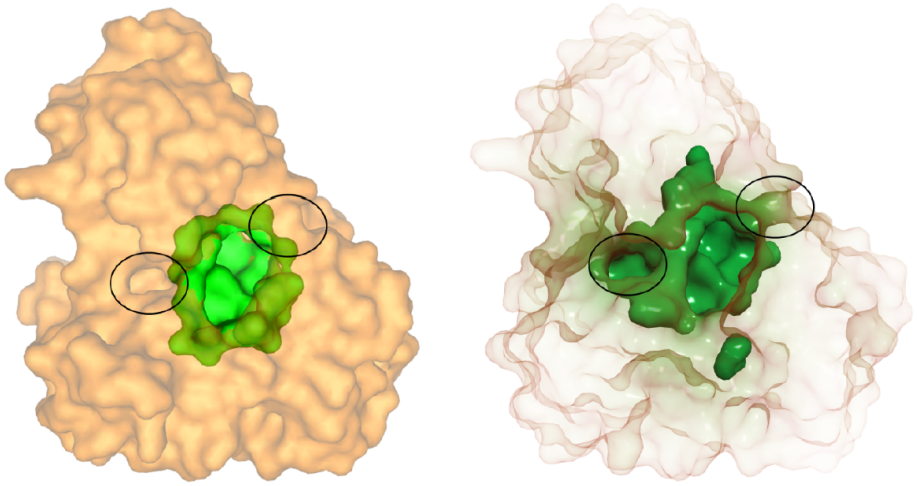

Figure 20: Comparison between CASTp (left) and the Ambient Occlusion Opacity Mapping (AOOM) proposed by Borland (right). AOOM also highlights the entrances to the channel (black circles). Image source: [5].

quently, Giard et al. [38] presented an algorithm for fast estimation of the travel depth based on a molecular surface mesh. The travel depth can be visualized by coloring the molecular surface according to the corresponding path length. Paths inside internal cavities are ignored in this approach.

Natarajan et al. [97] segment the molecular surface into grooves and pockets by means of a Morse-Smale complex. This includes the extraction of critical points on the surface and finding the appropriate edges of the Morse-Smale cells. The authors applied their method to correspondence matching of proteins in different conformations. Another application is the calculation of the atomic density function from the topological description. According to this density function, the authors are able to segment the molecular surface according to its protrusions and grooves and thus detect and visualize the protein pockets.

Scharnowski et al. [114] presented an algorithm for the pairwise comparison of local and global differences of molecular surfaces. After aligning the surfaces, one of them is deformed until it matches the second one. Local differences are derived from the deformation (geometric difference) as well as from the difference of the physicochemical properties between the matched surfaces. The global difference is measured by integrating over the local differences. Local differences are visualized using color and transparency, whereas global pairwise differences within an ensemble of proteins can be shown in a matrix plot. Similar to the methods of Cipriano et al. [15] or Natarajan et al. [97], this method can be used to analyze differences in the vicinity of a cavity. This can support users in drawing conclusions about accessibility and ligand binding.

Another visualization that focuses on the shape of a protein is the molecular surface abstraction proposed by Cipriano and Gleicher [13]. The method is based on geometric simplification of a meshed Solvent Excluded Surface. First, the mesh is simplified with a Taubin filter so that details that are smaller than an amino acid but bigger than an atom are smoothed. Afterwards, mid-sized features—indentations and protrusions—are removed based on their Gaussian curvature. Texture decals that represent the removed mid-sized feature (such as small and shallow pockets) are placed

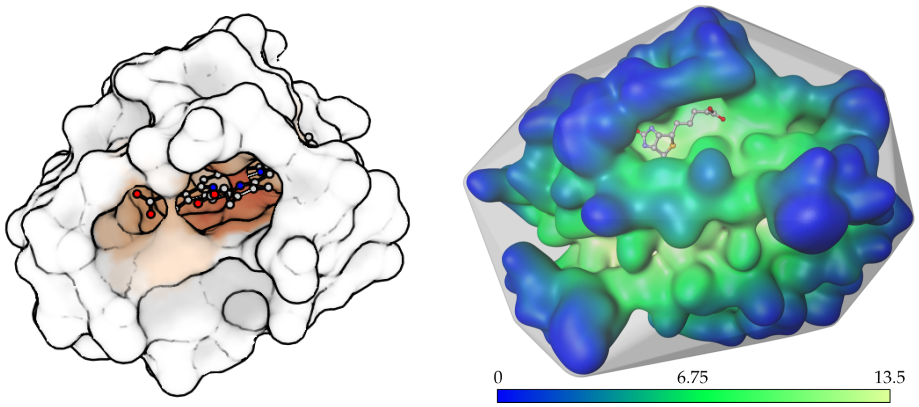

Figure 21: Different shape descriptors. Left: Multi-scale surface descriptor by Cipriano et al. [15]. Right: Reimplementation of travel depth by Coleman and Sharp [16].

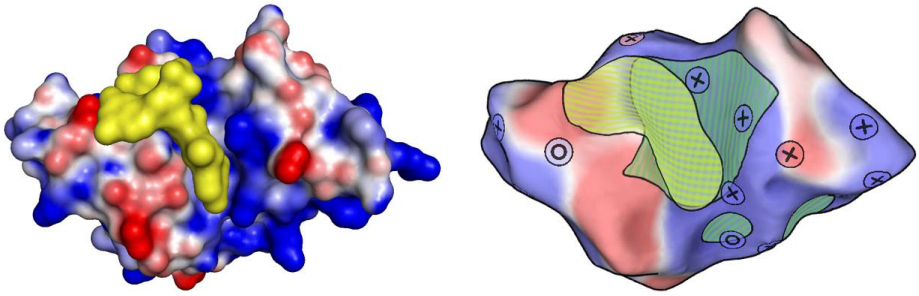

Figure 22: Left: Molecular surface of a protein colored by electrostatic potential with a ligand (yellow). Right: Molecular surface abstraction. Removed indentations and protrusions are illustrated by texture decals (ⓞ and ⊕). Putative ligand binding sites are highlighted in yellow on the surface. © 2007 IEEE. Reprinted, with permission, from [13].

in the corresponding location on the mesh surface. That is, high-frequency details of the surface are completely removed, but the overall shape is conserved. Consequently, the visibility of shape details is increased since the occlusion from mid-sized features has been resolved by the glyph replacement. The representation also allows for a clear representation of interaction sites. Interaction areas like binding sites are also projected onto the abstracted molecular surface using textures (see Figure 22).

