Published in final edited form as:

J Comput Chem. 2011 November 15; 32(14): 3023–3038. doi:10.1002/jcc.21884.

Geometric Measures of Large Biomolecules: Surface, Volume and Pockets

Paul Mach and

Graduate Group in Applied Mathematics, University of California, Davis, CA 95616

Patrice Koehl*

Department of Computer Science and Genome Center, University of California, Davis, CA 95616

Paul Mach: mach@math.ucdavis.edu; Patrice Koehl: koehl@cs.ucdavis.edu

Abstract

Geometry plays a major role in our attempt to understand the activity of large molecules. For example, surface area and volume are used to quantify the interactions between these molecules and the water surrounding them in implicit solvent models. In addition, the detection of pockets serves as a starting point for predictive studies of biomolecule-ligand interactions. The alpha shape theory provides an exact and robust method for computing these geometric measures. Several implementations of this theory are currently available. We show however that these implementations fail on very large macromolecular systems. We show that these difficulties are not theoretical; rather, they are related to the architecture of current computers that rely on the use of cache memory to speed up calculation. By rewriting the algorithms that implement the different steps of the alpha shape theory such that we enforce locality, we show that we can remediate these cache problems; the corresponding code, UnionBall has an apparent $\mathcal{O}(n)$ behavior over a large range of values of n (up to tens of millions), where n is the number of atoms. As an example, it takes 136 seconds with UnionBall to compute the contribution of each atom to the surface area and volume of a viral capsid with more than five million atoms on a commodity PC. UnionBall includes functions for computing the surface area and volume of the intersection of two, three and four spheres that are fully detailed in an appendix. UnionBall is available as an OpenSource software.

Keywords

space-filling diagrams; surface area; volume; pockets; macromolecules

1 Introduction

Cellular functions rest mostly on the activity of two types of large bio-molecules, namely proteins and nucleic acids. A fundamental finding that has shaped over the last forty years of studies of large molecules is that geometry plays a major role in our attempt to understand their activities. This paper emphasizes the former, i.e. the connection between geometry and chemistry. In particular we focus on measuring the shapes of molecules and detecting their cavities and pockets.

Significance of shape

The idea that shape defines function is a general concept from physical chemistry. Molecular structure or shape and chemical reactivity are highly correlated as the latter depends on the

*Corresponding author; koehl@cs.ucdavis.edu.

positions of the nuclei and electrons within the molecule. Indeed, chemists have long used three-dimensional plastic and metal models to understand the many subtle effects of structure on reactivity and have invested in experimentally determining the structure of important molecules. A common concrete model representing molecular shape is a union of balls, in which each ball corresponds to an atom. Properties of the molecule are then expressed in terms of properties of the union. For example, the putative active sites of an enzyme are detected as cavities and the interaction between a protein and its environment is quantified through the surface area and/or volume of the union of balls [1–4]. The most common use of molecular shape however is found in the quantification of the hydrophobic effect. For this, Lee and Richards introduced the concept of the solvent-accessible surface [5]. They computed the accessible area of each atom in both the folded and extended state of a protein, and found that the decrease in accessible area between the two states is greater for hydrophobic than for hydrophilic atoms. These ideas were refined by Eisenberg and McLachlan [1], who introduced the concept of a solvation free energy for large biomolecules, computed as a weighted sum of the accessible areas of all their atoms i . It is not clear, however, which surface area should be used to compute this solvation energy [6–8]. There is also some evidence that for small solute, the hydrophobic term is not proportional to the surface area [8], but rather to the solvent excluded volume of the molecule [9]. Current models for the non-polar part of the solvent energy include both a surface-based term and a volume-based term [10]. Within this debate on the exact form of the solvation energy, there is however a consensus that it depends on the geometry of the biomolecule under study, more specifically on its volume and surface area. In what follows, we discuss how these geometric measures are usually computed for a union of balls.

Geometric measures of biomolecules

The original approach of Lee and Richards [5] computed the accessible surface area by first cutting the molecule with a set of parallel planes. The intersection of a plane with an atomic ball, if it exists, is a circle which can be partitioned into accessible arcs on the boundary and occluded arcs in the interior of the union. The accessible surface area of atom i is the sum of the contributions of all its accessible arcs, computed approximately as the product of the arc length and the spacing between the plane. This method was originally implemented in the program ACCESS [5]. Shrake and Rupley [11] refined Lee and Richards' method and proposed a Monte Carlo numerical integration of the accessible surface area. Their method placed 92 points on each atomic sphere, and determined which points were accessible to solvent (not inside any other sphere). Efficient implementations of this method include applications of look-up tables [12], of vectorized algorithm [13] and of parallel algorithms [14]. Similar numerical methods have been developed for computing the volume of a union of balls [15–18].

The surface area and/or volume computed by numerical integration over a set of points, even if closely spaced, is not accurate and cannot be readily differentiated. To improve upon the numerical methods, analytical approximations to the accessible surface area have been developed, which either treat multiple overlapping balls probabilistically [19–21] or ignore them altogether [22, 23]. While these approaches are approximative, they are fast and lead to differentiable geometric measures. In addition, they are well suited for hardware acceleration on graphics processing units [24].

Better analytical methods describe the molecule as a union of pieces of balls, each defined by their center, radius, and arcs forming their boundary, and subsequently apply analytical geometry to compute the surface area and volume [25–29]. For example, Pavani and Ranghino [16] proposed a method for computing the volume of a molecule by inclusion-exclusion. In their implementation, only intersections of up to three balls were considered. Petitjean however noticed that practical situations for proteins frequently involve simultaneous overlaps of up to

six balls [28]. Subsequently, Pavani and Raghino's idea was generalized to any number of simultaneous overlaps by Gibson and Scheraga [30] and by Petitjean [28], applying a theorem that states that higher-order overlaps can always be reduced to lower-order overlaps [31]. Doing the reduction correctly remains however computationally difficult and expensive. The Alpha Shape Theory solves this problem using Delaunay triangulations and their filtrations, as described by Edelsbrunner [32].

The distinction between approximate and exact computation also applies to existing methods for computing the derivatives of the volume and surface area of a molecule with respect to its atomic coordinates [33–36]. In the case of the derivatives of the surface area, computationally efficient methods were implemented in the MSEED software by Perrot et al. [37] and in the SASAD software by Sridharan et al. [38]. All these methods introduce approximations to deal with singularities caused by numerical errors or by discontinuities in the derivatives [35, 37, 39].

Note that the complexity of the computation of the area and volume of a union of balls, the problems of singularities encountered when computing their derivatives, and the inherent existence of discontinuities have led to the development of alternative geometric representations of molecules. We cite for example the Gaussian description of molecular shape, that allows for easy analytical computation of surface area, volume and derivatives [40, 41], as well as the molecular skin that provides a smooth definition of the surface of a molecule [42].

Detecting pockets and cavities in biomolecular structure

The problem of detecting and measuring internal cavities of biomolecules is very popular as these cavities often serve as leads for drug design as they correspond to putative binding sites for these drugs. Most solutions to this problem rely heavily on geometry. They can be divided into three categories: (i) the grid based methods, (ii) the probe sphere detection and (ii) the analytical methods.

In the grid based method, the molecule is positioned in a 3D Cartesian grid whose vertices are then sorted into two groups: those that are covered by a protein atom and those that are not. The latter are further characterized as being inside a pocket if they satisfy some geometric conditions (such as being inside and at a distance greater than the radius of a water molecule from the convex hull of the biomolecule). The measures of these pockets (volume and surface area) are then computed by Monte Carlo integration over their corresponding grid points. POCKET [43], LIGSITE [44], LigandFit [45], PocketPicker [46], and McVol [18] are cavity-detecting programs that implement the grid-based method.

The probe sphere method proceeds by placing probe spheres that are tangent to the surfaces of two atoms of the biomolecules and then reducing their radii to eliminate overlaps with neighboring atoms; all remaining spheres whose radii exceed a minimal cutoff value (usually 1 Å) are used to define the pockets and cavities. This method was originally implemented in the program SURFNET [47] and later modified in the programs PASS [48] and PHECOM [49]. Interestingly, the grid and probe sphere methods were recently combined in the program POCASA [50].

The alpha shape theory combined with the discrete flow concept was the first analytical method proposed for detecting and measuring inaccessible cavities [4] as well as pockets [51, 52] in biomolecules. It has been extended since to detect channels between inner cavities and the outside [53]. The program CAVE implements a complementary approach in which the boundary of the pockets are directly triangulated, forming the so-called enveloping triangulation [54]

This work

Edelsbrunner and colleagues have developed analytical methods based on the alpha shape theory for computing the metrics of a union of balls, including surface area, volume, their derivatives with respect to the Cartesian coordinates of the centers of the balls and the detection and measurement of pockets [32, 55–57]. These methods have been implemented in different software packages, such as AlphaShape, CASTp [52, 58] and AlphaVol [57]. Most of these softwares however have not been recently updated; in addition, they have been written using generic algorithms that work fine for small molecules but have not been tested on vary large molecular systems (i.e. with more than one million atoms). In this paper, we show that these algorithms lead to inefficient programs for these very large systems. In response, we describe a new efficient implementation of these methods in an open source software package, UnionBall; this new implementation allows us to characterize and quantify the geometry of a molecular system with more than sixteen million atoms is less than eight minutes CPU time on a single processor running at 3.15 GHz. We also propose new geometric derivations of the equations that give the surface areas and volumes of the intersection of two, three and four balls, as well as their derivatives with respect to inter-atomic distances. The paper is organized as follows. The next section provides a brief description of the alpha shape theory and its application to measuring a union of balls. The following section describes our implementation of this theory in the program UnionBall; it includes testing on a set of large virus capsids. Appendix A covers the geometry of the intersections of two, three and four balls while appendix B describes the geometry of a tetrahedron.

2 Measuring Union of Balls

2.1 Surface area and volume of a union of balls

Given a collection P_i of N three dimensional sets, the volume and the surface area of the union of P can be computed using the principle of *inclusion-exclusion*. That is, the volume and surface area of the union $\cup P$ can be expressed as an alternating sum of volumes and surface areas of the common intersections of the subsets of P ,

$$\mathcal{M}(\cup P) = \sum_{i=1}^N \mathcal{M}(P_i) - \sum_{i < j} \mathcal{M}(P_i \cap P_j) + \sum_{i < j < k} \mathcal{M}(P_i \cap P_j \cap P_k) - \sum_{i < j < k < l} \mathcal{M}(P_i \cap P_j \cap P_k \cap P_l) + \dots \quad (1)$$

where $\mathcal{M}$ stands for the volume $\mathcal{V}$ of the union of sets or the area of its boundary $\mathcal{A}$. There are two issues that need to be solved to make these equations computationally tractable. Firstly, we need to have a consistent way to reduce significantly the number of terms in the Inclusion-Exclusion formula; brute force application of this formula would lead to an algorithm with exponential running time, as the total number of terms in in formula in $2^N - 1$, with each term corresponding to the measure of the intersection of at most N balls. Secondly, we need analytical formula for computing the non-empty intersection of sets. The next two sections overview solutions to these two issues when the sets are 3D balls.

2.1.1 A simplified inclusion-exclusion formula for union of balls—The Alpha Shape Theory provides a method for reducing significantly the number of terms in the inclusion-exclusion formula applied to unions of balls. It is based on the concept of Voronoi decompositions and Delaunay triangulations and their filtrations, as described by Edelsbrunner [32]. Note that the concept of using the Voronoi decomposition and Delaunay triangulation to simplify the inclusion-exclusion formula was originally introduced by Naiman and Wynn [59].

