1

GEOMETRIC MODELS OF PROTEIN STRUCTURE AND FUNCTION PREDICTION

1.1 INTRODUCTION

Three-dimensional atomic structures of protein molecules provide rich information for understanding how these working molecules of a cell carry out their biological functions. With the amount of solved protein structures rapidly accumulating, computation of geometric properties of protein structure becomes an indispensable component in studies of modern biochemistry and molecular biology. Before we discuss methods for computing the geometry of protein molecules, we first briefly describe how protein structures are obtained experimentally.

There are primarily three experimental techniques for obtaining protein structures: X-ray crystallography, solution nuclear magnetic resonance (NMR), and recently freeze-sample electron microscopy (cryo-EM). In X-ray crystallography, the diffraction patterns of X-ray irradiation of a high-quality crystal of the protein molecule are measured. Since the diffraction is due to the scattering of X-rays by the electrons of the molecules in the crystal, the position, the intensity, and the phase of each recorded diffraction spot provide information for the reconstruction of an *electron density map* of atoms in the protein molecule. Based on independent information of the amino acid sequence, a model of the protein conformation is then derived by fitting model conformations of residues to the electron density map. An iterative process called *refinement* is then applied to improve the quality of the fit of the electron density map. The final model of the protein conformation consists of the coordinates of each of the non-hydrogen atoms [46].

The solution NMR technique for solving protein structure is based on measuring the tumbling and vibrating motion of the molecule in solution. By assessing the chemical shifts of atomic nuclei with spins due to interactions with other atoms in the vicinity, a set of estimated distances between specific pairs of atoms can be derived from NOSEY spectra. When a large number of such distances are obtained, one can derive a set of conformations of the protein molecule, each being consistent with all of the distance constraints [10]. Although determining conformations from either X-ray diffraction patterns or NMR spectra is equivalent to solving an ill-posed inverse problem, a technique such as Bayesian Markov chain Monte Carlo with parallel tempering has been shown to be effective in obtaining protein structures from NMR spectra [52].

1.2   THEORY AND MODEL

1.2.1   Idealized Ball Model

The shape of a protein molecule is complex. The chemical properties of atoms in a molecule are determined by their electron charge distribution. It is this distribution that generates the scattering patterns of the X-ray diffraction. Chemical bonds between atoms lead to transfer of electronic charges from one atom to another, and the resulting isosurfaces of the electron density distribution depend not only on the location of individual nuclei but also on interactions between atoms. This results in an overall complicated isosurface of electron density [2].

The geometric model of macromolecule amenable to convenient computation is an idealized model, where the shapes of atoms are approximated by three-dimensional balls. The shape of a protein or a DNA molecule consisting of many atoms is then the space-filling shape taken by a set of atom balls. This model is often called the *interlocking hard-sphere model*, the *fused ball model*, the *space-filling model* [32,47,49,51], or the *union of ball* model [12]. In this model, details in the distribution of electron density (e.g., the differences between regions of covalent bonds and noncovalent bonds) are ignored. This idealization is quite reasonable, as it reflects the fact that the electron density reaches maximum at a nucleus and its magnitude decays almost spherically away from the point of the nucleus. Despite possible inaccuracy, this idealized model has found wide acceptance, because it enables quantitative measurement of important geometric properties (such as area and volume) of molecules. Insights gained from these measurements correlate well with experimental observations [9,21,32,48–50].

In this idealization, the shape of each atom is that of a ball, and its size parameter is the ball radius or atom radius. There are many possible choices for the parameter set of atomic radii [47,56]. Frequently, atomic radii are assigned the values of their van der Waals radii [7]. Among all these atoms, the hydrogen atom has the smallest mass and has a much smaller radius than those of other atoms.

For simplification, the model of *united atom* is often employed to approximate the union of a heavy atom and the hydrogen atoms connected by a covalent bond.

In this case, the radius of the heavy atom is increased to approximate the size of the union of the two atoms. This practice significantly reduces the total number of atom balls in the molecule. However, this approach has been questioned for possible inadequacy [60].

The mathematical model of this idealized model is that of the union of balls [12]. For a molecule M of n atoms, the ith atom is modeled as a ball b_i, whose center is located at $z_i \in \mathbb{R}^3$, and the radius of this ball is $r_i \in \mathbb{R}$, namely, we have $b_i \equiv \{x | x \in \mathbb{R}^3, ||x - z_i|| \le r_i\}$ parameterized by (z_i, r_i). The molecule M is formed by the union of a finite number n of such balls defining the set $\mathcal{B}$:

$$M = \bigcup \mathcal{B} = \bigcup_{i=1}^{n} \{b_i\}.$$

It creates a space-filling body corresponding to the union of the excluded volumes $\mathrm{vol}\left(\bigcup_{i=1}^{n} \{b_i\}\right)$ [12]. When the atoms are assigned the van der Waals radii, the boundary surface $\partial \bigcup \mathcal{B}$ of the union of balls is called the *van der Waals* surface.

1.2.2 Surface Models of Proteins

Protein folds into native three-dimensional shape to carry out its biological functional roles. The interactions of a protein molecule with other molecules (such as ligand, substrate, or other protein) determine its functional roles. Such interactions occur physically on the surfaces of the protein molecule.

The importance of the protein surface was recognized very early on. Lee and Richards developed the widely used *solvent accessible surface* (SA) model, which is also often called the *Lee–Richards surface* model [32]. Intuitively, this surface is obtained by rolling a ball of radius r_s everywhere along the van der Waals surface of the molecule. The center of the solvent ball will then sweep out the solvent accessible surface. Equivalently, the solvent accessible surface can be viewed as the boundary surface $\partial \bigcup \mathcal{B}_{r_s}$ of the union of a set of inflated balls $\mathcal{B}_{r_s}$, where each ball takes the position of an atom, but with an inflated radius $r_i + r_s$ (Fig. 1.1a).

The solvent accessible surface in general has many sharp crevices and sharp corners. In hope of obtaining a smoother surface, one can take the surface swept out by the front instead of the center of the solvent ball. This surface is the *molecular surface* (MS model), which is also often called the *Connolly's surface* after Michael Connolly, who developed the first algorithm for computing molecular surface [9]. Both solvent accessible surface and molecular surface are formed by elementary pieces of simpler shape.

Elementary Pieces. For the solvent accessible surface model, the boundary surface of a molecule consists of three types of elements: the convex spherical surface pieces, arcs or curved line segments (possibly a full circle) formed by two intersecting spheres, and a vertex that is the intersection point of three atom spheres. The whole boundary surface of the molecules can be thought of as a surface formed by stitching these elements together.

FIGURE 1.1 Geometric models of protein surfaces. (**a**) The solvent accessible surface (SA surface) is shown in the front. The van der Waals surface (beneath the SA surface) can be regarded as a shrunken version of the SA surface by reducing all atomic radii uniformly by the amount of the radius of the solvent probe $r_s = 1.4$Å. The elementary pieces of the solvent accessible surface are the three convex spherical surface pieces, the three arcs, and the vertex where the three arcs meet. (**b**) The molecular surface (MS, beneath the SA surface) also has three types of elementary pieces: the convex spherical pieces, which are shrunken version of the corresponding pieces in the solvent accessible surface, the concave toroidal pieces, and concave spherical surface. The latter two are also called the re-entrant surface. (**c**) The toroidal surface pieces in the molecular surface correspond to the arcs in the solvent accessible surface, and the concave spherical surface to the vertex. The set of elements in one surface can be continuously deformed to the set of elements in the other surface.

Similarly, the molecular surface swept out by the front of the solvent ball can also be thought of as being formed by elementary surface pieces. In this case, they are the convex spherical surface pieces, the toroidal surface pieces, and the concave or inverse spherical surface pieces (Fig. 1.1b). The latter two types of surface pieces are often called the ''re-entrant surfaces'' [9,49].

The surface elements of the solvent accessible surface and the molecular surface are closely related. Imagine a process where atom balls are shrunk or expanded. The vertices in solvent accessible surface becomes the concave spherical surface pieces, the arcs become the toroidal surfaces, and the convex surface pieces become smaller convex surface pieces (Fig. 1.1c). Because of this mapping, these two types of surfaces are combinatorially equivalent and have similar topological properties; that is, they are homotopy equivalent.

However, the SA surface and the MS surface differ in their metric measurement. In concave regions of a molecule, often the front of the solvent ball can sweep out a larger volume than the center of the solvent ball. A void of size close to zero in the solvent accessible surface model will correspond to a void of the size of a solvent ball $(4\pi r_s^3/3)$. It is therefore important to distinguish these two types of measurement when interpreting the results of volume calculations of protein molecules. The intrinsic structures of these fundamental elementary pieces are closely related to several geometric constructs we describe below.