7 Discussion

As shown in Section 5, there is a plethora of different methods to extract cavities from molecular data. Although the general goal of these methods is similar, they also use very different approaches to reach this goal. The methods are not only algorithmically different, but sometimes also in the way they define cavities. Consequently, the results are often also different. Apart from that, many of the tools that implement these methods offer additional analyses that provide users with more information about the cavities. This information can be an important factor for drawing conclusions about possible biological function of cavities. Individual tools might be optimal for a specific

analysis task, even though they can have deficiencies compared to other tools. That is, users have to choose carefully from the available methods depending on their research question or task. This of course also applies to visualization developers that devise new visual analysis methods for molecular cavities. They have to be aware that their choice of cavity extraction algorithm can influence the quality and utility of the results and, consequently, of their visual analysis tool.

One important aspect that we want to focus on is the analysis of dynamic data. The constant improvement of the capabilities of modern molecular simulations—resulting from improved hardware as well as improved simulation codes—leads not only to more accurate results for complex molecular systems, but also to longer simulation trajectories with large numbers of time steps. Today, molecular dynamics simulation has advanced to a point where it can be used to run virtual experiments that can provide novel insights into the characteristics and properties of molecular systems. Getting these insights, however, also requires analysis methods that are tailored to dynamic data. Additionally, conveying the analysis results to the user requires tailored visualization methods. The resulting visual analysis tools that provide information about the temporal evolution of features like cavities are important for users to understand the data and benefit from it. As observable in Figure 3, the number of methods and tools for the extraction of cavities that can deal with dynamic data is relatively small (green references). However, the figure also shows that in recent years, a trend towards dynamics data is emerging. This is also reflected in the increasing availability of visualization methods and visual analysis tools for cavities in dynamic data, which is described in Section 6. As mentioned above, visualization is crucial for the analysis of previously extracted cavities and their properties as well as secondary information. This is even more important for dynamic data, since the complexity rises with the number of time steps. As described in Section 6, modern visualization tools take different approaches to convey the evolution of the cavities to users. In general, one popular approach for visualizing dynamic data is to present the temporal information in one static representation. In case of visualizing cavities, this approach is often found in the non-spatial depictions described in Section 6.2. Spatial visualizations that show results for a whole trajectory in one static image often use aggregation to show the average cavity extent over time. However, the straightforward approach of visualizing temporal development as an animation is also found in cavity analysis and also has its benefits. Animation supports a more exploratory analysis of the data, where the user directly sees the changes over time. Even subtle changes are visible, in contrast to static depictions where detailed information might be lost due to aggregation or summarization. Furthermore, for in-situ visualization of interactively steered simulations, direct visualization of the results is the only possibility. Such scenarios of course pose the additional challenge that the cavity extraction algorithms as well as the visualizations have to be fast enough to be applicable in real time. In order to provide users with the benefits of both visualization approaches, a trend for the visual analysis that is apparent from Section 6 is the combination of spatial visualizations and non-spatial ones using multiple views. Such visual analysis tools can concurrently give an overview of the data and provide detailed views, as well as provide quantitative measurements.

A property that is currently taken into account by only few methods is the actual shape and orientation of a ligand within a cavity. Obvious reasons for that are the algorithmic complexity as well as the necessary computational power, which only recently became more widely available due to improvements in computing hardware. Available methods include the Ligand Excluded Surface by Lindow et al. [89] and TRAPP by Kokh et al. [65]. We think that this is an important challenge, especially for dynamic

data, since a method that considers detailed information about a ligand would be able to provide a more accurate estimation of the reachability of a binding site. An example for a tool that goes into this direction is the MoMA-LigPath web server [23], which simulates protein-ligand docking by calculating the ligand unbinding trajectory based on a simplified model considering mechanistic representation with partial flexibility. The protein-ligand complex serves as an input for the Manhattan-like Rapidly-exploring Random Tree (ML-RRT) that iteratively expands the search space of possible paths (similar to the work of Cortés et al. [18, 19]). This approach is borrowed from robotics, where it is used for path planning in mechanistic system.

7.1 Directions on the Comparison and Verification of Cavity Extraction Methods

A fundamental question concerns appropriate ways to compare and assess different methods, either on an individual basis or at a global scale. Here we want to provide a few directions and ideas for the verification and comparison of results and raise questions such as how to monitor the accuracy of measurements. As an example, we observed discrepancies in volume measurements among several tools, in some cases up to 200%. Currently no guidelines for a quantitative numerical comparison exist. More generally speaking, very few tools provide the option to measure inherent errors. This shortage makes it difficult to identify systematic errors, possibly induced by a chosen method or algorithm.

For the comparison of two methods, both the overall detected cavities and the related measurements should be taken into account. The detection could be handled by comparing the number of cavities found and assessing their similarity. Several studies provide ad hoc assessments, see for example [94] for a visual comparison of results from ChExVis, MOLE, CAVER, MolAxis, and PoreWalker for selected enzymes and transmembrane proteins. The detailed comparison of cavity characterization results by two methods could involve the identified surrounding amino acids, the cavity volume measurement, and, in the case of tunnels or pores—the path profile (e.g., width along centerline) measurement. A challenge is to move from qualitative to quantitative descriptors.

Rather than comparing methods individually, it would be desirable to have a gold standard for verification. In other fields, such as docking, dedicated benchmark data sets are used. For cavities, no commonly admitted reference data set currently exists. A database of biologically relevant cavities might be particularly useful in that respect. Some existing databases may form the basis for such a benchmark, for example free ones such as the pocketome one [74], or commercial ones such as, e.g., CavBase [75]. Specific use cases have been employed as a benchmark in several studies. For instance, the heat-shock protein HSP90 crystal structure collection was characterized through MDpocket, TRAPP, and Epock tools. A benchmark set including HSP90 can actually be downloaded from `http://epock.bitbucket.org/docs/epock_benchmarks.html`. A major limitation for these biological data sets is that measurements cannot be verified, e.g., the ”real” volume of a cavity is an unknown entity. For numerical and quantitative assessment, it may thus be better to resort to synthetic controlled data sets with known properties. One possibility is to generate artificial cavities of known shape and size, for example a sphere. We did so and Figure 23 provides an example assessment for illustration. A systematic underestimation of the volume is observed, with up to -40% for very small

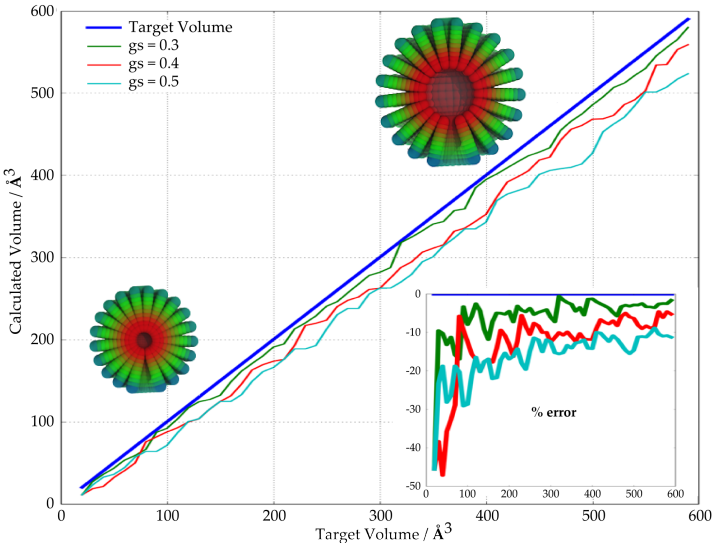

Figure 23: Example of volume measurement error assessment based on a sphere cavity trajectory of precisely controlled volume. The true sphere volume increases regularly (blue line). Three program settings are compared with percentual error indicated as inset. The smallest and largest spherical cavities are depicted. Data kindly provided by Dr. Benoist Laurent.