Voronoi decomposition and dual complex: Let us consider a finite set of spheres S_i with centers z_i and radii r_i and let B_i be the ball bounded by S_i . We define the square distance between a point x and a sphere S_i as $\pi_i(x) = \|x - z_i\|^2 - r_i^2$. This distance definition allows for varying radii for the spheres.

The *Voronoi region* of S_i consists of all points x at least as close to S_i as to any other sphere: $V_i = \{x \in \mathbb{R}^3 \mid \pi_i(x) \leq \pi_j(x)\}$. The Voronoi region of S_i is a convex polyhedron obtained as the common intersection of finitely many closed half-spaces, one per sphere $S_j \neq S_i$. These half-spaces are defined as follows. If S_i and S_j intersect in a circle then the plane bounding the corresponding half-spaces passes through that circle. The union of all Voronoi regions V_i defines the *Voronoi diagram* of the union of spheres; this union covers the whole space. The intersection of the Voronoi diagram with the union of balls B_i decomposes this union into convex regions of the form $B_i \cap V_i$, as illustrated in figure 1. The boundary of each such region consists of spherical patches on S_i and planar patches on the boundary of V_i . The spherical patches separate the inside from the outside and the planar patches decompose the inside of the union.

The *Delaunay triangulation* is the dual of the Voronoi diagram, obtained by drawing an edge between the centers of S_i and S_j if the two corresponding Voronoi regions share a common face. Furthermore, we draw a triangle connecting z_i, z_j and z_k if V_i, V_j and V_k intersect in a common line segment, and we draw a tetrahedron connecting z_i, z_j, z_k and z_ℓ if V_i, V_j, V_k and V_ℓ meet at a common point. Assuming general position of the spheres, there are no other cases to be considered. We refer to this as the *generic case*; it is important to mention that it is rare in practice because of limited precision. Nevertheless, it is possible to simulate a perturbation of the union of balls that restores the generic case [60]. This method, referred to as *simulation of simplicity*, consistently unfold potentially complicated degenerate cases to non-degenerate ones.

Let us limit the construction of the Delaunay triangulation to within the union of balls. In other words, we draw a dual edge between the two vertices z_i and z_j only if $B_i \cap V_i$ and $B_j \cap V_j$ share a common face, and similarly for triangles and tetrahedra. The result is a sub-complex of the Delaunay triangulation which we refer to as the *dual complex* K of the set of spheres.

Area and volume formulas: A simplex τ in the dual complex can be interpreted abstractly as a collection of balls, one ball if it is a vertex, two if it is an edge, etc. In this interpretation, the dual complex is a system of sets of balls. We write $\text{vol} \cap \tau$ for the volume of the intersection of the balls in τ . This is exactly the term we would see in an inclusion-exclusion formula for the volume of the union of balls, $\cup_i B_i$. As proved in [32, 59], the inclusion-exclusion formula that corresponds to the dual complex gives the correct volume of a union of balls, as well as the correct area of its boundary.

We state the corresponding theorems for the case in which the contribution of each ball B_i is weighted by a constant α_i , yielding the weighted volume $\mathcal{V}_w$ of the union of balls and weighted area $\mathcal{A}_w$ of its boundary. When the coefficients α_i correspond to atomic solvation parameters, these two terms estimate the solvation free energy of the molecule represented by the union of balls. Let τ_i be the simplex corresponding to the ball B_i , τ_{ij} the simplex formed by the edge between the balls B_i and B_j , τ_{ijk} the triangle corresponding to the three balls B_i, B_j and B_k , and finally τ_{ijkl} the tetrahedron defined by the four balls B_i, B_j, B_k and B_l . then:

Weighted Volume Theorem

$$\mathcal{A}_w(\cup_i B_i) = \sum_{\tau_i \in K} \alpha_i \mathcal{A}_i - \sum_{\tau_{ij} \in K} (\alpha_i \mathcal{A}_{i,j} + \alpha_j \mathcal{A}_{j,i}) + \sum_{\tau_{ijk} \in K} (\alpha_i \mathcal{A}_{i,jk} + \alpha_j \mathcal{A}_{j,ik} + \alpha_k \mathcal{A}_{k,ij}) - \sum_{\tau_{ijkl} \in K} (\alpha_i \mathcal{A}_{i,jkl} + \alpha_j \mathcal{A}_{j,ikl} + \alpha_k \mathcal{A}_{k,ijl} + \alpha_l \mathcal{A}_{l,ijk}) \quad (2)$$

and

Weighted Volume Theorem

$$\mathcal{V}_w(\cup_i B_i) = \sum_{\tau_i \in K} \alpha_i \mathcal{V}_i - \sum_{\tau_{ij} \in K} (\alpha_i \mathcal{V}_{i,j} + \alpha_j \mathcal{V}_{j,i}) + \sum_{\tau_{ijk} \in K} (\alpha_i \mathcal{V}_{i,jk} + \alpha_j \mathcal{V}_{j,ik} + \alpha_k \mathcal{V}_{k,ij}) - \sum_{\tau_{ijkl} \in K} (\alpha_i \mathcal{V}_{i,jkl} + \alpha_j \mathcal{V}_{j,ikl} + \alpha_k \mathcal{V}_{k,ijl} + \alpha_l \mathcal{V}_{l,ijk}) \quad (3)$$

Here $\mathcal{V}_i$ is the volume of the ball B_i , $\mathcal{V}_{ij}$ is the contribution of B_i to the volume of the intersection of the balls B_i and B_j , etc. Similar definitions are used for the surface areas $\mathcal{A}$.

These results are direct extensions of the Area and Volume Theorems derived by Edelsbrunner [32, 57]; they overcome past difficulties by implicitly reducing higher-order to lower-order overlaps. An added advantage of these formulas is that the balls in each term form a unique geometric configuration so that the analytic calculation of the volume can be done without case analysis [32, 57].

As a side note, it is interesting that the dual complex is not the only simplicial complex that leads to a minimal inclusion exclusion formulas: Attali and Edelsbrunner have shown that it is possible to construct a family of such complexes, that are characterized by the independence of their simplices and by geometric realizations with the same underlying space as the dual complex [61].

2.1.2 Angle weighted inclusion-exclusion formula for union of balls—Even though the equations described above are minimal, it is possible to find even shorter expressions for the weighted areas and volumes if non integer coefficients are considered. This is what is referred to as the short inclusion-exclusion method and is described in detail in [32]. In this method, the area and volume are expressed as the sums of the contributions of intersections of at most three balls, with angular coefficients. The corresponding expressions for the weighted areas and weighted volumes are:

Short Weighted Area Formula

$$\mathcal{A}_w(\cup_i B_i) = \sum_{\tau_i \in K} \alpha_i \gamma_i \mathcal{A}_i - \sum_{\tau_{ij} \in K} \gamma_{ij} (\alpha_i \mathcal{A}_{i,j} + \alpha_j \mathcal{A}_{j,i}) + \sum_{\tau_{ijk} \in K} \gamma_{ijk} (\alpha_i \mathcal{A}_{i,jk} + \alpha_j \mathcal{A}_{j,ik} + \alpha_k \mathcal{A}_{k,ij}) \quad (4)$$

and

Short Weighted Volume Formula

$$\begin{aligned} \mathcal{V}_w(\cup_i B_i) = & \sum_{\tau_i \in K} \gamma_i \alpha_i \mathcal{V}_i - \sum_{\tau_{ij} \in K} \gamma_{ij} (\alpha_i \mathcal{V}_{i,j} + \alpha_j \mathcal{V}_{j,i}) \\ & + \sum_{\tau_{ijk} \in K} \gamma_{ijk} (\alpha_i \mathcal{V}_{i,jk} + \alpha_j \mathcal{V}_{j,ik} + \alpha_k \mathcal{V}_{k,ij}) \\ & + \sum_{\tau_{ijkl} \in K} (\alpha_i \text{vol}(F_i) + \alpha_j \text{vol}(F_j) + \alpha_k \text{vol}(F_k) + \alpha_l \text{vol}(F_l)) \end{aligned} \quad (5)$$

where F_i is the fraction of the Voronoi region of S_i delimited by the planes defined by the triangles $\Delta z_i z_j z_k$, $\Delta z_i z_j z_l$ and $\Delta z_i z_k z_l$. We show in appendix A how to compute the volume of F_i .

The coefficients γ are the normalized exposed angles of the simplices [57]; they are given by:

$$\gamma_i = 1 - \sum_{j,k,l | \tau_{ijkl} \in K} \frac{\Omega_i}{4\pi} \quad (6)$$

$$\gamma_{ij} = 1 - \sum_{k,l | \tau_{ijkl} \in K} \frac{\varphi_{ij}}{2\pi} \quad (7)$$

$$\gamma_{ijk} = 1 - \sum_{l | \tau_{ijkl} \in K} \frac{1}{2} \quad (8)$$

where Ω_i is the solid angle at vertex z_i and φ_{ij} is the dihedral angle associated with the edge $z_i z_j$ in the tetrahedron defined by z_i, z_j, z_k and z_l . These coefficients can be interpreted as the fraction of solid angle (for a vertex), of dihedral angle (for an edge) or face of triangle that remains accessible in the dual complex. All edges and triangles that are fully buried have zero contribution in equations 4 and 5. In parallel, tetrahedra in the dual complex that are fully buried do not contribute to the area, and only contribute their volume (which is easier to compute than the volume of the intersection of balls) in the volume formula.

Note that in the special case that the weights of all atoms are equal to 1, these equations give the surface area and volume of a union of balls and can be written:

$$\mathcal{A}(\cup_i B_i) = \sum_{\tau_i \in K} \gamma_i \mathcal{A}(B_i) - \sum_{\tau_{ij} \in K} \gamma_{ij} \mathcal{A}(B_i \cap B_j) + \sum_{\tau_{ijk} \in K} \gamma_{ijk} \mathcal{A}(B_i \cap B_j \cap B_k) \quad (9)$$

and

$$\mathcal{V}(\cup_i B_i) = \sum_{\tau_i \in K} \gamma_i \mathcal{V}(B_i) - \sum_{\tau_{ij} \in K} \gamma_{ij} \mathcal{V}(B_i \cap B_j) + \sum_{\tau_{ijk} \in K} \gamma_{ijk} \mathcal{V}(B_i \cap B_j \cap B_k) + \sum_{\tau_{ijkl} \in K} \text{vol}(\tau_{ijkl}) \quad (10)$$

2.1.3 Area and volume derivatives—We are interested in the derivatives of the area and the volume of a union of N balls with respect to their positions. We have recently derived expressions for these derivatives with respect to the Cartesian coordinates of the center of the balls [55, 56]. We revisit this problem here and propose new expressions for the derivatives with respect to the distances between the center of these balls; these distances represent internal coordinates for the system that are rotationally invariant.