1.2.3   Geometric Constructs

Voronoi Diagram.   Voronoi diagram (Fig. 1.2a), also known as Voronoi tessellation, is a geometric construct that has been used for analyzing protein packing in the early days of protein crystallography [18,20,47]. For two-dimensional Voronoi diagram, we consider the following analogy. Imagine a vast forest containing a number of
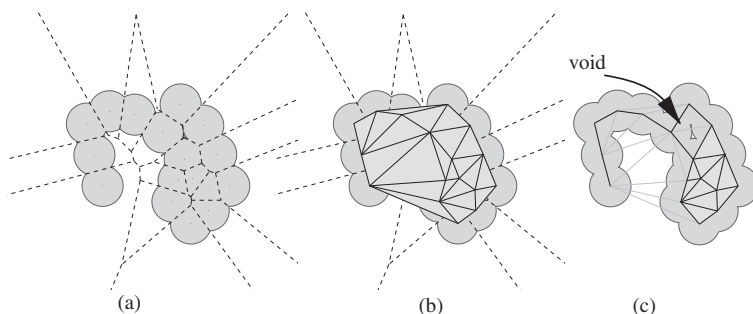
FIGURE 1.2 Geometry of a simplified two-dimensional model molecule to illustrate the geometric constructs and the procedure mapping the Voronoi diagram to the Delaunay triangulation. (**a**) The molecule is formed by the union of atom disks of uniform size. Voronoi diagram is in dashed lines. (**b**) The shape enclosed by the boundary polygon is the *convex hull*. It is tessellated by the *Delaunay triangulation*. (**c**) The alpha shape of the molecule is formed by removing those Delaunay edges and triangles whose corresponding Voronoi edges and Voronoi vertices do not intersect with the body of the molecule. A molecular void is represented in the alpha shape by two empty triangles.

fire observation towers. Each fire ranger is responsible for putting out any fire closer to his/her tower than to any other tower. The set of all trees for which a ranger is responsible constitutes the Voronoi cell associated with his/her tower, and the map of ranger responsibilities, with towers and boundaries marked, constitutes the Voronoi diagram.

We formalize this for three-dimensional space. Consider the point set S of atom centers in three-dimensional space $\mathbb{R}^3$. The *Voronoi region* or *Voronoi cell* V_i of an atom b_i with atom center $z_i \in \mathbb{R}^3$ is the set of all points that are at least as close to z_i as to any other atom centers in S:

$$V_i = \{x \in \mathbb{R}^3 | ||x - z_i|| \leq ||x - z_j||, z_j \in S\}.$$

We can have an alternative view of the Voronoi cell of an atom b_i. Consider the distance relationship between atom center z_i and the atom center z_k of another atom b_k. The plane bisecting the line segment connecting points z_i and z_k divides the full $\mathbb{R}^3$ space into two half-spaces, where points in one half-space is closer to z_i than to z_k, and points in the other allspice is closer to z_k than to z_i. If we repeat this process and take z_k in turn from the set of all atom centers other than z_i, we will have a number of half-spaces where points are closer to z_i than to each of the atom center z_k. The Voronoi region V_i is then the common intersections of these half-spaces, which is convex (see exercises). When we consider atoms of different radii, we replace the Euclidean distance $||x - z_i||$ with the power distance defined as $\pi_i(x) \equiv ||x - z_i||^2 - r_i^2$.

Delaunay Tetrahedrization. Delaunay triangulation in $\mathbb{R}^2$ or Delaunay tetrahedrization in $\mathbb{R}^3$ is a geometric construct that is closely related to the Voronoi diagram

(Fig. 1.2b). In general, it uniquely tessellates or tile up the space of the *convex hull* of the atom centers in $\mathbb{R}^3$ with tetrahedra. Convex hull for a point set is the smallest convex body that contains the point set[1]. The Delaunay tetrahedrization of a molecule can be obtained from the Voronoi diagram. Consider that the Delaunay tetrahedrization is formed by gluing four types of primitive elements together: vertices, edges, triangles, and tetrahedra. Here vertices are just the atom centers. We obtain a Delaunay edge by connecting atom centers z_i and z_j if and only if the Voronoi regions V_i and V_j have a common intersection, which is a planar piece that may be either bounded or extended to infinity. We obtain a Delaunay triangle connecting atom centers z_i, z_j, and z_k if the common intersection of Voronoi regions V_i, V_j and V_k exists, which is either a line segment, a half-line, or a line in the Voronoi diagram. We obtain a Delaunay tetrahedra connecting atom centers z_i, z_j, z_k, and z_l if and only if the Voronoi regions V_i, V_j, V_k, and V_l intersect at a point.

1.2.4   Topological Structures

Delaunay Complex. The structures in both Voronoi diagram and Delaunay tetrahedrization are better described with concepts from algebraic topology. We focus on the intersection relationship in the Voronoi diagram and introduce concepts formalizing the primitive elements. In $\mathbb{R}^3$, between two to four Voronoi regions may have common intersections. We use *simplices* of various dimensions to record these intersection or overlap relationships. We have vertices σ_0 as 0-simplices, edges σ_1 as 1-simplices, triangles σ_2 as 2-simplices, and tetrahedra σ_3 as 3-simplices. Each of the Voronoi plane, Voronoi edge, and Voronoi vertices corresponds to a 1-simplex (Delaunay edge), 2-simplex (Delaunay triangle), and 3-simplex (Delaunay tetrahedron), respectively. If we use 0-simplices to represent the Voronoi cells and add them to the simplices induced by the intersection relationship, we can think of the Delaunay tetrahedrization as the structure obtained by ''gluing'' these simplices properly together. Formally, these simplices form a *simplicial complex* $\mathcal{K}$:

$$\mathcal{K} = \left\{ \sigma_{|I|-1} | \bigcap_{i \in I} V_i \neq \emptyset \right\},$$

where I is an index set for the vertices representing atoms whose Voronoi cells overlap, and $|I| - 1$ is the dimension of the simplex.

Alpha Shape and Protein Surfaces. Imagine we can turn a knob to increase or decrease the size of all atoms simultaneously. We can then have a model of growing

---

[1]For a two-dimensional toy molecule, we can imagine that we put nails at the locations of the atom centers and tightly wrap a rubber band around these nails. The rubber band will trace out a polygon. This polygon and the region enclosed within is the convex hull of the set of points corresponding to the atom centers. Similarly, imagine if we can tightly wrap a tin-foil around a set of points in three-dimensional space, the resulting convex body formed by the tin-foil and space enclosed within is the convex hull of this set of points in $\mathbb{R}^3$.
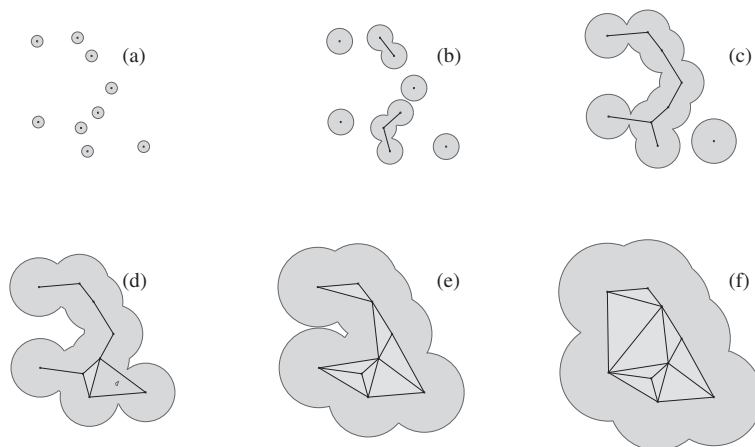
FIGURE 1.3 The family of alpha shapes or dual simplicial complexes for a two-dimensional toy molecule. (**a**) We collect simplices from the Delaunay triangulation as atoms grow by increasing the α value. At the beginning as α grows from $-\infty$, atoms are in isolation and we only have vertices in the alpha shape. (**b** and **c**) When α is increased such that some atom pairs start to intersect, we collect the corresponding Delaunay edges. (**d**) When three atoms intersect as α increases, we collect the corresponding Delaunay triangles. When $\alpha = 0$, the collection of vertices, edges, and triangles form the dual simplicial complex $\mathcal{K}_0$, which reflects the topological structure of the protein molecule. (**e**) More edges and triangles from the Delaunay triangulation are now collected as atoms continue to grow. (**f**) Finally, all vertices, edges, and triangles are now collected as atoms are grown to large enough size. We get back the full original Delaunay complex.

balls and obtain further information from the Delaunay complex about the shape of a protein structure. Formally, we use a parameter $\alpha \in \mathbb{R}$ to control the size of the atom balls. For an atom ball b_i of radius r_i, we modified its radius r_i at a particular α value to $r_i(\alpha) = (r_i^2 + \alpha)^{1/2}$. When $-r_i < \alpha < 0$, the size of an atom is shrunk. The atom could even disappear if $\alpha < 0$ and $|\alpha| > r_i$. With this construction of α, the weighted Voronoi diagram is invariant with regard to α (see exercises). We start to collect the simplices at different α values as we increase α from $-\infty$ to $+\infty$ (see Fig. 1.3 for a two-dimensional example). At the beginning, we only have vertices. When α is increased such that two atoms are close enough to intersect, we collect the corresponding Delaunay edge that connects these two atom centers. When three atoms intersect, we collect the corresponding Delaunay triangle spanning these three atom centers. When four atoms intersect, we collect the corresponding Delaunay tetrahedron.