cavities. Changing program settings such as the grid spacing in this case can significantly reduce the error. Therefore, it has to be noted that the accuracy of results may be intimately linked to the choice of program options and parameters, with a likely tradeoff between accuracy and efficiency. When program authors provide recommended settings, those should be used for assessment. We provide the sphere trajectory as example benchmark system along with a brief discussion of errors: `http://epock.bitbucket.org/docs/epock_error.html`. Of course, a spherical shape may only represent certain types of cavities, hence other shapes should be tested as well, e.g., ellipsoids, cylinders, and more complex forms. In summary, comparison and verification of cavity extraction methods are issues that the community has to work on. Common guidelines for evaluation and assessment of methods and results need to be elaborated.

7.2 A Brief Overview of Available Tools

In addition to the technical descriptions, we want to provide an overview of the current availability of the tools discussed in the previous sections. Table 1 presents this information as a comprehensible overview, which gives the current status of these tools, the accessibility of their source code, and their availability on the three most often used platforms.

Note that this overview only includes freely available tools. However, there are also several commercial tools available which can be used for the computation and visualization of cavities. Among such tools belongs Molsoft's ICM-Pro (`http://www.molsoft.com/`), which uses the method of An et al. [1]. One of the most used software for cavity detection is SiteMap [42, 41] which enables to detect the binding sites and to classify the druggability of proteins. Another example is the YASARA software

tool by YASARA Biosciences whose YASARA View version [68] is freely available for PC as well as Android platforms. YASARA enables to locate cavities and calculate their volume. Other tools, such as SeeSAR by BioSolveIT (`http://www.biosolveit.de/SeeSAR/`) or MakeReceptor by OpenEye (`https://docs.eyesopen.com/oedocking/make_receptor_gui.html`), announce that they are able to compute cavities. However, no information about the algorithms used by these tools is available.

To complete the overview of the existing tools, we have to mention also solutions which do not come with their own algorithm for computation of cavities. These tools can be divided into two groups. The first group is formed by the general-purpose visualization tools, such as PyMOL or VMD, which are able to visualize the results of the computational tools by enabling the users to write their own plugins. The second group is represented by solutions that combine more tools in order to provide the users with more features at once, such as prediction of protein binding site and cavity detection. One such tool is ConCavity [11]. It enables to predict protein ligand binding sites by combining evolutionary sequence conservation with 3D structure. ConCavity makes use of LIGSITE [46], SURFNET [76], and PocketFinder [1] for the geometrical cavity detection. These algorithms are extended by a 'voting' of the cavities, which is based on the sequence conservation of the surrounding residues. To do so, the authors used the Jensen-Shannon divergence.

MetaPocket by Huang [49] is another tool that does not introduce a new cavity detection method but combines multiple other methods. It uses the results of LIGSITE^{CSC} [50], PASS [6], SURFNET [76], and Q-SiteFinder [78] to improve the identification of possible binding sites. More recently, Zhang et al. [136] presented MetaPocket 2.0, which takes into account four further tools, namely fpocket [80], GHECOM [58], ConCavity [11], and POCASA [134].

HotSpot Wizard [103] is a web server for automatic identification of "hot spots" in proteins and for annotation of protein structures. It integrates the structural, functional, and evolutionary information from different databases and tools. HotSpot Wizard searches for the amino acids located around buried cavities and pockets containing the active site and around the access tunnels to them. It utilizes CASTp [25] and CAVER [108] tools. The output of HotSpot Wizard consists of the list of annotated amino acids and is visualized in the web browser using Jmol. The tool is useful in the design of mutations in site-directed mutagenesis and focused directed evolution experiments.

8 Conclusions and Outlook

In this report we have reviewed and organized research work on molecular cavity detection, analysis, and visualization. The focus of the computer science research has been primarily targeted at protein-ligand binding. We can see that the algorithms on cavity detection have been maturing throughout the last years and now offer a variety of approaches that can be either based on discretization of the space, or topological analysis, *negative* surface extraction, or on probing that simulates the interaction of the ligand with the host macromolecule directly. The analytical methods are predominantly visualization-centric, although until recently, mostly direct 3D visualization techniques have been used for structural biology research workflows. While the availability of 3D visualization is essential, we can nowadays witness the emergence of tailored visualization methods that abstract the rich and overwhelming structural detail

Table 1: List of tools along with the availability of their source code (*Src*), availability for individual platforms (*Lin*: Linux, *Mac*: Apple OS X, *Win*: Microsoft Windows, *Web*: Web-based), current status (*Stat*, i.e., whether we were able to successfully run the tool on a small data set; + available, − not available, ? we were unable to verify), and visualization options (*Vis*, i.e., if the tool uses its own or an external visualization software or it just outputs the results to a file; *V* has its own visualization, *E* uses external visualization software, *F* writes results to file).

Tool	**Availability**					**Stat**	**Vis**
	Src	Lin	Mac	Win	Web		
3V[131]	+	?	?	?	+	+	E
BetaCavityWeb[61]	−	−	−	−	+	+	E
BioMOVE3D[19]	+	?	?	?	−	+	E
CAST[85]	−	−	−	−	−	−	E
CAVER 3[12]	+	+	+	+	+	+	E
CAVER Analyst 1[67]	−	+	+	+	−	+	V
ChExVis[94]	−	−	−	−	+	+	E,V
ConCavity[10]	+	?	?	?	+	+	?
dxTuber[111]	−	−	−	−	−	−	E
Epock[77]	+	+	+	+	−	+	E
FPocket[49]	+	+	+	+	+	+	E
GHECOM[58]	+	?	?	?	+	+	E
HOLLOW[48]	+	+	+	+	−	+	E
HotSpot Wizard[103]	−	−	−	−	+	+	E
LIGSITE[46]	−	−	−	−	−	−	F
LIGSITE^{CSC}[50]	+	+	−	−	+	+	E
McVol[126]	−	+	−	−	−	+	?
MDPocket[115]	+	+	+	+	+	+	E
MegaMol[71]	+	+	−	+	−	+	V
metaPocket[49]	−	−	−	−	+	+	E
MolAxis[133]	−	+	−	−	+	+	E
Mole 2[117]	+	+	+	+	+	+	V
MoMALigPath[23]	−	+	+	−	+	+	E
PASS[6]	−	+	−	−	−	−	F
POCASA[134]	−	−	−	−	+	+	E
POCKET[82]	−	−	−	−	−	−	?
PocketFinder[1]	−	?	?	?	?	?	?
PocketPicker[132]	+	+	−	+	−	+	E
PoreWalker[105]	−	−	−	−	+	+	E
PrinCCes[20]	−	+	−	+	−	+	E
Q-SiteFinder[78]	−	−	−	−	−	−	?
RobustVoids[124]	−	?	?	?	?	?	?
SITEHOUND[47]	+	+	+	+	+	?	E
SURFNET[76]	−	+	+	+	−	+	F
TRAPP[65]	−	−	−	−	+	?	E
VOIDOO[63]	−	+	+	−	−	+	F
Voroprot[98]	+	+	+	+	+	+	V