Derivatives with respect to internal distances: The volume of a union of balls and area of its boundary are fully characterized by the simplified, angle-weighted inclusion-exclusion equations 5 and 4, respectively. In appendix A, we show that all terms included in these two formulas can be expressed as functions of the radii of the balls and the distances between their centers. We compute the derivatives of the volume and area with respect to these distances

algebraically. Note that the derivatives with respect to the distance r_{ab} between the centers z_a and z_b of the two balls B_a and B_b is non zero if and only if the edge $z_a z_b$ belongs to the dual complex. We get:

Weighted Area Derivative Theorem

$$\frac{\delta \mathcal{A}_w}{\delta r_{ab}} = \sum_{\tau i \in K} \alpha_i \mathcal{A}_i \frac{\delta \gamma_i}{\delta r_{ab}} - \gamma_{ab} \alpha_a \frac{\delta \mathcal{A}_{a;b}}{\delta r_{ab}} - \gamma_{ab} \alpha_b \frac{\delta \mathcal{A}_{b;a}}{\delta r_{ab}} - \sum_{\tau_{ij} \in K} \frac{\delta \gamma_{ij}}{\delta r_{ab}} (\alpha_i \mathcal{A}_{i;j} + \alpha_j \mathcal{A}_{j;i}) + \sum_{i| \tau_{abi} \in K} \gamma_{abi} \left(\alpha_a \frac{\delta \mathcal{A}_{a;bi}}{\delta r_{ab}} + \alpha_b \frac{\delta \mathcal{A}_{b;ai}}{\delta r_{ab}} + \alpha_i \frac{\delta \mathcal{A}_{i;ab}}{\delta r_{ab}} \right) \quad (11)$$

and

Weighted Volume Derivative Theorem

$$\begin{aligned} \frac{\delta \mathcal{V}_w}{\delta r_{ab}} = & \sum_{\tau i \in K} \alpha_i \mathcal{V}_i \frac{\delta \gamma_i}{\delta r_{ab}} - \gamma_{ab} \alpha_a \frac{\delta \mathcal{V}_{a;b}}{\delta r_{ab}} - \gamma_{ab} \alpha_b \frac{\delta \mathcal{V}_{b;a}}{\delta r_{ab}} - \sum_{\tau_{ij} \in K} \frac{\delta \gamma_{ij}}{\delta r_{ab}} (\alpha_i \mathcal{V}_{i;j} + \alpha_j \mathcal{V}_{j;i}) \\ & + \sum_{i| \tau_{abi} \in K} \gamma_{abi} \left(\alpha_a \frac{\delta \mathcal{V}_{a;bi}}{\delta r_{ab}} + \alpha_b \frac{\delta \mathcal{V}_{b;ai}}{\delta r_{ab}} + \alpha_i \frac{\delta \mathcal{V}_{i;ab}}{\delta r_{ab}} \right) \\ & + \sum_{i,j| \tau_{abij} \in K} \left(\alpha_a \frac{\delta \text{vol}(F_a)}{\delta r_{ab}} + \alpha_b \frac{\delta \text{vol}(F_b)}{\delta r_{ab}} + \alpha_i \frac{\delta \text{vol}(F_i)}{\delta r_{ab}} + \alpha_j \frac{\delta \text{vol}(F_j)}{\delta r_{ab}} \right) \end{aligned} \quad (12)$$

In the specific case that all weights are equal to 1, the derivative of the volume is:

Volume Derivative Theorem

$$\frac{\delta \mathcal{V}}{\delta r_{ab}} = \sum_{\tau i \in K} \mathcal{V}_i \frac{\delta \gamma_i}{\delta r_{ab}} - \gamma_{ab} \frac{\delta \mathcal{V}_{a;b}}{\delta r_{ab}} - \gamma_{ab} \frac{\delta \mathcal{V}_{b;a}}{\delta r_{ab}} - \sum_{\tau_{ij} \in K} \frac{\delta \gamma_{ij}}{\delta r_{ab}} (\mathcal{V}_{i;j} + \mathcal{V}_{j;i}) + \sum_{i| \tau_{abi} \in K} \gamma_{abi} \left(\frac{\delta \mathcal{V}_{a;bi}}{\delta r_{ab}} + \frac{\delta \mathcal{V}_{b;ai}}{\delta r_{ab}} + \frac{\delta \mathcal{V}_{i;ab}}{\delta r_{ab}} \right) + \sum_{i,j| \tau_{abij} \in K} \frac{\delta \text{vol}(\tau_{abij})}{\delta r_{ab}} \quad (13)$$

Note that there are no terms involving the derivatives of γ_{abi} as those are independent of distances.

Formulas for the derivatives of the different terms $\mathcal{A}_{i;j}$, $\mathcal{A}_{i;jk}$, $\mathcal{V}_{i;j}$, $\mathcal{V}_{i;jk}$ and $\text{vol}(F_i)$ are straightforward from their analytical expressions (see appendix A). The angular coefficient γ_i of a vertex z_i is computed over all tetrahedra of K that contain i . if z_i is such that it belongs to at least one tetrahedron of K that also contains z_a and z_b , then:

$$\frac{\delta \gamma_i}{\delta r_{ab}} = -\frac{1}{4\pi} \sum_{j| \tau_{ijab} \in K} \left(\frac{\delta \varphi_{ij}}{\delta r_{ab}} + \frac{\delta \varphi_{ia}}{\delta r_{ab}} + \frac{\delta \varphi_{ib}}{\delta r_{ab}} \right)$$

In all other cases, $\frac{\delta \gamma_i}{\delta r_{ab}} = 0$. Similarly,

$$\frac{\delta \gamma_{ij}}{\delta r_{ab}} = -\frac{1}{2\pi} \frac{\delta \varphi_{ij}}{\delta r_{ab}}$$

if $\tau_{ijab} \in K$, and 0 otherwise.

The derivatives of the dihedral angles φ_{ij} of a tetrahedron with respect to its edge lengths are given in appendix B. Finally, the volume derivative formula (13) also includes the derivatives of the volume of a tetrahedron with respect to its edge lengths, whose expressions are also given in appendix B.

Derivatives with respect to Cartesian coordinates: Once the derivatives with respect to internal coordinates are available, derivatives with respect to Cartesian coordinates are easily computed using the chain rule:

Cartesian Derivative Theorem The gradients $\mathbf{a}$ and $\mathbf{v} \in \mathbb{R}^{3n}$ of the area and volume derivatives are

$$\begin{bmatrix} \mathbf{a}_{3i+1} \\ \mathbf{a}_{3i+2} \\ \mathbf{a}_{3i+3} \end{bmatrix} = \sum_{j \neq i} \frac{\delta \mathcal{A}}{\delta r_{ij}} \mathbf{u}_{ij}$$

$$\begin{bmatrix} \mathbf{v}_{3i+1} \\ \mathbf{v}_{3i+2} \\ \mathbf{v}_{3i+3} \end{bmatrix} = \sum_{j \neq i} \frac{\delta V}{\delta r_{ij}} \mathbf{u}_{ij}$$

where $\mathbf{u}_{ij} = (\mathbf{z}_i - \mathbf{z}_j)/r_{ij}$ is the unit vector in the direction of the edge $\mathbf{z}_i\mathbf{z}_j$.

2.2 Voids and Pockets

A full description of how to detect and measure pockets in a union of balls based on the alpha shape theory is available in [51]. Briefly, the concept of pockets is ultimately connected to the notion of a continuous flow field defined on the Delaunay triangulation of these balls. Let T be the set of tetrahedra in the Delaunay triangulation and $T = T \cup \tau_\infty$ where τ_∞ is a dummy element representing the complement of the triangulation in $\mathbb{R}^3$. The flow relation ' $\succ$ ' with $\tau \succ \sigma$ is defined by:

- i. τ and σ share a common triangle Δ , and
- ii. The interior of τ and the orthogonal center $\mathbf{z}_\tau$ of τ lie on different sides of the plane defined by Δ .

where the orthogonal center $\mathbf{z}_\tau$ is the center of the smallest ball that is orthogonal to all four balls whose centers are the vertices of τ .

If $\tau \succ \sigma$, τ is a predecessor of σ and σ is a successor of τ . $\sigma \in T$ is a sink if it has no successors; in other words, a tetrahedron is a sink if and only if it contains its orthogonal center. Sinks are important since they are responsible for the formation of voids: if H is a void of the union of balls then at least one tetrahedron in H is a sink.

By definition, pockets consist of the Delaunay tetrahedra that do not belong to the dual complex K and are not ancestors of τ_∞ . The only type of pockets without connection to the outside are the voids. All other pockets connect to the outside at one or more places, called *mouth*. Figure 2 illustrates these concepts on a simple two-dimensional example.

The surface area and volume of a pocket are easily computed by first identifying their tetrahedra and their faces that belong to the dual complex followed by the application of simplified inclusion-exclusion formula similar to those used for measuring the dual complex (see [57] for details).

3 Algorithm & Implementation

AlphaVol is our original software package that implemented the Alpha Shape theory for measuring biomolecules [57]; its origins lie in the Alpha Shape package [62]. AlphaVol takes as input a set of balls in $\mathbb{R}^3$, each specified by the coordinates of its center and the radius. In the case of biomolecules, this set is extracted from the corresponding PDB file using one of several standard sets of van der Waals radii. The computation is performed through four successive tasks:

- Step 1. Construct the Delaunay triangulation.
- Step 2. Extract the dual complex.
- Step 3. Measure the union using inclusion-exclusion.
- Step 4. Detecting and measuring the pockets

AlphaVol uses a standard algorithm from computational geometry for each of these tasks: the incremental flipping algorithm from Edelsbrunner and Shah [63] for computing the Delaunay, an algorithm based on the primitives described by Edelsbrunner [64] for computing the dual complex, our own algorithm for implementing the inclusion-exclusion formula [55, 56] and the algorithm of Edelsbrunner, Facello and Liang [51] for computing pockets. We had made a few small modifications to these algorithms as our interests are mostly measuring biomolecules. For example, we only compute the dual complex and not the full filtration of the Delaunay complex.

AlphaVol showed good performances on a set of small to medium-sized proteins [57]; table 1 illustrates however that the algorithms we have implemented fail, or at least become very slow for very large system. This is especially true for computing the Delaunay (the capsid corresponding to 2dum is ten times larger than the capsid for 1ihm, however it takes more than 100 times longer to compute its Delaunay triangulation which is not in par with an expected $O(n \log(n))$ time complexity) and even more for detecting and measuring pockets. We have written a new version of the AlphaVol software [57] in which the original algorithms have been either modified or fully rewritten to alleviate these severe drawbacks. In the following we describe these modified algorithms and their implementations. The new program is called UnionBall.

We compare the original and modified algorithms as implemented in the corresponding programs AlphaVol and UnionBall for all four steps defined above on a dataset of 285 virus capsids varying in size from sixty thousand atoms to sixteen million atoms; the structures for these capsids were downloaded from the web enabled relational database VIPERdb2 [65]. Note that these capsids are highly symmetric (icosahedral) and repetitive; we do not however make use of these symmetries. For all capsids, we compute the accessible surface area, the corresponding solvent-excluded volume, we detect all pockets, and we compute their volume and surface area. Figure 3 illustrates this process on the Sindbis virus.

3.1 Improved Delaunay computations for large molecular systems

Our implementation of the Delaunay triangulation in the original program AlphaVol was based on the randomized incremental algorithm described in [63]. In this algorithm, the triangulation is constructed incrementally, by adding one sphere at a time. Before starting the construction, the spheres are re-indexed such that $S_1, S_2, \dots, S_n$ is a random permutation of the spheres as they appear in the input file. Four dummy additional spheres with their centers at infinity are added so that all input spheres are contained in the tetrahedron they define. Let D_i be the Delaunay triangulation of the four spheres at infinity together with $S_1, S_2, \dots, S_i$. The algorithm proceeds by iterating three steps:

```

for  $i = 1$  to  $n$  do
1   find tetrahedron  $\tau \in D_{i-1}$  that contains  $z_i$ ;
2   add  $z_i$  to decompose  $\tau$  into four tetrahedra;
3   flip locally non-Delaunay triangles attached to  $z_i$ 
endfor.