At any specific α value, we have a *dual simplicial complex* or *alpha complex* $\mathcal{K}_\alpha$ formed by the collected simplices. If all atoms take the incremented radius of $r_i + r_s$ and $\alpha = 0$, we have the dual simplicial complex $\mathcal{K}_0$ of the protein molecule. When α is sufficiently large, we have collected all simplices and we get the full Delaunay complex. This series of simplicial complexes at different α values form a family of
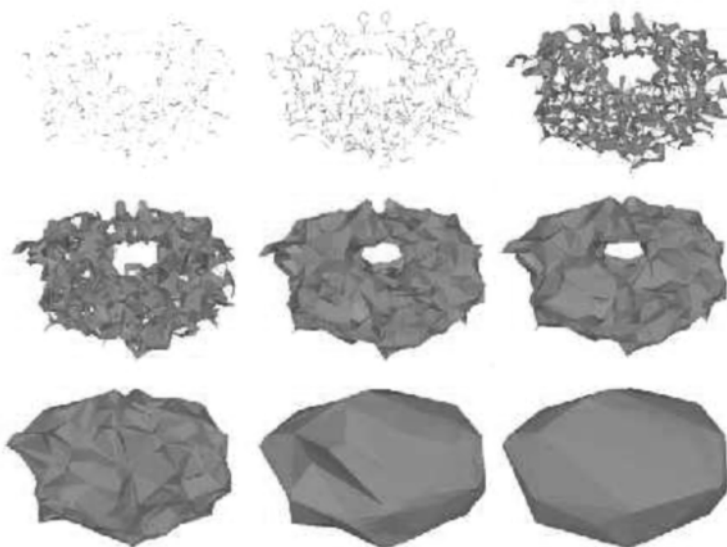
FIGURE 1.4   An illustration of a family of alpha shapes of HIV-1 protease as α value increases from left to right and top to bottom. As α increases, more edges, triangles, and tetrahedra enter the collection of simplices. At each α value, the collected simplices form a simplicial complex. When α is sufficiently large, we obtain the full Delaunay tetrahedrization.

shapes (Fig. 1.3), called *alpha shapes*, each faithfully represents the geometric and topological property of the protein molecule at a particular resolution parameterized by the α value. Figure 1.4 illustrates an example of the alpha shapes of the HIV-1 protease at different α values.

An equivalent way to obtain the alpha shape at $\alpha = 0$ is to take a subset of the simplices, with the requirement that the corresponding intersections of Voronoi cells must overlap with the body of the union of the balls. We obtain the dual complex or alpha shape $\mathcal{K}_0$ of the molecule at $\alpha = 0$ (Fig. 1.2c):

$$\mathcal{K}_0 = \left\{ \sigma_{|I|-1} \,\bigg|\, \bigcap_{i \in I} V_i \cap \bigcup \mathcal{B} \neq \emptyset \right\}.$$

Alpha shapes provides a guide map for computing geometric properties of the structures of biomolecules. Take the molecular surface as an example: The re-entrant surfaces are formed by the concave spherical patch and the toroidal surface. These can be mapped from the boundary triangles and boundary edges of the alpha shape, respectively [14]. Recall that a triangle in the Delaunay tetrahedrization corresponds to the intersection of three Voronoi regions, that is, a Voronoi edge. For a triangle on the boundary of the alpha shape, the corresponding Voronoi edge intersects with the body of the union of balls by definition. In this case, it intersects with the solvent accessible surface at the common intersecting vertex when the three atoms overlap.

This vertex corresponds to a concave spherical surface patch in the molecular surface. For an edge on the boundary of the alpha shape, the corresponding Voronoi plane coincides with the intersecting plane when two atoms meet, which intersect with the surface of the union of balls on an arc. This line segment corresponds to a toroidal surface patch. The remaining part of the surface are convex pieces, which correspond to the vertices, namely, the atoms on the boundary of the alpha shape.

The numbers of toroidal pieces and concave spherical pieces are exactly the numbers of boundary edges and boundary triangles in the alpha shape, respectively. Because of the restriction of bond length and the excluded volume effects, the number of edges and triangles in molecules are roughly on the order of $\mathcal{O}(n)$ [38].

1.2.5 Metric Measurements

We have described the relationship between the simplices and the surface elements of the molecule. Based on this type of relationship, we can compute efficiently size properties of the molecule. We take the problem of volume computation as an example.

Consider a grossly incorrect way to compute the volume of a protein molecule using the solvent accessible surface model. We could define that the volume of the molecule is the summation of the volumes of individual atoms, whose radii are inflated to account for solvent probe. By doing so, we would have significantly inflated the value of the true volume, because we neglected to consider volume overlaps. We can explicitly correct this by following the inclusion–exclusion formula: When two atoms overlap, we subtract the overlap; when three atoms overlap, we first subtract the pair overlaps, and (we then) add back the triple overlap, and so on. This continues when there are four, five, or more atoms intersecting. At the combinatorial level, the principle of inclusion–exclusion is related to the Gauss–Bonnet theorem used by Connolly [9]. The corrected volume $V(\mathcal{B})$ for a set of atom balls $\mathcal{B}$ can then be written as

$$V(\mathcal{B}) = \sum_{\substack{\mathrm{vol}(\bigcap T)>0 \\ T \subset \mathcal{B}}} (-1)^{\dim(T)-1} \mathrm{vol}\left(\bigcap T\right),$$
(1.1)

where $\mathrm{vol}\left(\bigcap T\right)$ represents volume overlap of various degree, $T \subset \mathcal{B}$ is a subset of the balls with nonzero volume overlap: $\mathrm{vol}\left(\bigcap T\right) > 0$.

However, the straightforward application of this inclusion–exclusion formula does not work. The degree of overlap can be very high: Theoretical and simulation studies showed that the volume overlap can be up to 7–8 degrees [29,45]. It is difficult to keep track of these high degree of volume overlaps correctly during computation. In addition, it is also difficult to compute the volume of these overlaps, because of the proliferation of different types of combinations of intersecting balls. That is, we need to quantify the k-volume overlap of each of the $\binom{7}{k}$ possible overlap patterns when k of the 7 balls overlap. Similarly, the volumes of $\binom{8}{k}$ overlapping atoms for all of $k = 2, \ldots, 7$ need to be quantified [45]. It turns out that for three-dimensional

molecules, overlaps of five or more atoms at a time can always be reduced to a ''+'' or a ''−'' signed combination of overlaps of four or fewer atom balls [12]. This requires that the 2-body, 3-body, and 4-body terms in Eq. (1.1) enter the formula if and only if the corresponding edge σ_{ij} connecting the two balls (1-simplex), triangles σ_{ijk} spanning the three balls (2-simplex), and tetrahedron σ_{ijkl} cornered on the four balls (3-simplex) all exist in the dual simplicial complex $\mathcal{K}_0$ of the molecule [12,38]. Atoms corresponding to these simplices will all have volume overlaps. In this case, we have the simplified exact expansion:

$$V(\mathcal{B}) = \sum_{\sigma_i \in \mathcal{K}} \text{vol}(b_i) - \sum_{\sigma_{ij} \in \mathcal{K}} \text{vol}(b_i \cap b_j)$$
$$+ \sum_{\sigma_{ijk} \in \mathcal{K}} \text{vol}(b_i \cap b_j \cap b_k) - \sum_{\sigma_{ijkl} \in \mathcal{K}} \text{vol}(b_i \cap b_j \cap b_k \cap b_l).$$

The same idea is applicable for the calculation of surface area of molecules.