to simpler representations, which are tightly related to specific questions of the analyst and are also more quantitative. Moreover, by simplifying the complex spatio-temporal structure into simpler form, visualization estate is freed up for additional chemical and physical properties, as these should be considered together with the geometrical characteristics. We foresee that the trend of research in design studies that are tailored to specific analytical reasoning will continue in the context of ligand-protein interaction in the coming years.

With increasing simulation detail, more and more simulations are performed with the ligand contained in the simulated solution, and its interaction characteristics with the host macromolecule will be important to study. Visualization methodology will play here a central role. We can also foresee that visualization can in future allow for a semi-automatic protein engineering, where the parameter space of an entire ensemble of simulated mutations can be visually explored, and the iterative trial-error process can be significantly shortened. From the reviewed literature we can also deduce that the analytical methods for ligand-protein interaction are well developed and reach the stage of maturation. This is, however, not the case for the accompanying visualization technology. In the context of protein-protein interactions several analytical methods have been developed to date, but this technology is still emerging. The accompanying visualization technology that would align to typical questions of an analyst is practically non-existent. We see this subfield of structural biology as a large opportunity where the molecular visualization community can move to and as enabling technology that assists new discoveries. The protein-protein interactions can be the key for understanding large set of complex molecular machineries, which can have a strong impact on the advances in medicine, biology, and nanotechnology.

9 Acknowledgments

We want to thank D. Borland, J. Cortés, M. Gleicher, T. B. Masood, and G. Schneider for the permission to (re-)use images of their work or for providing us with new figures. We would like to thank B. Laurent for providing benchmark datasets for comparing and assessing cavity analysis methods. This work was supported through grants from the German Research Foundation (DFG) as part of SFB 716, the Senate Department for Economics, Technology and Research in Berlin, the French Agency for Research grants ExaViz (ANR-11-MONU-003) and Dynamo (ANR-11-LABX-0011), the Vienna Science and Technology Fund (WWTF) through project VRG11-010, the OeAD ICM through project CZ 17/2015, the Norway grants project NF-CZ07-MOP-2-086-2014, and the PhysioIllustration research project 218023 funded by the Norwegian Research Council.

References

[1] J. An, M. Totrov, and R. Abagyan. Pocketome via comprehensive identification and classification of ligand binding envelopes. *Mol. Cell. Proteomics*, 4(6):752–761, 2005.

[2] H. J. C. Berendsen, D. van der Spoel, and R. van Drunen. GROMACS: A message-passing parallel molecular dynamics implementation. *Comput. Phys. Commun.*, 91(1-3):43–56, 1995.

[3] K. Berka, O. Hanák, D. Sehnal, P. Banáš, V. Navrátilová, D. Jaiswal, C.-M. Ionescu, R. Svobodová Vařeková, J. Koča, and M. Otyepka. MOLEonline 2.0: interactive web-based analysis of biomacromolecular channels. *Nucleic Acids Res.*, 40(W1):W222–227, 2012.

[4] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The Protein Data Bank. *Nucl. Acids Res.*, 28(1):235–242, 2000.

[5] D. Borland. Ambient Occlusion Opacity Mapping for Visualization of Internal Molecular Structure. *Journal of WSCG*, 19(1):17–24, 2011.

[6] G. P. Brady Jr. and P. F. W. Stouten. Fast prediction and visualization of protein binding pockets with PASS. *J. Comput. Aided Mol. Des.*, 14(4):383–401, 2000.

[7] J. Brezovský, E. Chovancová, A. Gora, A. Pavelka, L. Biedermannová, and J. Damborský. Software tools for identification, visualization and analysis of protein tunnels and channels. *Biotechnol. Adv.*, 31(1):38–49, 2013.

[8] J. Byška, A. Jurčík, M. E. Gröller, I. Viola, and B. Kozlíková. MoleCollar and Tunnel Heat Map Visualizations for Conveying Spatio-Temporo-Chemical Properties Across and Along Protein Voids. *Comput. Graph. Forum*, 34(3):1–10, 2015.

[9] J. Byška, M. L. Muzic, M. E. Gröller, I. Viola, and B. Kozlíková. AnimoAminoMiner: Exploration of Protein Tunnels and their Properties in Molecular Dynamics. *IEEE Trans. Vis. Comput. Graphics*, 22(1):747–756, 2016.

[10] J. A. Capra, R. A. Laskowski, J. M. Thornton, M. Singh, and T. A. Funkhouser. Predicting Protein Ligand Binding Sites by Combining Evolutionary Sequence Conservation and 3d Structure. *PLoS Comput Biol*, 5(12):e1000585, 2009.

[11] J. A. Capra and M. Singh. Predicting functionally important residues from sequence conservation. *Bioinformatics*, 23(15):1875–1882, 2007.

[12] E. Chovancová, A. Pavelka, P. Beneš, O. Strnad, J. Brezovský, B. Kozlíková, A. Gora, V. Šustr, M. Klvaňa, P. Medek, L. Biedermannová, J. Sochor, and J. Damborský. CAVER 3.0: A Tool for the Analysis of Transport Pathways in Dynamic Protein Structures. *PLoS Comput. Biol.*, 8(10):e1002708, 2012.

[13] G. Cipriano and M. Gleicher. Molecular Surface Abstraction. *IEEE Trans. Vis. Comput. Graphics*, 13(6):1608–1615, 2007.

[14] G. Cipriano, G. N. Phillips Jr., and M. Gleicher. Multi-Scale Surface Descriptors. *IEEE Trans. Vis. Comput. Graphics*, 15(6):1201–1208, 2009.

[15] G. M. Cipriano, G. N. Phillips, and M. Gleicher. Local functional descriptors for surface comparison based binding prediction. *BMC Bioinformatics*, 13:314, 2012.

[16] R. G. Coleman and K. A. Sharp. Travel Depth, a New Shape Descriptor for Macromolecules: Application to Ligand Binding. *J. Mol. Biol.*, 362(3):441–458, 2006.

[17] R. G. Coleman and K. A. Sharp. Finding and Characterizing Tunnels in Macromolecules with Application to Ion Channels and Pores. *Biophys. J.*, 96(2):632–645, 2009.