```

The randomization preprocessing in this algorithm guarantees an expected theoretical running time of $O(n \log(n) + n^2)$ in the worst case [63]. In practice however, a very different behavior is observed for very large dataset as illustrated in figure 4. This is unfortunately a known problem related to memory access on a computer observed for large dataset. Inherent to their nature, randomized algorithms access the data structures they maintain randomly, and random access works poorly with memory hierarchies available on modern computers. Virtual memory operating systems cache recently used data in memory, under the assumption that they are more likely to be used again soon. This assumption is violated by randomized algorithms who consequently perform poorly as the data structure exceeds the cache size.

A simple solution is to insert points in an order which improves locality. Amenta, Choi and Rote [66] developed such a scheme while at the same time maintaining enough randomness so that the algorithm remains theoretically optimal. Their Biased Randomized Insertion Order (BRIO) method was shown to significantly improve performance. Later, Liu and Snoeyink [67] proposed a different method for ordering the points based on the space-filling curve. In their methods, all input points are placed into a 3D grids of N^3 bins which are then visited in a Hilbert curve order; this method was found to speed up step 1 (point location). Both BRIO and this method require preprocessing of the data that comes with a computational cost.

Interestingly, the order in which data points are stored in a PDB file is inherently local. In most cases, two consecutive atoms either belong to the same amino acids or to two sequential amino acids that are in contact. Breaks occur for missing data and/or between chains in the case of a multimeric structures. These breaks may lead to non locality; they are however the exception and are not expected to play a significant role. We tested the effect of using the locality provided by the input PDB file by simply removing the randomization step in the algorithm described above. As illustrated in figure 4, this resulted in a significant improvement in performance. Removing the randomization leads to an observed linear dependence of the computing time with respect to the number of weighted spheres considered. We implemented this modification in UnionBall. Note that our initial attempts to implement BRIO and the Hilbert curve ordering did not lead to improved performances (data not shown).

3.2 Improved dual complex construction

Changing the Delaunay algorithm leads to a different ordering of the tetrahedra in the geometric data structure that stores the triangulation; this by itself is expected to result in faster construction of the dual complex. There is however another step that can be improved.

Given the Delaunay triangulation D of the input spheres, we construct the dual complex $K \subseteq D$ by labeling the Delaunay simplices. Specifically, for each simplex $\tau \in D$ there is a threshold α_τ such that $\tau \in K$ if and only if $\alpha_\tau^2 \leq 0$. To label the Delaunay simplices we therefore need to decide the signs of their square thresholds. This test can be expressed in terms of the signs of the determinants of small matrices whose entries are center coordinates and square radii of the input spheres. Detailed expressions for these tests can be found in [62, 64]. An important ingredient in this context is the treatment of singularities. Inexact versions of the numerical tests are vulnerable to roundoff errors and can lead to wrong output. Following work in computational geometry [68], we implemented these tests using a so-called floating-point filter that first evaluates the tests approximately, using floating-points arithmetic, and if the results

cannot be trusted, switches to exact arithmetic. The difficult part in implementing such a filter comes in the definition of "trust". Let us consider for example the determinant:

$$D = \begin{vmatrix} x_i & y_i & z_i & x_i^2 + y_i^2 + z_i^2 - r_i^2 \\ x_j & y_j & z_j & x_j^2 + y_j^2 + z_j^2 - r_j^2 \\ x_k & y_k & z_k & x_k^2 + y_k^2 + z_k^2 - r_k^2 \\ x_l & y_l & z_l & x_l^2 + y_l^2 + z_l^2 - r_l^2 \end{vmatrix}$$

that is needed for computing α_τ^2 for the tetrahedron τ defined by the four vertices z_i, z_j, z_k , and z_l . An upper bound on the error in computing D is given by:

$$\varepsilon(D) = C * C_{max}^5 * \varepsilon_0$$

where C is a constant that depends on the number of terms in the expansion of D , C_{max} is the maximum absolute value of all coordinates of the four vertices, and ε_0 is the IEEE machine precision equal to 2^{-53} . For large molecules, C_{max} can be quite large (in the order of several thousands) leading to large values for $\varepsilon(D)$. This value however is unnecessarily large for the predicates involved in the construction of the dual complex. A tetrahedron belongs to the dual complex if the four spheres it represents have a common intersection. As such, the distance between any two of these centers cannot be larger than the sum of the radii of the two spheres, typically lower than 10 in the case of molecule, i.e. much smaller than the absolute coordinates of the point. We can use this fact by first centering the simplex under consideration on its orthocenter; the corresponding C_{max} value is consequently much smaller, leading to smaller $\varepsilon(D)$ and consequently to a smaller number of switches to exact arithmetics. We have implemented this modification in UnionBall.

Figure 5 compares the computing times required to construct the dual complex for all virus capsids in our dataset by AlphaVol and UnionBall; the improvement is not as drastic as for computing the Delaunay triangulation but still significant.

Weighted surface areas, volumes, and their derivatives—In UnionBall, we compute the weight surface area and volume of the union of balls using the short weighted formulas given by equations 4 and 5 as well as the formulas for computing the intersections of one, two and three balls given in appendix A. Note that these formulas only depend on the distances between the centers of the balls, and not on their Cartesian coordinates. All the distances can be precomputed, resulting in a significant speedup. Figure 6 compares the computing times required to measure the dual complex for all virus capsids in our dataset by AlphaVol and UnionBall; we do believe that most of the improvement comes from the better ordering of the tetrahedron resulting from the modified algorithm used for computing the Delaunay triangulation.

The derivatives of the weighted surface area and volume are computed using equations 11, 12, and 14.

Detecting and measuring voids—As shown in table 1, the detection of voids and cavities as implemented in AphaVol is the most inefficient step in characterizing the geometry of a very large union of balls, with a near N^2 dependence with respect to the number of balls. The corresponding algorithm was originally designed for generic union of balls. It starts from the master list of tetrahedra in the Delaunay complex, stored in the order in which they belong to the alpha complex and proceeds in two steps (see [51] for more details). Firstly, it computes a depth for each tetrahedron which is the index of its largest successor based on the discrete flow

relationship. Secondly, pockets are constructed as sets of tetrahedra represented by a union-find data structure. The initial list of pockets is empty. The process then scans the tetrahedra that do not belong to the dual complex as they appear in the master list; when it reaches the tetrahedron with index j , all tetrahedra with depth j are added to the union-find structure, each as an individual pocket. When a tetrahedron is added however, the algorithm checks its four direct neighbours; if one of these belong to an existing pocket and the face between the two tetrahedra is not in the dual complex, the two corresponding pockets are joined. The algorithm stops when all tetrahedra have been processed. Each set of tetrahedra in the final union-find structure is deemed a pocket, with the exception of the set containing the dummy tetrahedron τ_{∞} which represents the outside.

This algorithm is theoretically optimal; in practice however, it suffers from the same problem of lack of locality when accessing data in memory, leading to significant thrashing. To circumvent this problem, we propose a different approach that is geared towards improved locality. Similar to the original approach, we compute the depth of each tetrahedron in the Delaunay complex; this step is fast as it is local by nature. From this knowledge, we can isolate the tetrahedra that flows to the outside. Each tetrahedron in the Delaunay complex is then assigned a flag, *visited*, initially set to one if it belongs to the dual complex or flows to outside and zero otherwise. The algorithm then proceeds as follows:

```

for all tetrahedra  $\sigma$  in  $\text{del}(B)$  do
  if  $\text{visited}(\sigma) = 0$  then
    Define new pocket  $P = \{\sigma\}$ ; Define  $L(P) = \{\sigma\}$ ;  $\text{visited}(\sigma) = 1$ ;
    while  $|L(P)| \neq 0$  do
       $\tau = \text{pop}(L(P))$ 
      for all  $\varphi \in N(\tau)$  with  $\text{visited}(\varphi)=0$  do
        let  $T$  be the triangle shared by  $\tau$  and  $\varphi$ 
        if  $T \in K$  then
           $P = P \cup \{\varphi\}$ ;  $L(P) = L(P) \cup \{\varphi\}$ ;  $\text{visited}(\varphi) = 1$ 
        endif
      endfor
    endwhile
  endif
endfor.

```

where $N(\tau)$ is the list of (up to) four neighbors of the tetrahedron τ . As written, this algorithm detects the pockets in the union of ball; it can easily be extended to compute their volume and surface area (as a tetrahedron is added to a pocket we also compute its contribution to the geometric measures).

The main element of this algorithm is the list $L(P)$: it enforces spatial locality, which to a first approximation matches with locality in the list of tetrahedron, resulting in a much better usage of cache; figure 7 illustrates this improvement.

Characterizing the geometry of large biomolecules—UnionBall incorporates all the modifications presented above. With this new program, it takes 18.4 s and 136.5 seconds to fully characterize the two viruses 1ihm and 2dum, respectively; these numbers are significantly better than the corresponding computing times for AlphaVol, namely 2903s for 1ihm and 214553 seconds for 2dum (see table 1).

The plots in Figure 8 shows the running times per point in millisecond for all four steps performed by UnionBall as a function of the virus size. All four steps show nearly constant

behavior over the whole range of virus sizes. Constructing the regular triangulation is the slowest step with an average running time of 13 μ -seconds per points. This running time compares favorably with those reported for five popular codes that compute 3D Delaunay tessalations (figures 3 and 4 in Liu and Snoeyink [67]), even after correcting for the difference in processor speed. Among these five codes, Tess3 is the fastest, with an average running time of 20 μ -second per point, which would correspond to 10 μ -second on the processor we have used; it should be mentioned however that Tess3 performs all calculations using floating points, resulting in some topological mistakes in a few rare cases. Filtering the regular triangulation to construct the dual complex, computing the accessible surface area and volume of the virus, and detecting and measuring the pockets require on average 3 μ -seconds, 7 μ -seconds and 5 μ -seconds, respectively.

4 Conclusion

The alpha shape theory provides an accurate and robust method for computing the geometric measures of a biomolecule. Among these measures, surface area and volume are used to quantify the interactions between such a biomolecule, and the water surrounding it, in implicit solvent models. In addition, the detection of pockets within a biomolecule and the determination of their sizes serve as a starting point for predictive studies of biomolecule-ligand interactions. Several implementations of the alpha shape theory exist, including our own, AlphaVol [57]; these implementations have mostly been tested on globular proteins or medium-size nucleic acids. In the last few years however, spectacular advances in structural biology have produces an abundance of data on large macromolecular complexes, such as the RNA polymerase transcription complexes [69], the ribosome complexes [70, 71], and full size viruses [65] that contain several millions of atoms. Modeling these large systems is as important as modeling smaller proteins or nucleic acids. We have shown in this paper that the standard implementations of the alpha shape theory fail on such large systems, or at least become impractical as their running times are then unrealistically large. We have shown also that these difficulties are not theoretical; rather, they are related to the architecture of current computers that rely on the use of cache memory based on the principle of locality to speed up calculation. By rewriting the algorithms that implement the different steps of the alpha shape theory such that we improve locality, we have shown that we can remediate these cache problems; the corresponding code, UnionBall has an apparent $O(n)$ behavior over a large range of values of n (up to tens of millions), where n is the number of atoms in the macromolecule. The two critical steps for which the largest improvements are observed are the construction of the Delaunay tessalation and the detection and measurements of the pockets.