An Example. An example of area computation by the alpha shape is shown in Fig. 1.5. Let b_1, b_2, b_3, b_4 be the four disks. To simplify the notation, we write A_i for the area of b_i, A_{ij} for the area of $b_i \cap b_j$, and A_{ijk} for the area of $b_i \cap b_j \cap b_k$. The total area of the union, $b_1 \cup b_2 \cup b_3 \cup b_4$, is

$$A_{\text{total}} = (A_1 + A_2 + A_3 + A_4)$$
$$- (A_{12} + A_{23} + A_{24} + A_{34})$$
$$+ A_{234}.$$

We add the area of b_i if the corresponding vertex belongs to the alpha complex (Fig. 1.5), we subtract the area of $b_i \cap b_j$ if the corresponding edge belongs to the alpha complex, and we add the area of $b_i \cap b_j \cap b_k$ if the corresponding triangle belongs to the alpha complex. Note that without the guidance of the alpha complex,

(a)   (b)
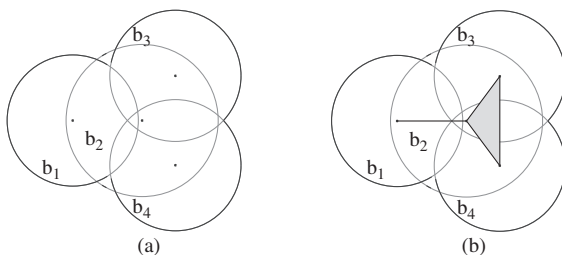
FIGURE 1.5   An example of analytical area calculation. (A) Area can be computed using the direct inclusion–exclusion. (B) The formula is simplified without any redundant terms when using the alpha shape.

the inclusion-exclusion formula may be written as

$$A_{\text{total}} = (A_1 + A_2 + A_3 + A_4)$$
$$- (A_{12} + A_{13} + A_{14} + A_{23} + A_{24} + A_{34})$$
$$+ (A_{123} + A_{124} + A_{134} + A_{234})$$
$$- A_{1234}.$$

This contains 6 canceling redundant terms: $A_{13} = A_{123}$, $A_{14} = A_{124}$, and $A_{134} = A_{1234}$. Computing these terms would be wasteful. Such redundancy does not occur when we use the alpha complex: The part of the Voronoi regions contained in the respective atom balls for the redundant terms do not intersect. Therefore, the corresponding edges and triangles do not enter the alpha complex. In two dimensions, we have terms of at most three disk intersections, corresponding to triangles in the alpha complex. Similarly, in three dimensions the most complicated terms are intersections of four spherical balls, and they correspond to tetrahedra in the alpha complex.

Voids and Pockets. Voids and pockets represent the concave regions of a protein surface. Because shape-complementarity is the basis of many molecular recognition processes, binding and other activities frequently occur in pocket or void regions of protein structures. For example, the majority of enzyme reactions take place in surface pockets or interior voids.

The topological structure of the alpha shape also offers an effective method for computing voids and pockets in proteins. Consider the Delaunay tetrahedra that are not included in the alpha shape. If we repeatedly merge any two such tetrahedra on the condition that they share a 2-simplex triangle, we will end up with discrete sets of tetrahedra. Some of them will be completely isolated from the outside, and some of them are connected to the outside by triangle(s) on the boundary of the alpha shape. The former corresponds to voids (or cavities) in proteins, whereas the latter corresponds to *pockets* and *depressions* in proteins.

A pocket differs from a depression in that it must have an opening that is at least narrower than one interior cross section. Formally, the *discrete flow* [17] explains the distinction between a depression and a pocket. In a two-dimensional Delaunay triangulation, the empty triangles that are not part of the alpha shape can be classified into obtuse triangles and acute triangles. The largest angle of an obtuse triangle is more than 90 degrees, and the largest angle of an acute triangle is less than 90 degrees. An empty obtuse triangle can be regarded as a ''source'' of empty space that ''flows'' to its neighbor, and an empty acute triangle can be regarded to be a ''sink'' that collects flow from its obtuse empty neighboring triangle(s). In Fig. 1.6a, obtuse triangles 1, 3, 4, and 5 flow to the acute triangle 2, which is a sink. Each of the discrete empty spaces on the surface of protein can be organized by the flow systems of the corresponding empty triangles: Those that flow together belong to the same discrete empty space. For a pocket, there is at least one sink among the empty triangles. For a depression, all triangles are obtuse, and the discrete flow goes from
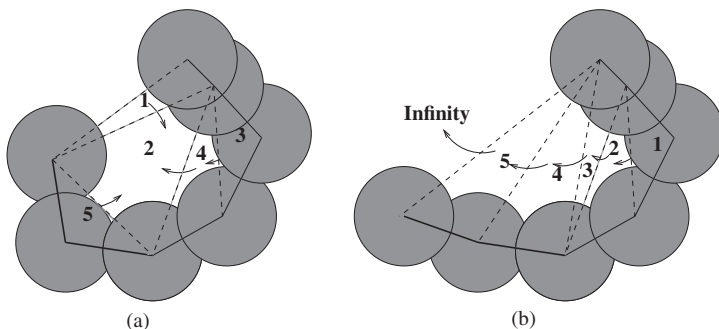
FIGURE 1.6   Discrete flow of empty space illustrated for two-dimensional disks.

one obtuse triangle to another, from the innermost region to outside the convex hull. The discrete flow of a depression therefore goes to infinity. Figure 1.6b gives an example of a depression formed by a set of obtuse triangles.

Once voids and pockets are identified, we can apply the inclusion–exclusion principle based on the simplices to compute the exact size measurement (e.g., volume and area) of each void and pocket [17,39]. Figure 1.7 shows the computed binding surface pockets on Ras21 protein and FtsZ protein.

The distinction between voids and pockets depends on the specific set of atomic radii and the solvent radius. When a larger solvent ball is used, the radii of all atoms will be inflated by a larger amount. This could lead to two different outcomes. A void or pocket may become completely filled and disappear. On the other hand, the inflated atoms may not fill the space of a pocket, but may close off the opening of the pocket. In this case, a pocket becomes a void. A widely used practice in the
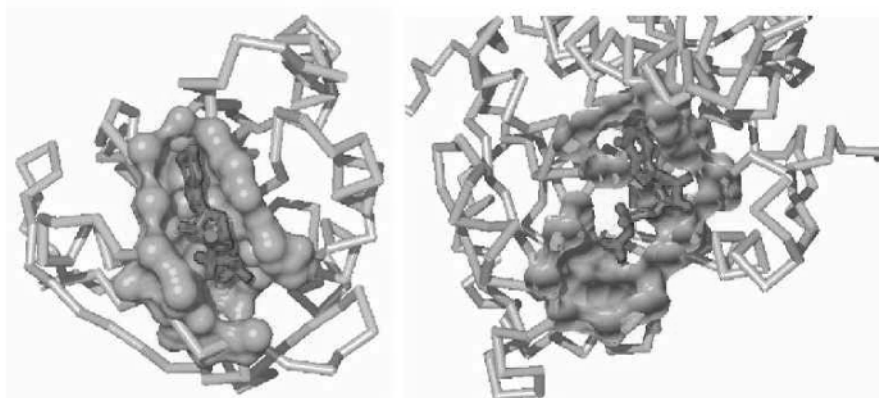
FIGURE 1.7   The computed surface pockets of binding sites on Ras21 protein and FtsZ protein.

past was to adjust the solvent ball and repeatedly compute voids, in the hope that some pockets will become voids and hence be identified by methods designed for cavity/void computation. The pocket algorithm [17] and tools such as CASTP [11,40] often makes this unnecessary.

1.3 ALGORITHM AND COMPUTATION

Computing Delaunay Tetrahedrization and Voronoi Diagram. It is easier to discuss the computation of tetrahedrization first. The incremental algorithm developed in [16] can be used to compute the weighted tetrahedrization for a set of atoms of different radii. For simplicity, we sketch the outline of the algorithm below for two-dimensional unweighted Delaunay triangulation.

The intuitive idea of the algorithm can be traced back to the original observation of Delaunay. For the Delaunay triangulation of a point set, the circumcircle of an edge and a third point forming a Delaunay triangle must not contain a fourth point. Delaunay showed that if all edges in a particular triangulation satisfy this condition, the triangulation is a Delaunay triangulation. It is easy to come up with an arbitrary triangulation for a point set. A simple algorithm to convert this triangulation to the Delaunay triangulation is therefore to go through each of the triangles and then make corrections using ''flips'' discussed below if a specific triangle contains an edge violating the above condition. The basic ingredients for computing Delaunay tetrahedrization are generalizations of these observations. We discuss the concept of *locally Delaunay* edge and the *edge-flip* primitive operation below.