[18] J. Cortés, T. Siméon, V. R. d. Angulo, D. Guieysse, M. Remaud-Siméon, and V. Tran. A Path Planning Approach for Computing Large-Amplitude Motions of Flexible Molecules. *Bioinformatics*, 21(suppl 1):i116–i125, 2005.

[19] J. Corts, S. Barbe, M. Erard, and T. Simon. Encoding Molecular Motions in Voxel Maps. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 8(2):557–563, 2011.

[20] G. Czirják. PrinCCes: Continuity-based geometric decomposition and systematic visualization of the void repertoire of proteins. *J. Mol. Graph. Modell.*, 62:118–127, 2015.

[21] W. L. DeLano. PyMOL: An Open-Source Molecular Graphics Tool. *CCP4 Newsletter On Protein Crystallography*, 40, 2002. http://pymol.sourceforge.net/.

[22] N. Desdouits, M. Nilges, and A. Blondel. Principal Component Analysis reveals correlation of cavities evolution and functional motions in proteins. *J. Mol. Graph. Model.*, 55:13–24, Feb 2015.

[23] D. Devaurs, L. Bouard, M. Vaisset, C. Zanon, I. Al-Bluwi, R. Iehl, T. Siméon, and J. Cortés. MoMA-LigPath: a web server to simulate protein-ligand unbinding. *Nucl. Acids Res.*, 41:297–302, 2013.

[24] R. Dias, J. de Azevedo, and F. Walter. Molecular docking algorithms. *Current Drug Targets*, 9(12):1040–1047, 2008.

[25] J. Dundas, Z. Ouyang, J. Tseng, T. A. Binkowski, Y. Turpaz, and J. Liang. CASTp: computed atlas of surface topography of proteins with structural and topographical mapping of functionally annotated residues. *Nucleic Acids Res.*, 34(Issue suppl 2):116–118, 2006.

[26] J. D. Durrant, C. A. de Oliveira, and J. A. McCammon. POVME: an algorithm for measuring binding-pocket volumes. *J. Mol. Graph. Model.*, 29(5):773–776, 2011.

[27] J. D. Durrant, L. Votapka, J. Sorensen, and R. E. Amaro. POVME 2.0: An Enhanced Tool for Determining Pocket Shape and Volume Characteristics. *J Chem Theory Comput*, 10(11):5047–5056, 2014.

[28] H. Edelsbrunner. Deformable Smooth Surface Design. *Discrete Comput. Geom.*, 21(1):87–115, 1999.

[29] H. Edelsbrunner, M. Facello, P. Fu, and J. Liang. Measuring Proteins and Voids in Proteins. In *Proceedings of the 28th Annual Hawaii International Conference on System Sciences*, volume 5, pages 256–264, 1995.

[30] H. Edelsbrunner, M. Facello, and J. Liang. On the definition and the construction of pockets in macromolecules. *Biocomputing: Proceedings of the 1996 Pacific Symposium*, pages 272 – 281, 1996.

[31] H. Edelsbrunner and E. P. Mücke. Three-dimensional Alpha Shapes. *ACM T Graphics (TOG)*, 13(1):43–72, 1994.

[32] M. Edes, C. Özturan, T. Halilolu, A. Luna, and R. Nussinov. MIMTool: A Tool for Drawing Molecular Interaction Maps. In *IEEE Symposium on Biological Data Visualization*, 2014. arXiv: 1407.2073.

[33] K. Edman, A. Hosseini, M. K. Bjursell, A. Aagaard, L. Wissler, A. Gunnarsson, T. Kaminski, C. Köhler, S. Bäckström, T. J. Jensen, et al. Ligand binding mechanism in steroid receptors: From conserved plasticity to differential evolutionary constraints. *Structure*, 23(12):2280–2290, 2015.

[34] F. Eisenhaber, P. Lijnzaad, P. Argos, C. Sander, and M. Scharf. The double cubic lattice method: Efficient approaches to numerical integration of surface area and volume and to dot surface contouring of molecular assemblies. *J. Comput. Chem.*, 16(3):273–284, 1995.

[35] T. Exner, M. Keil, G. Möckel, and J. Brickmann. Identification of Substrate Channels and Protein Cavities. *J. Mol. Model.*, 4(10):340–343, 1998.

[36] E. Fischer. Einfluss der Configuration auf die Wirkung der Enzyme. *Ber. Dtsch. Chem. Ges.*, 27(3):2985–2993, 1894.

[37] R. R. Gabdoulline, R. C. Wade, and D. Walther. MolSurfer: two-dimensional maps for navigating three-dimensional structures of proteins. *Trends Biochem. Sci.*, 24(7):285–287, Jul 1999.

[38] J. Giard, P. Rondao Alface, J.-L. Gala, and B. Macq. Fast Surface-Based Travel Depth Estimation Algorithm for Macromolecule Surface Shape Description. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 8(1):59–68, 2011.

[39] S. Grottel, M. Krone, C. Müller, G. Reina, and T. Ertl. MegaMol - A Prototyping Framework for Particle-based Visualization. *IEEE Trans. Vis. Comput. Graphics*, 21(2):201–214, 2015.

[40] S. Grottel, M. Krone, K. Scharnowski, and T. Ertl. Object-Space Ambient Occlusion for Molecular Dynamics. In *IEEE Pacific Visualization Symposium*, pages 209–216, 2012.

[41] T. Halgren. New method for fast and accurate binding-site identification and analysis. *Chem Biol Drug Des*, 69(2):146–148, 2007.

[42] T. A. Halgren. Identifying and characterizing binding sites and assessing druggability. *J Chem Inf Model*, 49(2):377–389, 2009.

[43] R. Hanson. JSmol: JavaScript-Based Molecular Viewer From Jmol, 2013. [Online; accessed 02.04.2014].

[44] M. Haranczyk and J. A. Sethian. Navigating molecular worms inside chemical labyrinths. *Proc. Natl. Acad. Sci. USA*, 106(51):21472–21477, 2009.

[45] J. Heinrich, M. Krone, S. I. O'Donoghue, and D. Weiskopf. Visualising Intrinsic Disorder and Conformational Variation in Protein Ensembles. *Faraday Discuss.*, 169:179–193, 2014.

[46] M. Hendlich, F. Rippmann, and G. Barnickel. LIGSITE: automatic and efficient detection of potential small molecule-binding sites in proteins. *J. Mol. Graphics Modell.*, 15(6):359 – 363, 1997.

[47] M. Hernandez, D. Ghersi, and R. Sanchez. SITEHOUND-web: a server for ligand binding site identification in protein structures. *Nucleic Acids Res.*, 37(suppl 2):W413–W416, 2009.

[48] B. K. Ho and F. Gruswitz. HOLLOW: Generating Accurate Representations of Channel and Interior Surfaces in Molecular Structures. *BMC Struct. Biol.*, 8(1):49+, 2008.