The key to improving the construction of the Delaunay complex for large sets of points was to recognize that a fully randomized algorithm is impractical as it brakes the locality principle. We have remediated this problem by biasing the order in which the points are inserted, using the simplest possible scheme, i.e. the order in which the points appear in the PDB structure file. Since biomolecules are basically long chains of monomers that are stored sequentially as they appear along the chain, this natural ordering ensures locality. Obviously, this "trick" is specific to biomolecular structural data and would not apply to generic data sets. The concept of biasing the order of insertion for constructing the Delaunay complex is however more general and has been implemented before [66, 67].

Further improvements might still be possible if we take into account the nature of the data even more. For example, all the viral capsids that were used as a test set in this study have a large empty cavity in their center (see figure 3). In the process of building the Delaunay complex for such a capsid, the algorithm will cover this cavity with a large number of complicated, elongated tetrahedra. It might be possible to generate a more regular tessalation of this cavity

by introducing dummy points in the cavity, following the idea of adding points in a tetrahedral mesh to improve its quality [72]. We are currently exploring this idea.

We conclude this paper by mentioning that UnionBall is available as OpenSource software by contacting P.K.

Acknowledgments

This work derives from a long standing collaboration between P.K. and Prof. Herbert Edelsbrunner; we thank him for his mentorship and guidance. P.K. acknowledges current support from the NIH under contract GM080399.

References

1. Eisenberg D, McLachlan AD. *Nature (London)*. 1986; 319:199–203. [PubMed: 3945310]
2. Ooi T, Oobatake M, Nemethy G, Scheraga HA. *Proc Natl Acad Sci (USA)*. 1987; 84:3086–3090. [PubMed: 3472198]
3. Liang J, Edelsbrunner H, Fu P, Sudhakar PV, Subramaniam S. *Proteins: Struct Func Genet*. 1998; 33:1–17.
4. Liang J, Edelsbrunner H, Fu P, Sudhakar PV, Subramaniam S. *Proteins: Struct Func Genet*. 1998; 33:18–29.
5. Lee B, Richards FM. *J Mol Biol*. 1971; 55:379–400. [PubMed: 5551392]
6. Wood RH, Thompson PT. *Proc Natl Acad Sci (USA)*. 1990; 87:946–949. [PubMed: 11607062]
7. Tunon I, Silla E, Pascual-Ahuir JL. *Protein Eng*. 1992; 5:715–716. [PubMed: 1287651]
8. Simonson T, Brünger AT. *J Phys Chem*. 1994; 98:4683–4694.
9. Lum K, Chandler D, Weeks JD. *J Phys Chem B*. 1999; 103:4570–4577.
10. Wagoner J, Baker N. *Proc Natl Acad Sci (USA)*. 2006; 103:8331–8336. [PubMed: 16709675]
11. Shrake A, Rupley JA. *J Mol Biol*. 1973; 79:351–371. [PubMed: 4760134]
12. Legrand SM, Merz KM. *J Comp Chem*. 1993; 14:349–352.
13. Wang H, Levinthal C. *J Comp Chem*. 1991; 12:868–871.
14. Futamura N, Alura S, Ranjan D, Hariharan B. *IEEE Trans Parallel Dist Syst*. 2004; 13:544–555.
15. Rowlinson JS. *Mol Phys*. 1963; 6:517–524.
16. Pavani R, Ranghino G. *Computers and Chemistry*. 1982; 6:133–135.
17. Gavezzotti A. *J Am Chem Soc*. 1983; 105:5220–5225.
18. Till M, Ullmann GM. *J Mol Model*. 2010; 16:419–429. [PubMed: 19626353]
19. Wodak SJ, Janin J. *Proc Natl Acad Sci (USA)*. 1980; 77:1736–1740. [PubMed: 16592793]
20. Hasel W, Hendrikson TF, Still WC. *Tetrahed Comp Method*. 1988; 1:103–106.
21. Cavallo LJK, Fraternali F. *Nucl Acids Res*. 2003; 31:3364–3366. [PubMed: 12824328]
22. Street AG, Mayo SL. *Folding & Design*. 1998; 3:253–258. [PubMed: 9710572]
23. Weiser J, Shenkin PS, Still WC. *J Comp Chem*. 1999; 20:217–230.
24. Dynerman D, Butzlaff E, Mitchell J. *J Comput Biol*. 2009; 16:523–537. [PubMed: 19361325]
25. Richmond TJ. *J Mol Biol*. 1984; 178:63–89. [PubMed: 6548264]
26. Connolly ML. *J Am Chem Soc*. 1985; 107:1118–1124.
27. Dodd LR, Theodorou DN. *Mol Phys*. 1991; 72:1313–45.
28. Petitjean M. *J Comp Chem*. 1994; 15:507–523.
29. Irida M. *Comp Phys Comm*. 1996; 98:317–338.
30. Gibson KD, Scheraga HA. *Mol Phys*. 1987; 62:1247–1265.
31. Kratky KW. *J Phys A: Math Gen*. 1978; 11:1017–1024.
32. Edelsbrunner H. *Discrete Comput Geom*. 1995; 13:415–440.
33. Kundrot CE, Ponder JW, Richards FM. *J Comp Chem*. 1991; 12:402–409.
34. Gogonea V, Osawa E. *J Mol Struct (Theochem)*. 1994; 311:305–324.
35. Gogonea V, Osawa E. *J Comp Chem*. 1995; 16:817–842.

36. Cossi M, Mennucci B, Cammi R. *J Comp Chem*. 1996; 17:57–73.
37. Perrot G, Cheng B, Gibson KD, Vila J, Palmer KA, Nayeem A, et al. *J Comp Chem*. 1992; 13:1–11.
38. Sridharan S, Nicholls A, Sharp KA. *J Comp Chem*. 1994; 16:1038–1044.
39. Wawak RJ, Gibson KD, Scheraga HA. *J Math Chem*. 1994; 15:207–232.
40. Grant JA, Pickup BT. *J Phys Chem*. 1995; 99:3503–3510.
41. Weiser J, Shenkin PS, Still WC. *J Comp Chem*. 1999; 20:688–703.
42. Edelsbrunner H. *Discrete Comput Geom*. 1999; 21:87–115.
43. Levitt D, Banaszak L. *J Mol Graph*. 1992; 10:229–234. [PubMed: 1476996]
44. Hendlich M, Rippmann F, Barnickel G. *J Mol Graph Model*. 1997; 15:359–363. [PubMed: 9704298]
45. Venkatachalam C, Jiang X, Oldfield T, Waldman M. *J Mol Graph Model*. 2003; 21:289–307. [PubMed: 12479928]
46. Weisel M, Proschak E, Schneider G. *Chem Central J*. 2007; 1:7.
47. Laskowski R. *J Mol Graph*. 1995; 13:323–330. [PubMed: 8603061]
48. Brady G, Stouten P. *J Comput Aided Mol Des*. 2000; 14:383–401. [PubMed: 10815774]
49. Kawabata T, Go N. *Proteins: Struct Func Genet*. 2007; 68:516–529.
50. Yu J, Zhou Y, Tanaka I, Yao M. *Bioinformatics*. 2010; 26:46–52. [PubMed: 19846440]
51. Edelsbrunner H, Facello MA, Liang J. *Discrete Appl Math*. 1998; 88:83–102.
52. Liang J, Edelsbrunner H, Woodward C. *Prot Sci*. 1998; 7:1884–1897.
53. Yaffe E, Fishelovitch D, Wolfson H, Halperin D, Nussinov R. *Nucl Acids Res*. 2008; 36:W210–W215. [PubMed: 18448468]
54. Busa J, Hayryan S, Hu C-K, Skrivaneck J, Wu M-C. *J Comp Chem*. 2009; 30:346–357. [PubMed: 18629810]
55. Edelsbrunner H, Koehl P. *Proc Natl Acad Sci (USA)*. 2003; 100:2203–2208. [PubMed: 12601153]
56. Bryant R, Edelsbrunner H, Koehl P, Levitt M. *Discrete Comput Geom*. 2004
57. Edelsbrunner H, Koehl P. *Discrete and Computational Geometry (MSRI Publications)*. 2005; 52:243–275.
58. Dundas J, Ouyang Z, Tseng J, Binkowski A, Turpaz Y, Liang J. *Nucl Acids Res*. 2006; 34:W116–W118. [PubMed: 16844972]
59. Naiman D, Wynn H. *Annals of Stat*. 1992:43–76.
60. Edelsbrunner H, Mücke EP. *ACM Trans Graphics*. 1990; 9:66–104.
61. Attali D, Edelsbrunner H. *Discrete Comput Geom*. 2007; 37:59–77.
62. Edelsbrunner H, Mücke EP. *ACM Trans Graphics*. 1994; 13:43–72.
63. Edelsbrunner H, Shah NR. *Algorithmica*. 1996; 15:223–241.
64. Edelsbrunner, H. *Weighted alpha shapes Technical Report UIUC-CS-R-92-1760. Comput. Sci. Dept., Univ. Illinois; Urbana, Illinois: 1992.*
65. Carrillo-Tripp M, Shepherd C, Borelli I, Venkataram S, Lander G, Natarajan P, Johnson J III, CB, Reddy V. *Nucl Acids Res*. 2009; 37:D436–D442. [PubMed: 18981051]
66. Amenta, N.; Choi, S.; Rote, G. *Proc. 19th ACM Sympos. Comput. Geom*; 2003. p. 211–219.
67. Liu Y, Snoeyink J. *Discrete and Computational Geometry (MSRI Publications)*. 2005; 52:439–458.
68. Fortune S, VanWyk CJ. *ACM Trans Graph*. 1996; 15:223–248.
69. Cramer P, Bushnell DA, Kornberg RD. *Science*. 2001; 292:1863–1876. [PubMed: 11313498]
70. Wimberly BT, Brodersen DE, Clemons WM Jr, Morgan-Warren RJ, Carter AP, Vornrhein C, et al. *Nature (London)*. 2000; 407:327–39. [PubMed: 11014182]
71. Ban N, Nissen P, Hansen J, Moore PB, Steitz TA. *Science*. 2002; 289:905–20. [PubMed: 10937989]
72. Shewchuk, J. *Proc. 14th Ann. Sympos. Comput. Geom*; 1998. p. 86–95.
73. Gibson KD, Scheraga HA. *Mol Phys*. 1988; 64:641–644.
74. Edelsbrunner, H.; Fu, P. *Measuring space filling diagrams and voids Technical Report UIUC-BI-MB-94-01. Beckman Inst., Univ. Illinois; Urbana, Illinois: 1994.*
75. Lee J. *J Korea Soc Math Educ Ser B: Pure Appl Math*. 1997; 4:1–6.

76. Yang, L.; Zhang, J. Metric equations in geometry and their applications Technical Report IC/89/281. International Center for Theoretical Physics; Trieste, Italy: 1989.
77. Yang, L.; Zeng, Z. Proc ADG2006, LNAI. Botana, F.; Recio, T., editors. Vol. 4869. 2007. p. 203-211.
78. Zhang W, Mukhopadhyay S, Pletnev S, Baker T, Kuhn R, Rossmann M. J Virol. 2002; 76:11645–11658. [PubMed: 12388725]

Appendix A: Measuring the intersections of two, three and four balls

Several formulas have been presented for the volume and surface areas of the intersection of two, three and four spheres with unequal radii (see for example [30, 73, 74]). Here we propose new geometric derivations of these formulas that satisfy a specific constraint, namely we need expressions for the intersections that only depend of the radii of the spheres and the distance between their centers.