Locally Delaunay Edge. We say an edge *ab* is locally Delaunay if either it is on the boundary of the convex hull of the point set or it belongs to two triangles *abc* and *abd*, and the circumcircle of *abc* does not contain *d* (e.g., edge *cd* in Fig. 1.8a).
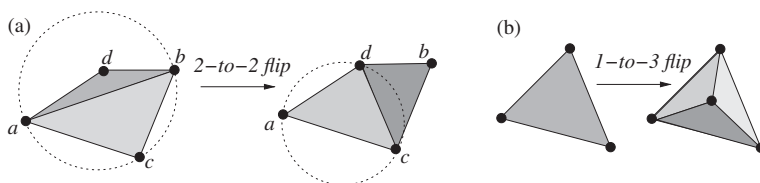
FIGURE 1.8 An illustration of *locally Delaunay edge* and *flips*. (**a**) For the quadrilateral *abcd*, edge *ab* is not locally Delaunay, as the circumcircle passing through edge *ab* and a third point *c* contains a fourth point *d*. Edge *cd* is locally Delaunay, as *b* is outside the circumcircle *adc*. An *edge-flip* or *2-to-2 flip* replaces edge *ab* by edge *cd*, and replace the original two triangles *abc* and *adb* with two new triangles *acd* and *bcd*. (**b**) When a new vertex is inserted, we replace the old triangle containing this new vertex with three new triangles. This is called *1-to-3* flip.

Edge-Flip. If *ab* is not locally Delaunay (edge *ab* in Fig. 1.8a), then the union of the two triangles $abc \cup abd$ is a convex quadrangle *acbd*, and edge *cd* is locally Delaunay. We can replace edge *ab* by edge *cd*. We call this an *edge-flip* or *2-to-2 flip*, as two old triangles are replaced by two new triangles.

We recursively check each boundary edge of the quadrangle *abcd* to see if it is also locally Delaunay after replacing *ab* by *cd*. If not, we recursively edge-flip it.

Incremental Algorithm for Delaunay Triangulation. Assume that we have a finite set of points (namely, atom centers) $S = \{z_1, z_2, \ldots, z_i, \ldots, z_n\}$. We start with a large auxiliary triangle that contains all these points. We insert the points one by one. At all times, we maintain a Delaunay triangulation $\mathcal{D}_i$ up to insertion of point z_i.

After inserting point z_i, we search for the triangle τ_{i-1} that contains this new point. We then add z_i to the triangulation and split the original triangle τ_{i-1} into three smaller triangles. This split is called *1-to-3 flip*, as it replaces one old triangle with three new triangles. We then check if each of the three edges in τ_{i-1} still satisfies the locally Delaunay requirement. If not, we perform a recursive edge-flip. This algorithm is summarized in Algorithm I.

Algorithm I Delaunay Triangulation

Obtain random ordering of points $\{z_1, \cdots, z_n\}$;
for $i = 1$ to n **do**
    find τ_{i-1} such $z_i \in \tau_{i-1}$;
    add z_i, and split τ_{i-1} into three triangles (1-to-3 flip);
    while any edge *ab* not locally Delaunay **do**
        flip *ab* to other diagonal *cd* (2-to-2 edge flip);
    end while
end for

In $\mathbb{R}^3$, the algorithm of tetrahedrization becomes more complex, but the same basic ideas apply. In this case, we need to locate a tetrahedron instead of a triangle that contains the newly inserted point. The concept of locally Delaunay is replaced by the concept of locally convex, and there are flips different than the 2-to-2 flip in $\mathbb{R}^3$ [16]. Although an incremental approach (i.e., sequentially adding points) is not necessary for Delaunay triangulation in $\mathbb{R}^2$, it is necessary in $\mathbb{R}^3$ to avoid non-flippable cases and to guarantee that the algorithm will terminate. This incremental algorithm has excellent expected performance [16].

The computation of Voronoi diagram is conceptually easy once the Delaunay triangulation is available. We can take advantage of the mathematical duality and compute all of the Voronoi vertices, edges, and planar faces from the Delaunay tetrahedra, triangles, and edges (see exercises). Because one point z_i may be an vertex of many Delaunay tetrahedra, the Voronoi region of z_i therefore may contain many Voronoi vertices, edges, and planar faces. The efficient quad-edge data structure can be used for software implementation [24].

Volume and Area Computation. Let V and A denote the volume and area of the molecule, respectively, $\mathcal{K}_0$ for the alpha complex, σ for a simplex in $\mathcal{K}$, i for a vertex, ij for an edge, ijk for a triangle, and $ijkl$ for a tetrahedron. The algorithm for volume and area computation can be written as Algorithm II.

Algorithm II Volume and Area Measurement

$V := A := 0.0;$

for all $\sigma \in \mathcal{K}$ **do**

  if σ is a vertex i **then**

    $V := V + \text{vol}(b_i); A := A + \text{area}(b_i);$

  end if

  if σ is an edge ij **then**

    $V := V - \text{vol}(b_i \cap b_j); A := A - \text{area}(b_i \cap b_j);$

  end if

  if σ is a triangle ijk **then**

    $V := V + \text{vol}(b_i \cap b_j \cap b_k); A := A + \text{area}(b_i \cap b_j \cap b_k);$

  end if

  if σ is a tetrahedron $ijkl$ **then**

    $V := V - \text{vol}(b_i \cap b_j \cap b_k \cap b_l); A := A - \text{area}(b_i \cap b_j \cap b_k \cap b_l);$

  end if

end for

Additional details of volume and area computation can be found in references 14 and 38.

Software. The CASTP webserver for pocket computation can be found at cast.engr.uic.edu. There are other studies that compute or use Voronoi diagrams of protein structures [8,23,25], although not all computes the weighted version which allows atoms to have different radii.

In this short description of algorithm, we have neglected many details important for geometric computation. For example, the problem of how to handle geometric degeneracy, namely, when three points are collinear, or when four points are co-planar. Interested readers should consult the excellent monograph by Edelsbrunner for a detailed treatise of these and other important topics in computational geometry [13].

1.4 APPLICATIONS

1.4.1 Protein Packing

An important application of the Voronoi diagram and volume calculation is the measurement of protein packing. Tight packing is an important feature of protein structure [47,48] and is thought to play important roles in protein stability and folding dynamics [33]. The packing density of a protein is measured by the ratio of its van der Waals volume and the volume of the space it occupies. One approach is to calculate
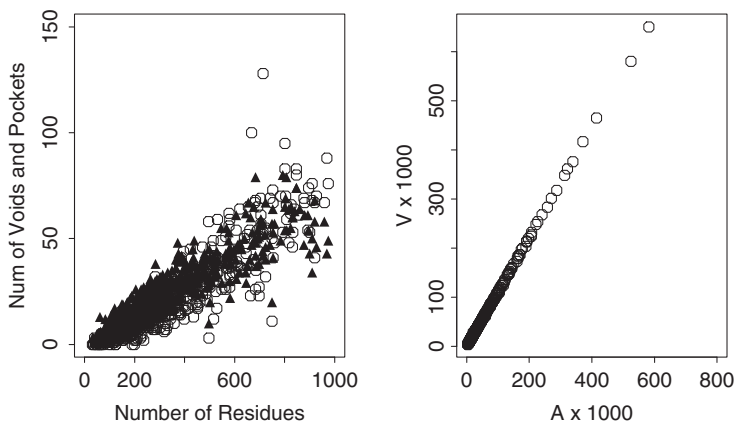
FIGURE 1.9   Voids and pockets for a set of 636 proteins representing most of the known protein folds, and the scaling behavior of the geometric properties of proteins. (left) The number of voids and pockets detected with a 1.4 Å probe is linearly correlated with the number of residues in a protein. Only proteins with less than 1000 residues are shown. Solid triangles and empty circles represent the pockets and the voids, respectively. (right) The van der Waals (*vdw*) volume and van der Waals area of proteins scale linearly with each other. Similarly, molecular surface (*ms*) volume also scales linearly with molecular surface area using a probe radius of 1.4 Å. (Data not shown. Figure adapted from reference 37.)

the packing density of buried residues and atoms using the Voronoi diagram [47,48]. This approach was also used to derive radii parameters of atoms [56].

Based on the computation of voids and pockets in proteins, a detailed study surveying major representatives of all known protein structural folds showed that there is a substantial amount of voids and pockets in proteins [37]. On average, every 15 residues introduces a void or a pocket (Fig. 1.9 (left side)). For a perfectly solid three-dimensional sphere of radius r, the relationship between volume $V = 4\pi r^3/3$ and surface area $A = 4\pi r^2$ is $V \propto A^{3/2}$. In contrast, Fig. 1.9 (right side) shows that the van der Waals volume scales linearly with the van der Waals surface areas of proteins. The same linear relationship holds irrespective of whether we relate molecular surface volume and molecular surface area, or solvent accessible volume and solvent accessible surface area. This and other scaling behavior point out that protein interior is not packed as tight as solid [37]. Rather, packing defects in the form of voids and pockets are common in proteins.