[49] B. Huang. MetaPocket: a meta approach to improve protein ligand binding site prediction. *Omics*, 13(4):325–330, 2009.

[50] B. Huang and M. Schroeder. LIGSITEcsc: Predicting Ligand Binding Sites Using the Connolly Surface and Degree of Conservation. *BMC Structural Biology*, 6(1):19, 2006.

[51] S.-Y. Huang. Search strategies and evaluation in proteinprotein docking: principles, advances and challenges. *Drug Discov. Today*, 19(8):1081–1096, 2014.

[52] W. Humphrey, A. Dalke, and K. Schulten. VMD Visual Molecular Dynamics. *J. Mol. Graph.*, 14:33–38, 1996.

[53] G. Iakovou, S. Hayward, and S. Laycock. A real-time proximity querying algorithm for haptic-based molecular docking. *Faraday Discuss.*, 169:359–377, 2014.

[54] R. M. Jackson. Q-fit: A probabilistic method for docking molecular fragments by sampling low energy conformational space. *J. Comput. Aided Mol. Des.*, 16(1):43–57, 2002.

[55] L. Jin, W. Wang, and G. Fang. Targeting Protein-Protein Interaction by Small Molecules. *Annu. Rev. Pharmacool. Toxicol.*, 54(1):435–456, 2014.

[56] Jmol Development Team. Jmol: an open-source Java viewer for chemical structures in 3d, 2009. http://www.jmol.org/, [Online; accessed 14.12.2016].

[57] A. Jurčík, J. Parulek, J. Sochor, and B. Kozlíková. Accelerated Visualization of Transparent Molecular Surfaces in Molecular Dynamics. In *IEEE Pacific Visualization Symposium*, 2016.

[58] T. Kawabata. Detection of multiscale pockets on protein surfaces using mathematical morphology. *Proteins: Struct., Funct., Bioinf.*, 78(5):1195–1211, 2010.

[59] B. Kim, J. E. Lee, Y. J. Kim, and K.-J. Kim. GPU Accelerated Finding of Channels and Tunnels for a Protein Molecule. *Int. J. Parallel. Prog.*, 44(1):87–108, 2016. (First online: 2014).

[60] D.-S. Kim, Y. Cho, J.-K. Kim, and K. Sugihara. Tunnels and Voids in Molecules via Voronoi Diagrams and Beta-Complexes. In M. L. Gavrilova, C. J. K. Tan, and B. Kalantari, editors, *Transactions on Computational Science XX*, number 8110 in Lecture Notes in Computer Science, pages 92–111. Springer Berlin Heidelberg, 2013.

[61] J.-K. Kim, Y. Cho, M. Lee, R. A. Laskowski, S. E. Ryu, K. Sugihara, and D.-S. Kim. BetaCavityWeb: a webserver for molecular voids and channels. *Nucl. Acids Res.*, page gkv360, 2015.

[62] L. J. Kingsley and M. A. Lill. Ensemble generation and the influence of protein flexibility on geometric tunnel prediction in cytochrome P450 enzymes. *PLoS ONE*, 9(6):e99408, 2014.

[63] G. J. Kleywegt and T. A. Jones. Detection, delineation, measurement and display of cavities in macromolecular structures. *Acta Crystallogr., Sect D: Biol. Crystallogr.*, 50(2):178–185, 1994.

[64] K. W. Kohn, M. I. Aladjem, J. N. Weinstein, and Y. Pommier. Molecular Interaction Maps of Bioregulatory Networks: A General Rubric for Systems Biology. *Mol. Biol. Cell.*, 17(1):1–13, 2006.

[65] D. B. Kokh, S. Richter, S. Henrich, P. Czodrowski, F. Rippmann, and R. C. Wade. TRAPP: A Tool for Analysis of Transient Binding Pockets in Proteins. *J. Chem. Inf. Model.*, 53(5):1235–1252, 2013.

[66] B. Kozlíková, M. Krone, N. Lindow, M. Falk, M. Baaden, D. Baum, I. Viola, J. Parulek, and H.-C. Hege. Visualization of Biomolecular Structures: State of the Art. In *Eurographics Conference on Visualization - STARs*, pages 61–81, 2015.

[67] B. Kozlíková, E. Šebestová, V. Šustr, J. Brezovský, O. Strnad, L. Daniel, D. Bednář, A. Pavelka, M. Maňák, M. Bezděka, P. Beneš, M. Kotry, A. Gora, J. Damborský, and J. Sochor. CAVER Analyst 1.0: Graphic tool for interactive visualization and analysis of tunnels and channels in protein structures. *Bioinformatics*, 30(18):btu364, 2014.

[68] E. Krieger and G. Vriend. YASARA Viewmolecular graphics for all devicesfrom smartphones to workstations. *Bioinformatics*, 30(20):2981–2982, 2014.

[69] M. Krone, M. Falk, S. Rehm, J. Pleiss, and T. Ertl. Interactive Exploration of Protein Cavities. *Comput. Graph. Forum*, 30(3):673–682, 2011.

[70] M. Krone, S. Grottel, and T. Ertl. Parallel Contour-Buildup Algorithm for the Molecular Surface. In *IEEE Symposium on Biological Data Visualization*, pages 17–22, 2011.

[71] M. Krone, D. Kauker, G. Reina, and T. Ertl. Visual Analysis of Dynamic Protein Cavities and Binding Sites. In *IEEE PacificVis - Visualization Notes*, volume 1, pages 301–305, 2014.

[72] M. Krone, G. Reina, C. Schulz, T. Kulschewski, J. Pleiss, and T. Ertl. Interactive Extraction and Tracking of Biomolecular Surface Features. *Comput. Graph. Forum*, 32(3):331–340, 2013.

[73] M. Krone, J. E. Stone, T. Ertl, and K. Schulten. Fast Visualization of Gaussian Density Surfaces for Molecular Dynamics and Particle System Trajectories. In *EuroVis - Short Papers*, volume 1, pages 67–71, 2012.

[74] I. Kufareva, A. V. Ilatovskiy, and R. Abagyan. Pocketome: an encyclopedia of small-molecule binding sites in 4d. *Nucleic Acids Res.*, 40(Database issue):D535–540, 2012.

[75] D. Kuhn, N. Weskamp, E. Hullermeier, and G. Klebe. Functional classification of protein kinase binding sites using Cavbase. *ChemMedChem*, 2(10):1432–1447, 2007.

[76] R. A. Laskowski. SURFNET: A program for visualizing molecular surfaces, cavities, and intermolecular interactions. *J. Mol. Graphics*, 13(5):323 – 330, 1995.

[77] B. Laurent, M. Chavent, T. Cragnolini, A. C. E. Dahl, S. Pasquali, P. Derreumaux, M. S. P. Sansom, and M. Baaden. Epock: rapid analysis of protein pocket dynamics. *Bioinformatics*, 31(9):1478–1480, 2015.