Notation

We consider up to four balls B_i, B_j, B_k and B_l whose boundaries are the spheres S_i, S_j, S_k and S_l , respectively. Let z_i and r_i be the center and radius of ball B_i and let r_{ij} be the distance between z_i and z_j . The intersection between the two balls B_i and B_j is the union of two caps C_{ij} and C_{ji} , illustrated in red and blue respectively in figure A.1. These two caps are connected at the level of the plane that separates the Voronoi cells of S_i and S_j ; this plane cuts the two spheres in a circle with center $y_{i,j}$ and radius $r_{i,j}$. We also define the height of spherical cap C_{ij} as $h_{i,j}$. It can easily be shown that:

$$h_{i,j} = r_i - \frac{r_{ij}}{2} - \frac{r_i^2 - r_j^2}{2r_{ij}} \quad (\text{A.1})$$

$$r_{i,j} = \sqrt{2r_i h_{i,j} - h_{i,j}^2} \quad (\text{A.2})$$

As above, $\mathcal{A}_i$ is the surface area of the sphere S_i ; $\mathcal{A}_{i,j}$, $\mathcal{A}_{i,j,k}$ and $\mathcal{A}_{i,j,k,l}$ are the contributions of S_i to the surface areas of the intersections of S_i and S_j , of S_i, S_j and S_k , and of S_i, S_j, S_k and S_l , respectively:

$$\mathcal{A}_{i,j} = \text{area}(C_{i,j}) \quad \mathcal{A}_{i,j,k} = \text{area}(C_{i,j} \cap C_{i,k}) \quad \mathcal{A}_{i,j,k,l} = \text{area}(C_{i,j} \cap C_{i,k} \cap C_{i,l})$$

Similar expressions are used for volumes.

Intersection of two balls

Lemma 1

The intersection between two balls is illustrated in figure A.1. We have:

$$\mathcal{A}_{i,j} = 2\pi r_i h_{i,j} \quad (\text{A.3})$$

$$\mathcal{V}_{i,j} = \frac{1}{3}\pi h_{i,j}^2 (3r_i - h_{i,j}) \quad (\text{A.4})$$

with $h_{i,j}$ defined in equation A.1.

Proof—Computing the volume and surface area of a spherical cap is a standard textbook problem that has been solved in many forms. Proofs for the formula given above can be found for example at the MathWorld web site (<http://mathworld.wolfram.com/SphericalCap.html>).

Intersection of three balls

The contribution of B_i to the surface area and volume of the intersection of the three balls B_i , B_j and B_k is defined by the intersection of the two caps C_{ij} and C_{ik} , illustrated in red and blue respectively in panel A of figure A.2. The three spheres S_i , S_j and S_k intersect in two points P_{ijk} and P_{ikj} . We consider the tetrahedron T_3 formed by the centers of the three balls and P_{ijk} (see panel B of figure A.2). The faces of T_3 are labeled $z_i z_j z_k$, $z_i z_j P_{ijk}$, $z_i z_k P_{ijk}$ and $z_j z_k P_{ijk}$ with areas s_P , s_k , s_j and s_i , respectively. The areas are computed using Heron's formula (see appendix A). The dihedral angles corresponding to the edges $z_i z_j$ and $z_i z_k$ are denoted as $\theta_{ij,k}$ and $\theta_{ik,j}$, respectively, while ψ_i is the dihedral angle corresponding to the edge $z_i P_{ijk}$.

Lemma 2

The contributions of S_i and B_i to the surface area and volume of the triple intersection are given by:

$$\mathcal{A}_{i,j,k} = 2r_i h_{i,j} \theta_{ij,k} + 2r_i h_{i,k} \theta_{ik,j} - 2r_i^2 (\theta_{ij,k} + \theta_{ik,j} + \psi_i - \pi) \quad (\text{A.5})$$

$$\mathcal{V}_{i,j,k} = \frac{1}{3} r_i \mathcal{A}_{i,j,k} - \frac{1}{3} (r_i - h_{i,j}) (2r_i h_{i,j} - h_{i,j}^2) (\theta_{ij,k} - \sin(\theta_{ij,k}) \cos(\theta_{ij,k})) - \frac{1}{3} (r_i - h_{i,k}) (2r_i h_{i,k} - h_{i,k}^2) (\theta_{ik,j} - \sin(\theta_{ik,j}) \cos(\theta_{ik,j})) \quad (\text{A.6})$$

where the dihedral angles are computed from the edge lengths of the tetrahedron T_3 (see appendix B). Formulas for the contributions of B_j and B_k to the intersection are easily deduced by index permutation on these equations.

Proof—We focus on the geometric proofs of equations A.5 and A.6.

Surface area

Let $z_{i,j}$ and $z_{i,k}$ be the points of intersection of the sphere S_i bounding the ball B_i with the lines $z_i z_j$ and $z_i z_k$, respectively; these two points can be seen as "centers" of the two caps. P_{ijk} and P_{ikj} are the two points that are common to all three spheres. These four points form a spherical quadrangle, with spherical angles $\beta_{ij,k}$, $\beta_{ik,j}$, α_{ijk} and α_{ikj} (see figure A.3). Note that this quadrangle is symmetric with respect to the plane formed by the centers of the three balls which is also the plane passing by the three points z_i , $z_{i,j}$ and $z_{i,k}$. Consequently, $\alpha_{ijk} = \alpha_{ikj}$.

The spherical angle $\beta_{ij,k}$ is the dihedral angle between the plane $\Delta z_i z_{i,j} P_{ijk}$ and the plane $\Delta z_i z_{i,j} P_{ikj}$. Because of the symmetry with respect to the plane containing the three centers, and because $z_{i,j}$ belongs to the line $z_i z_j$, we find:

$$\beta_{ij,k} = 2 * \text{dihed}(\Delta z_i z_j z_k, \Delta z_i z_j P_{ijk}) = 2\theta_{ij,k}$$

Similarly, $\beta_{ik,j} = 2\theta_{ik,j}$ and $\alpha_{ijk} = \varphi_i$.

We compute the surface area Q of this spherical quadrangle in two different ways. First, we

use the formula for the area of a polygon on a sphere ($A = R^2 \left(\sum_{i=1}^n \theta_i - (n-2)\pi \right)$, where R is the radius of the sphere, n the number of vertices in the polygon, and θ_i the internal angle at vertex i):

$$Q = r_i^2 (2\theta_{ij;k} + 2\theta_{ik;j} + 2\alpha_i - 2\pi) \quad (\text{A.7})$$

Second, we observe that the area of the quadrangle can be decomposed as:

- + the area A_1 of the sector of the cap $\mathcal{C}_{ij}$ that is delimited by the two arcs $z_{i;j}P_{ijk}$ and $z_{i;j}P_{ikj}$
- + the area A_2 of the sector of the cap $\mathcal{C}_{ik}$ that is delimited by the two arcs $z_{i;k}P_{ijk}$ and $z_{i;k}P_{ikj}$,
- – the area of the intersection $\mathcal{A}_{i,jk}$ as it appears twice.

Therefore

$$Q = A_1 + A_2 - \mathcal{A}_{i,jk} \quad (\text{A.8})$$

The surface areas A_1 and A_2 are the fraction of the surface areas of the caps $\mathcal{C}_{ij}$ and $\mathcal{C}_{ik}$ covered by the angles $2\theta_{ij}$ and $2\theta_{ik}$

$$\begin{aligned} A_1 &= 2r_i h_{i;j} \theta_{ij;k} \\ A_2 &= 2r_i h_{i;k} \theta_{ik;j} \end{aligned} \quad (\text{A.9})$$

where $h_{i;j}$ and $h_{i;k}$ are the heights of the two caps.

Combining equations A.7, A.8 and A.9, we validate equation A.5.

Volume

To compute the contribution $\mathcal{V}_{ijk}$ of ball B_i to the volume of the intersection of the three balls, we consider the sector of B_i that joins its center z_i to the sphere sector whose surface is $\mathcal{A}_{i,jk}$. The volume V_s of this sector can be computed in two different ways:

- First, the volume V_s of a sector is given as $r_i A/3$, where r_i is the radius of the ball and A is the area of the sector on the surface of the ball:

$$V_s = \frac{1}{3} r_i \mathcal{A}_{i,jk} \quad (\text{A.10})$$

- Second, the same sector can be divided into three parts (see panel A in figure A.4): two fractions of cones (filled in red and blue), and the region $B_{i,jk}$, whose volume is $\mathcal{V}_{ijk}$ (shown in green):

$$V_s = \text{vol}(F_{ij;k}) + \text{vol}(F_{ik;j}) + \mathcal{V}_{i,jk} \quad (\text{A.11})$$

The volume of $F_{ij;k}$ is:

$$\text{vol}(F_{ij,k}) = \frac{1}{3}(r_i - h_{i,j})AS_{ij,k} \quad (\text{A.12})$$

where $AS_{ij,k}$ is the area of the base of $F_{ij,k}$, i.e. the area of the disk of intersection between B_i and B_j covered by the cap $C_{i,k}$ (see panel B in figure A.4). $AS_{ij,k}$ is computed as the difference between the area of the disk covered by $2\theta_{ik}$ and the triangle $\Delta y_{i,j}P_{ijk}P_{ikj}$.

$$AS_{ij,k} = r_{i,j}^2 (\theta_{ij,k} - \sin\theta_{ij,k}\cos\theta_{ij,k}) \quad (\text{A.13})$$

where $r_{i,j}$ is the radius of the disk (see equation A.2). Note that this formula is valid even if the disk sector covers the disk center. Similar expressions are derived for the volume of $F_{ik,j}$.

Combining equations A.10 to A.13, we validate equation A.6.

Intersection of four balls

Let B_i, B_j, B_k and B_l be the four balls with a common intersection. Their centers define a tetrahedron T_4 with faces T_i, T_j, T_k and T_l and corresponding areas s_i, s_j, s_k and s_l , respectively, defined such that $z_a \notin T_a$ for all $a = i, j, k, l$. We denote the dihedral angle of T_4 between the two faces that share the edge $z_i z_j$ as φ_{ij} . We also define the solid angle subtended by the vertex z_i as Ω_i . These angles can be computed from the edge lengths of the tetrahedron (see appendix B). Note that:

$$\Omega_i = \varphi_{ij} + \varphi_{ik} + \varphi_{il} - \pi$$

Lemma 3

The contribution of B_i to the surface area and volume of the intersection of the four balls is defined by the intersection of the three caps $C_{i,j}, C_{i,k}$ and $C_{i,l}$:

$$\mathcal{A}_{i,jkl} = \frac{\Omega_i}{4\pi} \mathcal{A}_i - \frac{\varphi_{ij}}{2\pi} \mathcal{A}_{i,j} - \frac{\varphi_{ik}}{2\pi} \mathcal{A}_{i,k} - \frac{\varphi_{il}}{2\pi} \mathcal{A}_{i,l} + \frac{1}{2} \mathcal{A}_{i,jk} + \frac{1}{2} \mathcal{A}_{i,jl} + \frac{1}{2} \mathcal{A}_{i,kl} \quad (\text{A.14})$$

$$\mathcal{V}_{i,jkl} = \frac{\Omega_i}{4\pi} \mathcal{V}_i - \frac{\varphi_{ij}}{2\pi} \mathcal{V}_{i,j} - \frac{\varphi_{ik}}{2\pi} \mathcal{V}_{i,k} - \frac{\varphi_{il}}{2\pi} \mathcal{V}_{i,l} + \frac{1}{2} \mathcal{V}_{i,jk} + \frac{1}{2} \mathcal{V}_{i,jl} + \frac{1}{2} \mathcal{V}_{i,kl} - \text{vol}(F_i) \quad (\text{A.15})$$

where

$$\begin{aligned} \text{vol}(F_i) = & \frac{1}{6}(r_i - h_{i,j})r_{i,j}^2 \frac{2\cos\theta_{ijk}\cos\theta_{ijl} - (\cos^2\theta_{ijk} + \cos^2\theta_{ijl})\cos\varphi_{ij}}{\sin\varphi_{ij}} \\ & + \frac{1}{6}(r_i - h_{i,k})r_{i,k}^2 \frac{2\cos\theta_{ikj}\cos\theta_{ilk} - (\cos^2\theta_{ikj} + \cos^2\theta_{ilk})\cos\varphi_{ik}}{\sin\varphi_{ik}} \\ & + \frac{1}{6}(r_i - h_{i,l})r_{i,l}^2 \frac{2\cos\theta_{ilj}\cos\theta_{ilk} - (\cos^2\theta_{ilj} + \cos^2\theta_{ilk})\cos\varphi_{il}}{\sin\varphi_{il}} \end{aligned} \quad (\text{A.16})$$

and the angles θ have been defined above for the different intersections of three balls.