If voids and pockets are prevalent in proteins, an interesting question is what is then the origin of the existence of these voids and pockets. This question was studied by examining the scaling behavior of packing density and coordination number of residues through the computation of voids, pockets, and edge simplices in the alpha shapes of random compact chain polymers [62]. For this purpose, a 32-state discrete state model was used to generate a large ensemble of compact selfavoiding walks. This is a difficult task, as it is very challenging to generate a large number

of independent conformations of very compact chains that are self-avoiding. The results in reference 63 showed that it is easy for compact random chain polymers to have similar scaling behavior of packing density and coordination number with chain length. This suggests that proteins are not optimized by evolution to eliminate voids and pockets, and the existence of many pockets and voids is random in nature and is due to the generic requirement of compact chain polymers. The frequent occurrence and the origin of voids and pockets in protein structures raise a challenging question: How can we distinguish voids and pockets that perform biological functions such as binding from those formed by random chance? This question is related to the general problem of protein function prediction.

1.4.2   Predicting Protein Functions from Structures

Conservation of protein structures often reveals a very distant evolutionary relationship, which are otherwise difficult to detect by sequence analysis [55]. Comparing protein structures can provide insightful ideas about the biochemical functions of proteins (*e.g.*, active sites, catalytic residues, and substrate interactions) [26,42,44].

A fundamental challenge in inferring protein function from structure is that the functional surface of a protein often involves only a small number of key residues. These interacting residues are dispersed in diverse regions of the primary sequences and are difficult to detect if the only information available is the primary sequence. Discovery of local spatial motifs from structures that are functionally relevant has been the focus of many studies.

Graph-Based Methods for Spatial Patterns in Proteins.   To analyze local spatial patterns in proteins, Artymiuk *et al.* developed an algorithm based on subgraph isomorphism detection [1]. By representing residue side-chains as simplified pseudo-atoms, a molecular graph is constructed to represent the patterns of side-chain pseudo-atoms and their interatomic distances. A user-defined query pattern can then be searched rapidly against the Protein Data Bank for similarity relationship. Another widely used approach is the method of geometric hashing. By examining spatial patterns of atoms, Fischer *et al.* developed an algorithm that can detect surface similarity of proteins [19,43]. This method has also been applied by Wallace *et al.* for the derivation and matching of spatial templates [59]. Russell developed a different algorithm that detects side-chain geometric patterns common to two protein structures [53]. With the evaluation of statistical significance of measured root mean square distance, several new examples of convergent evolution were discovered, where common patterns of side-chains were found to reside on different tertiary folds.

These methods have a number of limitations. Most require a user-defined template motif, restricting their utility for automated database-wide search. In addition, the size of the spatial pattern related to protein function is also often restricted.

Predicting Protein Functions by Matching Pocket Surfaces.   Protein functional surfaces are frequently associated with surface regions of prominent concavity [30,40]. These include pockets and voids, which can be accurately computed as we have

discussed. Computationally, one wishes to automatically identify voids and pockets on protein structures where interactions exist with other molecules such as substrate, ions, ligands, or other proteins.

Binkowski et al. developed a method for predicting protein function by matching a surface pocket or void on a protein of unknown or undetermined function to the pocket or void of a protein of known function [4,6]. Initially, the Delaunay tetrahedrization and alpha shapes for almost all of the structures in the PDB databank are computed [11]. All surface pockets and interior voids for each of the protein structures are then exhaustively computed [17,39]. For each pocket and void, the residues forming the wall are then concatenated to form a short sequence fragment of amino acid residues while ignoring all intervening residues that do not participate in the formation of the wall of the pocket or void. Two sequence fragments, one from the query protein and another from one of the proteins in the database, both derived from pocket or void surface residues, are then compared using dynamic programming. The similarity score for any observed match is assessed for statistical significance using an empirical randomization model constructed for short sequence patterns.

For promising matches of pocket/void surfaces showing significant sequence similarity, we can further evaluate their similarity in shape and in relative orientation. The former can be obtained by measuring the coordinate root mean square distance (RMSD) between the two surfaces. The latter is measured by first placing a unit sphere at the geometric center $z_0 \in \mathbb{R}^3$ of a pocket/void. The location of each residue $z = (x, y, z)^T$ is then projected onto the unit sphere along the direction of the vector from the geometric center: $u = (z - z_0)/||z - z_0||$. The projected pocket is represented by a collection of unit vectors located on the unit sphere, and the original orientation of residues in the pocket is preserved. The RMSD distance of the two sets of unit vectors derived from the two pockets are then measured, which is called the ORMSD for *orientation RMSD* [4]. This allows similar pockets with only minor conformational changes to be detected [4].

The advantage of the method of Binkowski *et al.* is that it does not assume prior knowledge of functional site residues and does not require *a priori* any similarity in either the full primary sequence or the backbone fold structures. It has no limitation in the size of the spatially derived motif and can successfully detect patterns small and large. This method has been successfully applied to detect similar functional surfaces among proteins of the same fold but low sequence identities and among proteins of different fold [4,5].

Function Prediction through Models of Protein Surface Evolution. To match local surfaces such as pockets and voids and to assess their sequence similarity, an effective scoring matrix is critically important. In the original study of Binkowski et al. , the BLOSUM matrix was used. However, this is problematic, as BLOSUM matrices were derived from analysis of precomputed large quantities of sequences, while the information of the particular protein of interest has limited or no influence. In addition, these precomputed sequences include buried residues in protein core, whose conservation reflects the need to maintain protein stability rather than to maintain protein function. In references 57 and 58, a continuous time Markov process was
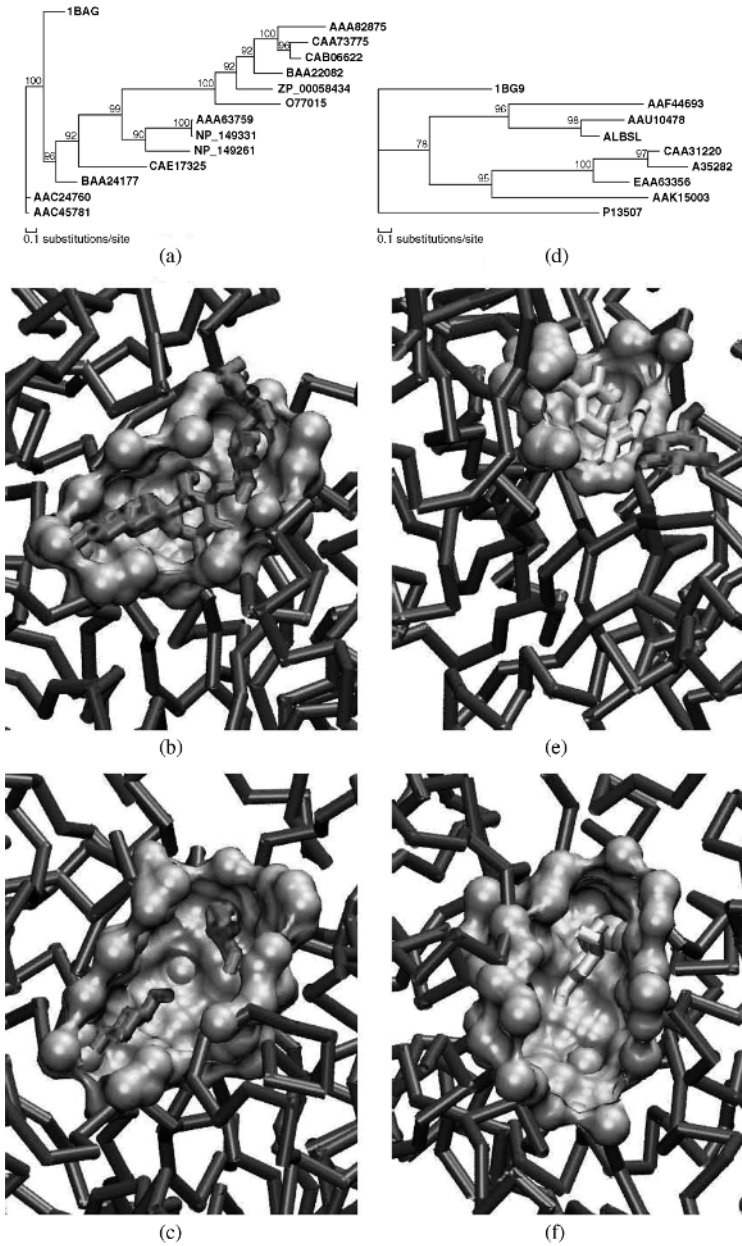
FIGURE 1.10 Protein function prediction as illustrated by the example of alpha amylases. Two template binding surfaces are used to search the database of protein surfaces to identify protein structures that are of similar functions. **(a)** The phylogenetic tree for the template P_DB structure `1bag` from *B. subtilis*. **(b)** The template binding pocket of alpha amylase on `1bag`. **(c)** A matched binding surface on a different protein structure (`1b2y` from human, full sequence identity 22%) obtained by querying with `1bag`. **(d)** The phylogenetic tree for the template structure `1bg9` from *H. vulgare*. **(e)** The template binding pocket on `1bg9`. **(f)** A matched binding surface on a different protein structure (`1u2y` from human, full sequence