[78] A. T. R. Laurie and R. M. Jackson. Q-SiteFinder: an energy-based method for the prediction of protein-ligand binding sites. *Bioinformatics*, 21(9):1908–1916, May 2005.

[79] S. M. Lavalle. *Rapidly-exploring random trees: A new tool for path planning*. 1998.

[80] V. Le Guilloux, P. Schmidtke, and P. Tuffery. Fpocket: An open source platform for ligand pocket detection. *BMC Bioinformatics*, 10(1):168, 2009.

[81] C. H. Lee and A. Varshney. Computing and Displaying Intermolecular Negative Volume for Docking. In G.-P. Bonneau, T. Ertl, and G. M. Nielson, editors, *Scientific Visualization: The Visual Extraction of Knowledge from Data*, Mathematics and Visualization, pages 49–64. Springer, 2006.

[82] D. G. Levitt and L. J. Banaszak. POCKET: A Computer Graphics Method for Identifying and Displaying Protein Cavities and Their Surrounding Amino Acids. *J. Mol. Graph.*, 10(4):229–234, 1992.

[83] J. Liang, H. Edelsbrunner, P. Fu, P. V. Sudhakar, and S. Subramaniam. Analytical shape computation of macromolecules: I. Molecular area and volume through alpha shape. *Proteins*, 33(1):1–17, 1998.

[84] J. Liang, H. Edelsbrunner, P. Fu, P. V. Sudhakar, and S. Subramaniam. Analytical shape computation of macromolecules: II. Inaccessible cavities in proteins. *Proteins*, 33(1):18–29, 1998.

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

[86] N. Lindow, D. Baum, A.-N. Bondar, and H.-C. Hege. Dynamic Channels in Biomolecular Systems: Path Analysis and Visualization. In *IEEE Symposium on Biological Data Visualization*, pages 99–106, 2012.

[87] N. Lindow, D. Baum, A.-N. Bondar, and H.-C. Hege. Exploring cavity dynamics in biomolecular systems. *BMC Bioinformatics*, 14(Suppl 19):S5, 2013.

[88] N. Lindow, D. Baum, and H.-C. Hege. Voronoi-Based Extraction and Visualization of Molecular Paths. *IEEE Trans. Vis. Comput. Graphics*, 17(12):2025–2034, 2011.

[89] N. Lindow, D. Baum, and H.-C. Hege. Ligand Excluded Surface: A New Type of Molecular Surface. *IEEE Trans. Vis. Comput. Graphics*, 20(12):2486–2495, 2014.

[90] N. Lindow, D. Baum, S. Prohaska, and H.-C. Hege. Accelerated Visualization of Dynamic Molecular Surfaces. *Comput. Graph. Forum*, 29(3):943–952, 2010.

[91] W. E. Lorensen and H. E. Cline. Marching Cubes: A High Resolution 3d Surface Construction Algorithm. In *ACM SIGGRAPH Computer Graphics and Interactive Techniques*, volume 21, pages 163–169, 1987.

[92] R. Maciejewski, S. Choi, D. S. Ebert, and H. Z. Tan. Multi-modal perceptualization of volumetric data and its application to molecular docking. In *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics Conference*, pages 511–514, 2005.

[93] M. H. Maeda and K. Kinoshita. Development of new indices to evaluate protein-protein interfaces: Assembling space volume, assembling space distance, and global shape descriptor. *J. Mol. Graph. Modell.*, 27(6):706–711, 2009.

[94] T. B. Masood, S. Sandhya, N. Chandra, and V. Natarajan. CHEXVIS: a tool for molecular channel extraction and visualization. *BMC Bioinformatics*, 16:119, 2015.

[95] P. Medek, P. Beneš, and J. Sochor. Computation of tunnels in protein molecules using Delaunay triangulation. *Journal of WSCG*, 15(1-3):107–114, 2007.

[96] X.-Y. Meng, H.-X. Zhang, M. Mezei, and M. Cui. Molecular docking: A powerful approach for structure-based drug discovery. *Current Computer Aided-Drug Design*, 7(2):146–157, 2011.

[97] V. Natarajan, Y. Wang, P. T. Bremer, V. Pascucci, and B. Hamann. Segmenting Molecular Surfaces. *Comput. Aided Geom. D.*, 23(6):495–509, 2006.

[98] K. Olechnovič, M. Margelevicius, and C. Venclovas. Voroprot: an interactive tool for the analysis and visualization of complex geometric features of protein structure. *Bioinformatics*, 27(5):723–724, 2011.

[99] S. H. Oliveira, F. A. Ferraz, R. V. Honorato, J. Xavier-Neto, T. J. Sobreira, and P. S. de Oliveira. Kvfinder: steered identification of protein cavities as a pymol plugin. *BMC Bioinformatics*, 15(1):1–8, 2014.

[100] T. Paramo, A. East, D. Garzon, M. B. Ulmschneider, and P. J. Bond. Efficient Characterization of Protein Cavities within Molecular Simulation Trajectories: trj_cavity. *J. Chem. Theory Comput.*, 10(5):2151–2164, 2014.

[101] J. Parulek, C. Turkay, N. Reuter, and I. Viola. Implicit surfaces for interactive graph based cavity analysis of molecular simulations. In *IEEE Symposium on Biological Data Visualization*, 2012.

[102] J. Parulek, C. Turkay, N. Reuter, and I. Viola. Visual cavity analysis in molecular simulations. *BMC Bioinformatics*, 14(19):1–15, 2013.

[103] A. Pavelka, E. Chovancová, and J. Damborský. HotSpot Wizard: A Web Server for Identification of Hot Spots in Protein Engineering. *Nucl. Acids Res.*, 37(2):W376–383, 2009.

[104] A. Pavelka, E. Šebestová, B. Kozlíková, J. Brezovský, J. Sochor, and J. Damborský. CAVER: Algorithms for Analyzing Dynamics of Tunnels in Macromolecules. *IEEE ACM T. Comput. Bi.*, 2015.

[105] M. Pellegrini-Calace, T. Maiwald, and J. M. Thornton. PoreWalker: A Novel Tool for the Identification and Characterization of Channels in Transmembrane Proteins from Their Three-Dimensional Structure. *PLoS Comput. Biol.*, 5(7):e1000440, 2009.

[106] E. F. Pettersen, T. D. Goddard, C. C. Huang, G. S. Couch, D. M. Greenblatt, E. C. Meng, and T. E. Ferrin. UCSF Chimera - A Visualization System for Exploratory Research and Analysis. *J. Comput. Chem.*, 25(13):1605–1612, 2004.

[107] M. Petřek, P. Košinová, J. Koča, and M. Otyepka. MOLE: A Voronoi Diagram-Based Explorer of Molecular Channels, Pores, and Tunnels. *Structure*, 15(11):1357–1363, 2007.