Proof—We focus on the geometric proofs of equations A.14 and A.15.

Surface area

Let us consider the spherical triangle $ST = \{z_{i;j}z_{i;k}z_{i;l}\}$ whose vertices are the intersections of the edges $z_{i;j}$, $z_{i;k}$ and $z_{i;l}$ with the sphere S_i . The spherical angle at vertex $z_{i;j}$ is the dihedral angle between the planes defined by $z_{i;j}z_{i;k}$ and $z_{i;j}z_{i;l}$; it is therefore the dihedral angle between the planes $z_{i;j}z_{i;k}$ and $z_{i;j}z_{i;l}$, i.e. φ_{ij} . Similarly, the spherical angles at vertices $z_{i;k}$ and $z_{i;l}$ are φ_{ik} and φ_{il} , respectively. We compute the area A_{tot} of the spherical triangle ST using two approaches:

- Firstly, we use the formula for the area of a spherical polygon:

$$A_{tot} = r_i^2(\varphi_{ij} + \varphi_{ik} + \varphi_{il} - \pi) = \mathcal{A}_i \frac{\Omega_i}{4\pi} \quad (\text{A.17})$$

where $\mathcal{A}_i$ is the surface area of sphere S_i , and Ω_i the solid angle subtended by the vertex z_i of T_4 .

- Secondly, A_{tot} is computed using an inclusion-exclusion formula: it is the sum of the surface areas A_j , A_k and A_l of sectors of the three spherical caps $C_{i;j}$, $C_{i;k}$ and $C_{i;l}$, minus the surface areas A_{jk} , A_{jl} and A_{kl} of the intersections of these sectors, plus the surface area of their triple intersection (see panel B in figure A.5).

The surface areas of the sectors are:

$$A_j = \frac{\varphi_{ij}}{2\pi} \mathcal{A}_{i;j} \quad A_k = \frac{\varphi_{ik}}{2\pi} \mathcal{A}_{i;k} \quad A_l = \frac{\varphi_{il}}{2\pi} \mathcal{A}_{i;l} \quad (\text{A.18})$$

By noticing that the arc circle $z_{i;a}z_{i;b}$ cuts the intersection between the two caps $C_{i;a}$ and $C_{i;b}$ in half for all $a \neq b = j, k, l$, we get:

$$A_{jk} = \frac{1}{2} \mathcal{A}_{i;jk} \quad A_{jl} = \frac{1}{2} \mathcal{A}_{i;jl} \quad A_{kl} = \frac{1}{2} \mathcal{A}_{i;kl} \quad (\text{A.19})$$

Finally, the surface area of the triple intersection is $\mathcal{A}_{i;jkl}$. Therefore,

$$A_{tot} = \frac{\varphi_{ij}}{2\pi} \mathcal{A}_{i;j} + \frac{\varphi_{ik}}{2\pi} \mathcal{A}_{i;k} + \frac{\varphi_{il}}{2\pi} \mathcal{A}_{i;l} - \frac{1}{2} \mathcal{A}_{i;jk} - \frac{1}{2} \mathcal{A}_{i;jl} - \frac{1}{2} \mathcal{A}_{i;kl} + \mathcal{A}_{i;jkl} \quad (\text{A.20})$$

Combining the two equations A.17 and A.20, we validate equation A.14.

Volume

Let us now consider the sector of the ball B_i obtained by joining its center z_i to the spherical triangle $ST = \{z_{i;j}z_{i;k}z_{i;l}\}$ (see panel A of figure A.6). Similar to the computation of the surface area, we compute the volume V_{tot} of this sector using two approaches:

- Firstly, the volume of a ball sector is equal to 1/3 times the radius of the ball times the surface area of the sector:

$$V_{tot} = \frac{1}{3} r_i A_{tot} = \mathcal{V}_i \frac{\Omega_i}{4\pi} \quad (\text{A.21})$$

- Secondly, we divide the sector into two regions: the region F_i delimited by the tetrahedron T_4 and the three Voronoi planes that separates B_i from B_j , B_k and B_l , and the region G_i delimited by these three planes and the sphere S_i (panel A of figure A.6). The volume of G_i is computed using the same inclusion-exclusion formula that was used for the surface area of ST :

$$\text{vol}(G_i) = \frac{\varphi_{ij}}{2\pi} \mathcal{V}_{i,j} + \frac{\varphi_{ik}}{2\pi} \mathcal{V}_{i,k} + \frac{\varphi_{il}}{2\pi} \mathcal{V}_{i,l} - \frac{1}{2} \mathcal{V}_{i,jk} - \frac{1}{2} \mathcal{V}_{i,jl} - \frac{1}{2} \mathcal{V}_{i,kl} + \mathcal{V}_{ijkl} \quad (\text{A.22})$$

therefore,

$$V_{\text{tot}} = \frac{\varphi_{ij}}{2\pi} \mathcal{V}_{i,j} + \frac{\varphi_{ik}}{2\pi} \mathcal{V}_{i,k} + \frac{\varphi_{il}}{2\pi} \mathcal{V}_{i,l} - \frac{1}{2} \mathcal{V}_{i,jk} - \frac{1}{2} \mathcal{V}_{i,jl} - \frac{1}{2} \mathcal{V}_{i,kl} + \mathcal{V}_{ijkl} + \text{vol}(F_i) \quad (\text{A.23})$$

Combining the two equations A.21 and A.23, we validate equation A.15.

The volume of F_i is computed as the sum of the volumes of the three pyramids with apex z_i and bases on the Voronoi planes relative to B_i (see figure A.6):

$$\text{vol}(F_i) = \frac{1}{3} (r_i - h_{i,j}) \text{area}(A_1) + \frac{1}{3} (r_i - h_{i,k}) \text{area}(A_2) + \frac{1}{3} (r_i - h_{i,l}) \text{area}(A_3) \quad (\text{A.24})$$

The surface area of the base A_1 is computed as the difference between the area of the triangles $\Delta y_{i,j} D_l D$ and $\Delta D_k D P_{ijkl}$ (see panel B in figure A.6):

$$\begin{aligned} \text{area}(A_1) &= \frac{1}{2} d_l^2 \tan \varphi_{ij} - \frac{1}{2} \left(\frac{d_l}{\cos \varphi_{ij}} - d_k \right)^2 \frac{1}{\tan \varphi_{ij}} \\ &= \frac{2d_k d_l - (d_k^2 + d_l^2) \cos \varphi_{ij}}{\sin \varphi_{ij}} \end{aligned} \quad (\text{A.25})$$

where $d_k = r_{i,j} \cos \theta_{ij;k}$ and $d_l = r_{i,j} \cos \theta_{ij;l}$ (see figure A.4). Similar equations are derived for the areas of A_2 and A_3 . Combining these equations with equation A.24 validates equation A.16.

Note that:

$$F_i + F_j + F_k + F_l = \text{vol}(T_4) \quad (\text{A.26})$$

Angle weighted inclusion-exclusion formula for union of balls

Equation A.14 implies that the surface area of the intersection of four spheres is a linear combination of the surface areas of the individual spheres and of the intersections of two and three spheres, where the linear coefficients are related to the six dihedral angles of the tetrahedron formed by the centers of the four spheres. The same relationship exists for the volume of the intersection of four balls, with additional terms for the intersection of the tetrahedron formed by the center of the four balls and their Voronoi cells. Replacing the corresponding equations (A.14 and A.15) in the Weighted Volume and Weighted Area Theorems, we get the simplified inclusion-exclusion formulas 4 and 5.

Appendix B: The geometry of a tetrahedron

Let us consider the tetrahedron T defined by the four vertices P_1, P_2, P_3 and P_4 . The four faces of this tetrahedron are $T_1 = \Delta P_2 P_3 P_4$, $T_2 = \Delta P_1 P_3 P_4$, $T_3 = \Delta P_1 P_2 P_4$, and $T_4 = \Delta P_1 P_2 P_3$ and their areas are s_1, s_2, s_3 and s_4 , respectively. We denote the dihedral angle with respect T_i and T_j for $i \neq j = 1, 2, 3, 4$ as θ_{ij} . The edge between P_i and P_j has length l_{ij} , for $i \neq j = 1, 2, 3, 4$.

Surface area and volume

The Cayley-Menger matrix M associated with T is given by:

$$M = \begin{pmatrix} 0 & l_{12}^2 & l_{13}^2 & l_{14}^2 & 1 \\ l_{12}^2 & 0 & l_{23}^2 & l_{24}^2 & 1 \\ l_{13}^2 & l_{23}^2 & 0 & l_{34}^2 & 1 \\ l_{14}^2 & l_{24}^2 & l_{34}^2 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix}. \quad (\text{B.1})$$

We also define the submatrix M_{ij} of M obtained by deleting its i -th row and j -th column.