developed to explicitly model the substitution rates of residues in binding pockets. Using a Bayesian Markov chain Monte Carlo method, the residue substitution rates at functional pocket are estimated. The substitution rates are found to be very different for residues in the binding site and residues on the remaining surface of proteins. In addition, substitution rates are also very different for residues in the buried core and residues on the solvent exposed surfaces. These rates are then used to generate a set of scoring matrices of different time intervals for residues located in the functional pocket. Application of protein-specific and region-specific scoring matrices in matching protein surfaces result in significantly improved sensitivity and specificity in protein function prediction [57,58].

In a large-scale study of predicting protein functions from structures, a subset of 100 enzyme families are collected from a total of 286 enzyme families containing between 10 and 50 member protein structures with known Enzyme Classification (E.C.) labels. By estimating the substitution rate matrix for residues on the active site pocket of a query protein, a series of scoring matrices of different evolutionary time is derived. By searching for similar pocket surfaces from a database of 770,466 pockets derived from the CASTP database (with the criterion that each must contain at least 8 residues), this method can recover active site surfaces on enzymes similar to that on the query structure at an accuracy of 92% or higher. An example of identifying human amylase using template surfaces from *B. subtilis* and from barley is shown in Fig. 1.10.

The method of surface matching based on evolutionary model is also especially effective in solving the challenging problems of protein function prediction of orphan structures of unknown function (such as those obtained in structural genomics projects), which have only sequence homologs that are themselves hypothetical proteins with unknown functions.

1.5   DISCUSSION AND SUMMARY

A major challenge in studying protein geometry is to understand our intuitive notions of various geometric aspects of molecular shapes and to quantify these notions with mathematical models that are amenable to fast computation. The advent of the union of ball model of protein structures enabled rigorous definition of important geometric concepts such as solvent accessible surface and molecular surface. It also led to the development of algorithms for area and volume calculations of proteins. Deep understanding of the topological structure of molecular shapes is also based on the idealized union of ball model [12]. A success in approaching these problems is exemplified in the development of the pocket algorithm [17]. Another example is the recent development of a rigorous definition of protein–protein binding or interaction interface and algorithm for its computation [3].

Perhaps a more fundamental problem we face is to identify important structural and chemical features that are the determinants of biological problems of interest. For example, we would like to know the shape features that have significant influence on protein solvation, protein stability, ligand specific binding, and protein conforma-

tional changes. It is not clear whether our current geometric intuitions are sufficient or are the correct or the most relevant ones. There may still be important unknown shape properties of molecules that elude us at the moment.

An important application of geometric computation of protein structures is to detect patterns important for protein function. The shape of local surface regions on a protein structure and their chemical texture are the basis of its binding interactions with other molecules. Proteins fold into specific native structure to form these local regions for carrying out various biochemical functions. The geometric shape and chemical pattern of the local surface regions and how they change dynamically are therefore of fundamental importance in computational studies of proteins.

Another important application is the development of geometric potential functions. Potential functions are important for generating conformations, for distinguishing native and near native conformations from other decoy conformations in protein structure predictions [34,36,54,63] and in protein–protein docking [35]. They are also important for peptide and protein design [27,35].

We have not described in detail the approach of studying protein geometry using graph theory. In addition to side-chain pattern analysis briefly discussed earlier, the graph-based protein geometric model also has led to a number of important insights, including the optimal design of model proteins formed by hydrophobic and polar residues [28] and methods for optimal design of side-chain packing [31,61].

Further development of descriptions of geometric shape and topological structure, as well as algorithms for their computation, will provide a solid foundation for studying many important biological problems. The other important tasks are then to show how these descriptors may be effectively used to deepen our biological insights and to develop accurate predictive models of biological phenomena. For example, in computing protein–protein interfaces, a challenging task is to discriminate surfaces that are involved in protein binding from other nonbinding surface regions and to understand in what fashion this depends on the properties of the binding partner protein.

Undoubtedly, evolution plays central roles in shaping up the function and stability of protein molecules. The method of analyzing residue substitution rates using continuous-time Markov models [57,58] and the method of surface mapping of conservation entropy and phylogeny [22,41] only scratches the surface of this important issue. Much remains to be done in incorporating evolutionary information in protein shape analysis for understanding biological functions.

Remark. The original work of Lee and Richards can be found in reference 32, where they also formulated the molecular surface model [49]. Michael Connolly developed the first method for the computation of the molecular surface [9]. Tsai *et al.* described a method for obtaining atomic radii parameter [56]. The mathematical theory of the union of balls and alpha shape was developed by Herbert Edelsbrunner and colleagues [12,15]. Algorithm for computing weighted Delaunay tetrahedrization can be found in reference 16 or in a concise monograph with in-depth discussion of geometric computing [13]. Details of area and volume calculations can be found in references 14, 38 and 39. The theory of pocket computation and applications can be

found in references 17 and 40. Richards and Lim offered a comprehensive review on protein packing and protein folding [50]. A detailed packing analysis of proteins can be found in reference 37. The study on inferring protein function by matching surfaces is described in references 4,58. The study of the evolutionary model of protein binding pocket and its application in protein function prediction can be found in reference 8.

Summary. The accumulation of experimentally solved molecular structures of proteins provides a wealth of information for studying many important biological problems. With the development of a rigorous model of the structure of protein molecules, various shape properties, including surfaces, voids, and pockets, and measurements of their metric properties can be computed. Geometric algorithms have found important applications in protein packing analysis, in developing potential functions, in docking, and in protein function prediction. It is likely the further development of geometric models and algorithms will find important applications in answering additional biological questions.

REFERENCES

1. P. J. Artymiuk, A. R. Poirrette, H. M. Grindley, D. W. Rice, and P. Willett. A graph-theoretic approach to the identification of three-dimensional patterns of amino acid side-chains in protein structure. *J. Mol. Biol.*, **243**:327–344, 1994.

2. R. F. W. Bader. *Atoms in Molecules: A Quantum Theory*. The International Series of Monographs on Chemistry, no. 22. Oxford University Press, New York, 1994.

3. Y. Ban, H. Edelsbrunner, and J. Rudolph. Interface surfaces for protein–protein complexes. In *RECOMB*, pages 205–212, 2004.

4. T. A. Binkowski, L. Adamian, and J. Liang. Inferring functional relationship of proteins from local sequence and spatial surface patterns. *J. Mol. Biol.*, **332**:505–526, 2003.

5. T. A. Binkowski, P. Freeman, and J. Liang. pvSOAR: Detecting similar surface patterns of pocket and void surfaces of amino acid residues on proteins. *Nucleic Acid Res.*, **32**:W555–W558, 2004.

6. T. A. Binkowski, A. Joachimiak, and J. Liang. Protein surface analysis for function annotation in high-throughput structural genomics pipeline. *Protein Sci.*, **14**(12):2972–81, 2005.

7. A. Bondi. VDW volumes and radii. *J. Phys. Chem.*, **68**:441–451, 1964.

8. S. Chakravarty, A. Bhinge, and R. Varadarajan. A procedure for detection and quantitation of cavity volumes proteins. Application to measure the strength of the hydrophobic driving force in protein folding. *J Biol Chem.*, **277**(35):31345–31353, 2002.

9. M. L. Connolly. Analytical molecular surface calculation. *J. Appl. Cryst.*, **16**:548–558, 1983.

10. G. M. Crippen and T. F. Havel. *Distance Geometry and Molecular Conformation*. John Wiley & Sons, New York, 1988.

11. J. Dundas, Z. Ouyang, J. Tseng, 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**(Web Server issue):W116–W118, 2006.