[108] M. Petřek, M. Otyepka, P. Banáš, P. Košinová, J. Koča, and J. Damborský. CAVER: A New Tool to Explore Routes From Protein Clefts, Pockets and Cavities. *BMC Bioinform.*, 7(1):316, 2006.

[109] M. Phillips, I. Georgiev, A. K. Dehof, S. Nickels, L. Marsalek, H.-P. Lenhof, A. Hildebrandt, and P. Slusallek. Measuring Properties of Molecular Surfaces Using Ray Casting. In *Proc. Intl. Workshop on High Performance Computational Biology*, 2010.

[110] Z. Prokop, A. Gora, J. Brezovský, R. Chaloupková, V. Štěpánková, and J. Damborský. Engineering of Protein Tunnels: Keyhole-lock-key Model for Catalysis by the Enzymes with Buried Active Sites. In S. Lutz and U. Bornscheuer, editors, *Protein Engineering Handbook*, pages 421–464. Wiley-VCH, 2012.

[111] M. Raunest and C. Kandt. dxTuber: Detecting Protein Cavities, Tunnels and Clefts Based on Protein and Solvent Dynamics. *J. Mol. Graph. Modell.*, 29(7):895–905, 2011.

[112] N. Ray, X. Cavin, J. C. Paul, and B. Maigret. Intersurf: dynamic interface between proteins. *J. Mol. Graph. Model.*, 23(4):347–354, Jan 2005.

[113] M. F. Sanner, A. J. Olson, and J.-C. Spehner. Reduced Surface: An Efficient Way to Compute Molecular Surfaces. *Biopolymers*, 38(3):305–320, 1996.

[114] K. Scharnowski, M. Krone, G. Reina, T. Kulschewski, J. Pleiss, and T. Ertl. Comparative Visualization of Molecular Surfaces Using Deformable Models. *Comput. Graph. Forum*, 33(3):191–200, 2014.

[115] P. Schmidtke, A. Bidon-Chanal, F. J. Luque, and X. Barril. MDpocket: Open-Source Cavity Detection and Characterization on Molecular Dynamics Trajectories. *Bioinformatics*, 27(23):3276–3285, 2011.

[116] L. Schrödinger. The PyMOL Molecular Graphics System, Version 1.8. 2016.

[117] D. Sehnal, R. Svobodová Vařeková, K. Berka, L. Pravda, V. Navrátilová, P. Banáš, C.-M. Ionescu, M. Otyepka, and J. Koča. MOLE 2.0: advanced approach for analysis of biomacromolecular channels. *J. Cheminform.*, 5(1), 2013.

[118] P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker. Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks. *Genome Res.*, 13(11):2498–2504, 2003.

[119] R. Skanberg, P.-P. Vázquez, V. Guallar, and T. Ropinski. Real-time molecular visualization supporting diffuse interreflections and ambient occlusion. *IEEE Trans. Vis. Comput. Graph.*, 22(1):718–727, 2016.

[120] O. S. Smart, J. Breed, G. R. Smith, and M. S. Sansom. A novel method for structure-based prediction of ion channel conductance properties. *Biophys. J.*, 72(3):1109–1126, 1997.

[121] O. S. Smart, J. M. Goodfellow, and B. A. Wallace. The pore dimensions of gramicidin A. *Biophys. J.*, 65(6):2455–2460, 1993.

[122] O. S. Smart, J. G. Neduvelil, X. Wang, B. A. Wallace, and M. S. P. Sansom. HOLE: A program for the analysis of the pore dimensions of ion channel structural models. *J. Mol. Graph.*, 14(6):354–360, 1996.

[123] F. Spyrakis, P. Benedetti, S. Decherchi, W. Rocchia, A. Cavalli, S. Alcaro, F. Ortuso, M. Baroni, and G. Cruciani. A pipeline to enhance ligand virtual screening: Integrating molecular dynamics and fingerprints for ligand and proteins. *J Chem Inf Model*, 55(10):2256–2274, 2015. PMID: 26355717.

[124] R. Sridharamurthy, H. Doraiswamy, S. Patel, R. Varadarajan, and V. Natarajan. Extraction of robust voids and pockets in proteins. In *EuroVis-Short Papers*, pages 67–71, 2013.

[125] M. Tarini, P. Cignoni, and C. Montani. Ambient Occlusion and Edge Cueing for Enhancing Real Time Molecular Visualization. *IEEE Trans. Vis. Comput. Graphics*, 12(5):1237–1244, 2006.

[126] M. Till and G. M. Ullmann. McVol - A Program for Calculating Protein Volumes and Identifying Cavities by a Monte Carlo Algorithm. *J. Mol. Mod.*, 16(3):419–429, 2010.

[127] M. Totrov and R. Abagyan. The Contour-Buildup Algorithm to Calculate the Analytical Molecular Surface. *J. Struct. Biol.*, 116:138–143, 1995.

[128] I. Tuszynska, M. Magnus, K. Jonak, W. Dawson, and J. M. Bujnicki. NPDock: a web server for proteinnucleic acid docking. *Nucleic Acids Res.*, 43(W1):W425–W430, 2015.

[129] V. N. Uversky and A. K. Dunker. Understanding protein non-folding. *BBA Proteins Proteom.*, 1804(6):1231–1264, 2010.

[130] R. Voorintholt, M. T. Kosters, G. Vegter, G. Vriend, and W. G. J. Hol. A very fast program for visualizing protein surfaces, channels and cavities. *J. Mol. Graph.*, 7(4):243–245, 1989.

[131] N. R. Voss and M. Gerstein. 3V: cavity, channel and cleft volume calculator and extractor. *Nucl. Acids Res.*, 38(suppl 2):W555–W562, 2010.

[132] M. Weisel, E. Proschak, and G. Schneider. PocketPicker: Analysis of Ligand Binding-Sites with Shape Descriptors. *Chem. Cent. J.*, 1(1):7, 2007.

[133] E. Yaffe, D. Fishelovitch, H. J. Wolfson, D. Halperin, and R. Nussinov. MolAxis: Efficient and accurate identification of channels in macromolecules. *Proteins: Structure, Function, and Bioinformatics*, 73(1):72–86, 2008.

[134] J. Yu, Y. Zhou, I. Tanaka, and M. Yao. Roll: A new algorithm for the detection of protein pockets and cavities with a rolling probe sphere. *Bioinformatics*, 26(1):46–52, 2010.

[135] X. Yu, P. Nandekar, G. Mustafa, V. Cojocaru, G. I. Lepesheva, and R. C. Wade. Ligand tunnels in t. brucei and human cyp51: Insights for parasite-specific drug design. *Biochim. Biophys. Acta (BBA) - General Subjects*, 1860(1, Part A):67 – 78, 2016.

[136] Z. Zhang, Y. Li, B. Lin, M. Schroeder, and B. Huang. Identification of cavities on protein surface using multiple computational approaches for drug binding site prediction. *Bioinformatics*, 2011.