The volume of the tetrahedron T and the surface areas of its faces can be expressed in terms of the determinants of these matrices:

$$\text{vol}(T)^2 = \frac{1}{288} \det(M) \quad (\text{B.2})$$

$$s_i^2 = -\frac{1}{16} \det(M_{i,i}) \quad (\text{B.3})$$

Dihedral angles

The well-known relationship between the volume of a tetrahedron and any of its dihedral angle [75]

$$V = \frac{2}{3l_{ij}} s_i s_j \sin(\theta_{ij}) \quad (\text{B.4})$$

cannot be used directly to compute the latter as it does not distinguish if the angle is obtuse or not. We use instead a result referred to as the law of cosine of dihedrals [76, 77]:

$$\cos \theta_{ij} = \frac{(-1)^{i+j} \det(M_{ij})}{16s_i s_j} \quad (\text{B.5})$$

for $1 \leq i < j \leq 4$. Combining these two equations, we get:

$$\cot \theta_{ij} = \frac{\cos \theta_{ij}}{\sin \theta_{ij}} = \frac{2}{3} \frac{(-1)^{i+j} l_{ij} \det(M_{ij})}{V} \quad (\text{B.6})$$

Derivatives of the volume of a tetrahedron

Lemma 4

Let T be a non degenerate tetrahedron whose volume is V . The derivative of V with respect to the length of the edge P_aP_b is given by:

$$\frac{\delta V}{\delta l_{ab}} = \frac{1}{6} l_{ab}^2 \cot \theta_{ab} \quad (\text{B.7})$$

Proof—The Cayley-Menger matrix M of a non degenerate tetrahedron T is invertible (if it is not invertible, its determinant is 0 and the volume of the tetrahedron is 0). Let us call M^{-1} the inverse of M . Using Jacobi's formula for the differential of a determinant, we get:

$$\begin{aligned} \frac{\delta \det(M)}{\delta l_{ab}} &= \det(M) \text{Tr} \left(M^{-1} \frac{\delta M}{\delta l_{ab}} \right) \\ &= 4 l_{ab} \det(M) (M^{-1})_{ab} \end{aligned} \quad (\text{B.8})$$

where $(M^{-1})_{ab}$ is the element of the matrix M^{-1} at row a and column b : this element is the co-factor of M corresponding to the positions (a, b) , i.e.:

$$(M^{-1})_{ab} = (-1)^{a+b} \frac{\det(M_{ab})}{\det(M)}. \quad (\text{B.9})$$

Therefore,

$$\frac{\delta \det(M)}{\delta l_{ab}} = (-1)^{a+b} 4 l_{ab} \det(M_{ab}) \quad (\text{B.10})$$

Then we have:

$$\begin{aligned} \frac{\delta V}{\delta l_{ab}} &= \frac{1}{576V} \frac{\delta \det(M)}{\delta l_{ab}} \\ &= (-1)^{a+b} \frac{l_{ab}}{144V} \det(M_{ab}) \end{aligned} \quad (\text{B.11})$$

Combining this equation with the equations B.4 and B.5, we validate equation B.7.

Derivatives of the surface areas of the faces of a tetrahedron

Jacobi's formula can also be used to compute the derivatives of the surface areas of the faces of a tetrahedron, based on equation B.3. It is easier however to expand the determinant:

$$s_i^2 = \frac{1}{16} (4l_{kl}^2 l_{jl}^2 - (l_{kl}^2 + l_{jl}^2 - l_{jk}^2)^2) \quad (\text{B.12})$$

Then:

$$\begin{aligned}\frac{\delta s_i}{\delta l_{jk}} &= \frac{l_{jk}}{4s_i} (l_{kl}^2 + l_{jl}^2 - l_{jk}^2) \\ \frac{\delta s_i}{\delta l_{jl}} &= \frac{l_{jl}}{4s_i} (l_{kl}^2 + l_{jk}^2 - l_{jl}^2) \\ \frac{\delta s_i}{\delta l_{kl}} &= \frac{l_{kl}}{4s_i} (l_{jk}^2 + l_{jl}^2 - l_{kl}^2)\end{aligned}\quad (\text{B.13})$$

and

$$\frac{\delta s_i}{\delta l_{ab}} = 0 \quad (\text{B.14})$$

if $a = i$ or $b = i$.

Derivatives of the dihedral angles

Deriving equation B.6 with respect to the length l_{ab} of the edge $P_a P_b$, we get:

$$-(1 + \cot^2 \theta_{ij}) \frac{\delta \theta_{ij}}{\delta l_{ab}} = \delta_{ij;ab} \frac{\cot \theta_{ij}}{l_{ij}} + \frac{2}{3} \frac{(-1)^{i+j} l_{ij}}{V} \frac{\delta \det(M_{ij})}{\delta l_{ab}} - \frac{l_{ab}^2 \cot \theta_{ij} \cot \theta_{ab}}{6V} \quad (\text{B.15})$$

where $\delta_{ij;ab}$ is 1 if the pair (i, j) is equal to the pair (a, b) and equal to 0 otherwise.

All terms in this equation are known except for the derivatives of $\det(M_{ij})$. While we could use Jacobi's formula to compute these derivatives, it is easier to expand the determinant:

$$\det(M_{ij}) = 2r_{ij}^2(r_{ik}^2 + r_{il}^2 - r_{kl}^2) - (r_{ij}^2 + r_{ik}^2 - r_{jk}^2)(r_{ij}^2 + r_{il}^2 - r_{jl}^2) \quad (\text{B.16})$$

Its derivatives with respect to each edge length are then straightforward.

Figure 1. Voronoi decomposition and dual complex

Given a finite set of disks, the Voronoi diagram decomposes the plane into regions, one per disk, such that any point in the region assigned to disk S_i is closer to that disk than to any other disk, where the distance to S_i is defined as $\|x - z_i\|^2 - r_i^2$. In the drawing, we restrict the Voronoi diagram (dashed lines) to within the portion of the plane covered by the disks (magenta) and get a decomposition of the union into convex regions. The dual Delaunay triangulation is obtained by drawing edges between circle centers of neighboring Voronoi regions. The dual complex is a subset of the Delaunay triangulation, limited to the edges (in blue) and triangles (light green) whose corresponding Voronoi regions intersect within the union of disks.

Figure 2. Illustration of the discrete flow and pockets in a union of disks

The dual complex of the union of disks is shown in red; all triangles in the Delaunay complex that do not belong to the dual complex are referred to as empty. Acute empty triangles contain their orthocenters: they correspond to sinks. We identify them with large blue dots to mark the position of the orthocenter. The obtuse empty triangles either flow to these acute triangles or to the outside ("infinity"). Triangles III, IV and V (shown in light blue) for example flow to infinity: they do not define pockets. The remaining triangles can be partitioned into two groups: region I is completely surrounded by the union of disks and therefore defines a void, while region II is connected to the outside by one mouth, and is referred to as a pocket.

Figure 3. Characterizing the geometry of the Sindbis virus

The Sindbis virus is an RNA virus, member of the alphavirus; it is transmitted by mosquitoes and is responsible for the Sindbis fever, most common in South and East Africa, the Philippines and Australia. The structure of the full virion consists of two protein capsids (the outside capsid made of glycoproteins and the inner nucleocapsid, a lipid bilayer sandwiched between the two capsids, a set of transmembrane domains that cross the lipid bilayer and connect the two capsids, and the single stranded RNA that occupies the cavity inside the nucleocapsid; it was determined by combination of X-ray crystallography on individual proteins of the capsid and cryoelectron microscopy [78]. All images shown here are based on the complete structure of the capsids obtained from the VIPERdb2 database (file 1ld4.vdb); note that this file only includes the proteins (capsids and transmembrane domains). **A.** Surface of the virus: the outer glycoprotein capsid. **B.** Cross section through the capsid, showing the outer capsid, the inner nucleocapsid and the transmembrane. **C.** Cross section of the dual complex corresponding to the two capsids and transmembrane domains. The simplices of this complex define all the terms of the inclusion-exclusion formula needed to compute the volume and surface area of the virus. **D.** The two main pockets identified by UnionBall: the central pocket (in green) occupies the whole region in the center; it includes many large tetrahedra that are cut by the cross section. The second largest pocket (shown in pink) between the two capsids corresponds to the region where the lipid bilayer is found.

Figure 4.
The running times of AlphaVol (with randomization) and UnionBall (without randomization) for computing the regular Delaunay triangulation.

Figure 5. The running times of AlphaVol and UnionBall (with a new floating point filter) for constructing the dual complex.

Figure 6. The running times of AlphaVol and UnionBall for computing the volume and surface area of the dual complex.

Figure 7. The running times of AlphaVol and UnionBall for detecting and measuring voids in large biomolecules.

Figure 8.

The running time of the different steps performed by UnionBall. The timings are computed on a single Intel Xeon processor running at 3.16 GHz with 6MB of cache memory and 8 GB of RAM. UnionBall is written in Fortran, except for all calculation in arbitrary precision arithmetics that are performed with the C library libgmp. The program is compiled with the Intel Fortran and C compilers. Lines correspond to the averages of five runs.

Figure A1.
Intersection of two disks.

Figure A2.

A. Intersection of three balls. **B.** The core tetrahedron T that defines the intersection of the three balls. z_i , z_j and z_k are the centers of the spheres. P_{ijk} is one of the two points common to the three spheres; as such, it is located at distances r_i , r_j and r_k from z_i , z_j and z_k , respectively.

Figure A3.

Intersection of three spheres B_i , B_j and B_k viewed on the flattened surface of B_i . Key to our approach is the spherical quadrangle formed by the two points P_{ijk} and P_{ikj} that are common to the three spheres and by the "centers" of the caps, $z_{i;j}$ and $z_{i;k}$.

Figure A4. Computing the volume of the intersection of 3 balls

A The plane passing through the centers z_i , z_j and z_k of the three balls. $v_{i,j}$ and $v_{i,k}$ are the distances between the center z_i and the Voronoi planes separating i and j , and i and k , respectively, while $y_{i,j}$ and $y_{i,k}$ are the points of intersection between the edges $z_i z_j$ and $z_i z_k$ with these two planes. The contribution $B_{i,jk}$ of B_i to the intersection of the three balls is shown in green. The sector joining z_i to $B_{i,jk}$ is the key to computing its volume. This sector can be divided into three parts: $B_{i,jk}$ itself, and two fractions of cones $F_{ij,k}$ and $F_{ik,j}$, filled in blue and red, respectively. **B.** Projected view on the plane identified with arrows on panel A, i.e. the Voronoi plane between balls B_i and B_j . The base of the fraction of cone $F_{ij,k}$ is shown filled in blue.

Figure A5. Computing the surface area of the intersection of four spheres

A Projected view on the flattened surface of S_i . Key to our approach is the spherical triangle ST formed by the "centers" of the caps, $z_{i,j}$, $z_{i,k}$ and $z_{i,l}$, corresponding to the points of intersection of S_i with the lines $z_i z_j$, $z_i z_k$ and $z_i z_l$, respectively. **B**. The surface area of ST is computed using an inclusion exclusion formula.

Figure A6.

A We consider the sector of B_i whose apex z_i joins the spherical triangle ST . This sector can be divided into two parts: the region F_i (shown in grey) corresponding to the intersection of the tetrahedron T_4 formed by the centers of the four balls and the three Voronoi planes that separates B_i from the three other balls, and the region G_i (light grey) that sits between the sphere S_i and these three planes. The volume of G_i is computed by inclusion-exclusion (see text). F_i is the union of three pyramids, all three with apex z_i ; their bases lie in the three Voronoi planes; for example, the base of the pyramid corresponding to B_i and B_j is labeled A_l . **B.** A_l is the quadrilateral defined by $y_{i,j}$ (the center of the disk of intersection between B_i and B_j), D_k and D_l (the projections of $y_{i,j}$ on the Voronoi planes between B_i and B_k and B_l , respectively), and P_{ijkl} the Voronoi vertex dual to the tetrahedron T_4 .

Table 1

CPU times for measuring biomolecules using AlphaVol

Molecules ^a	Number of atoms	Delaunay	Dual complex	Volume	Pockets
1TIM	2288	0.02	0.01	0.01	0.02
GroEL	66136	1.76	0.35	0.80	23.16
1ihm	677040	78.27	9.39	5.87	2810.00
2dum	5214540	8577.00	99.00	59.30	205818.00

^aThe four different molecules are: the chicken triose phosphate isomerase (PDB code 1TIM), the GroEL chaperonin (PDB code 1SX3), a Norwalk virus (PDB code 1ihm; we use the fully reconstructed capsid available at the viperdb web site [65], and the full capsid of a human adenovirus, PDB code 2dum, available at viperdb. For each molecule, we compute the accessible surface area, the corresponding volume inaccessible to solvent, and identify and measure all pockets. Calculation are performed on a single Intel processor running at 3.16 GHz, with 6MB of cache memory and 8 GB of RAM. Computing times are reported in seconds.