12. H. Edelsbrunner. The union of balls and its dual shape. *Discrete Comput. Geom.*, **13**:415–440, 1995.

13. H. Edelsbrunner. *Geometry and Topology for Mesh Generation*. Cambridge University Press, New York, 2001.

14. H. Edelsbrunner, M. Facello, P. Fu, and J. Liang. Measuring proteins and voids in proteins. In *Proc. 28th Ann. Hawaii Int'l Conf. System Sciences*, Vol. 5, pp. 256–264, IEEE Computer Society Press, Los Alamitos, CA, 1995.

15. H. Edelsbrunner and E. P. Mücke. Three-dimensional alpha shapes. *ACM Trans. Graphics*, **13**:43–72, 1994.

16. H. Edelsbrunner and N. R. Shah. Incremental topological flipping works for regular triangulations. *Algorithmica*, **15**:223–241, 1996.

17. H. Edeslbrunner, M. Facello, and J. Liang. On the definition and the construction of pockets in macromolecules. *Disc. Appl. Math.*, **88**:18–29, 1998.

18. J. L. Finney. Volume occupation, environment and accessibility in proteins. The problem of the protein surface. *J. Mol. Biol.*, **96**:721–732, 1975.

19. D. Fischer, R. Norel, H. Wolfson, and R. Nussinov. Surface motifs by a computer vision technique: Searches, detection, and implications for protein-ligand recognition. *Proteins: Struct., Funct. Genet.*, **16**:278–292, 1993.

20. B. J. Gellatly and J. L. Finney. Calculation of protein volumes: An alternative to the Voronoi procedure. *J. Mol. Biol.*, 161:305–322, 1982.

21. M. Gerstein, F. M. Richards, M. S. Chapman, and M. L. Connolly. *Protein surfaces and volumes: measurement and use, International Tables for Crystallography, Volume F: Crystallography of biological macromolecules,* pp 531–545. Springer, Netherlands, 2001.

22. F. Glaser, T. Pupko, I. Paz, R.E. Bell, D. Shental, E. Martz, and N. Ben-Tal. Consurf: Identification of functional regions in proteins by surface-mapping of phylogenetic information. *Bioinformatics*, 19(1):163–164, 2003.

23. A Goede, R Preissner, and C Frömmel. Voronoi cell: New method for allocation of space among atoms: Elimination of avoidable errors in calculation of atomic volume and density. *J. Comput. Chem.*, **18**(9):1113–1123, 1997.

24. L. Guibas and J. Stolfi. Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *ACM Trans. Graphiques*, **4**:74–123, 1985.

25. Y. Harpaz, M. Gerstein, and C Chothia. Volume changes on protein folding. *Structure (London, England: 1993)*, **2**(7):641–649, 1994.

26. L. Holm and C. Sander. New structure: Novel fold? *Structure*, **5**:165–171, 1997.

27. C. Hu, X. Li, and J. Liang. Developing optimal nonlinear scoring function for protein design. *Bioinformatics*, **20**:3080–3098, 2004.

28. J. Kleinberg. Efficient algorithms for protein sequence design and the analysis of certain evolutionary fitness landscapes. In *RECOMB*, pp. 205–212, 2004.

29. K. W. Kratky. Intersecting disks (and spheres) and statistical mechanics. I. Mathematical basis. *J. Stat. Phys.*, **25**:619–634, 1981.

30. R. A. Laskowski, N. M. Luscombe, M. B. Swindells, and J. M. Thornton. Protein clefts in molecular recognition and function. *Protein Sci.*, **5**:2438–2452, 1996.

31. A. Leaver-Fay, B. Kuhlman, and J. Snoeyink. An adaptive dynamic programming algorithm for the side chain placement problem. In *Pacific Symposium on Biocomputing*, pp. 17–28, 2005.

32. B. Lee and F. M. Richards. The interpretation of protein structures: Estimation of static accessibility. *J. Mol. Biol.*, **55**:379–400, 1971.

33. M. Levitt, M. Gerstein, E. Huang, S. Subbiah, and J. Tsai. Protein folding: The endgame. *Annu. Rev. Biochem.*, **66**:549–579, 1997.

34. X. Li, C. Hu, and J. Liang. Simplicial edge representation of protein structures and alpha contact potential with confidence measure. *Proteins*, **53**:792–805, 2003.

35. X. Li and J. Liang. Computational design of combinatorial peptide library for modulating protein–protein interactions. *Pac. Symp. Biocomput.*, pages 28–39, 2005.

36. X. Li and J. Liang. Geometric cooperativity and anticooperativity of three-body interactions in native proteins. *Proteins*, **60**(1):46–65, 2005.

37. J. Liang and K. A. Dill. Are proteins well-packed? *Biophys. J.*, **81**:751–766, 2001.

38. J. Liang, H. Edelsbrunner, P. Fu, P. V. Sudhakar, and S. Subramaniam. Analytical shape computing of macromolecules I: Molecular area and volume through alpha-shape. *Proteins*, **33**:1–17, 1998.

39. J. Liang, H. Edelsbrunner, P. Fu, P. V. Sudhakar, and S. Subramaniam. Analytical shape computing of macromolecules II: Identification and computation of inaccessible cavities inside proteins. *Proteins*, **33**:18–29, 1998.

40. J. Liang, H. Edelsbrunner, and C. Woodward. Anatomy of protein pockets and cavities: Measurement of binding site geometry and implications for ligand design. *Protein Sci.*, **7**:1884–1897, 1998.

41. O. Lichtarge, H. R. Bourne, and F. E. Cohen. An evolutionary trace method defines binding surfaces common to protein families. *J. Mol. Biol.*, **257**(2):342–358, 1996.

42. A. C. R. Martin, C. A. Orengo, E. G. Hutchinson, A. D. Michie, A. C. Wallace, M. L. Jones, and J. M. Thornton. Protein folds and functions. *Structure*, **6**:875–884, 1998.

43. R. Norel, D. Fischer, H. J. Wolfson, and R. Nussinov. Molecular surface recognition by a computer vision-based technique. *Protein Engineering*, **7**(1):39–46, 1994.

44. C. A. Orengo, A. E. Todd, and J. M. Thornton. From protein structure to function. *Curr. Opini. Struct. Biol.*, **9**(4):374–382, 1999.

45. M. Petitjean. On the analytical calculation of van der waals surfaces and volumes: Some numerical aspects. *J. Comput. Chem.*, **15**:507–523, 1994.

46. G. Rhodes. *Crystallography Made Crystal Clear: A Guide for Users of Macromolecular Models*. Academic Press, Walthan, MA, 1999.

47. F. M. Richards. The interpretation of protein structures: Total volume, group volume distributions and packing density. *J. Mol. Biol.*, **82**:1–14, 1974.

48. F. M. Richards. Areas, volumes, packing, and protein structures. *Annu. Rev. Biophys. Bioeng.*, **6**:151–176, 1977.

49. F. M. Richards. Calculation of molecular volumes and areas for structures of known geometries. *Methods Enzymol.*, **115**:440–464, 1985.

50. F. M. Richards and W. A. Lim. An analysis of packing in the protein folding problem. *Q. Rev. Biophys.*, **26**:423–498, 1994.

51. T. J. Richmond. Solvent accessible surface area and excluded volume in proteins: analytical equations for overlapping spheres and implications for the hydrophobic effect. *J. Mol. Biol.*, **178**:63–89, 1984.

52. W. Rieping, M. Habeck, and M. Nilges. Inferential structure determination. *Science*, **309**(5732):303–306, 2005.

53. R. Russell. Detection of protein three-dimensional side-chain patterns: New examples of convergent evolution. *J. Mol. Biol.*, **279**:1211–1227, 1998.

54. R. K. Singh, A. Tropsha, and I. I. Vaisman. Delaunay tessellation of proteins: four body nearest-neighbor propensities of amino-acid residues. *J. Comp. Biol.*, **3**:213–221, 1996.

55. A. E. Todd, C. A. Orengo, and J. M. Thornton. Evolution of function in protein superfamilies, from a structural perspective. *J. Mol. Biol.*, **307**:1113–1143, 2001.

56. J. Tsai, R. Taylor, C. Chothia, and M. Gerstein. The packing density in proteins: Standard radii and volumes. *J. Mol. Biol.*, **290**(1):253–66, 1999.

57. Y.Y. Tseng and J. Liang. Estimating evolutionary rate of local protein binding surfaces: A bayesian monte carlo approach. *Proceedings of 2005 IEEE-EMBC Conference*, 2005.

58. Y. Y. Tseng and J. Liang. Estimation of amino acid residue substitution rates at local spatial regions and application in protein function inference: A Bayesian Monte Carlo approach. *Mol. Biol. Evol.*, **23**(2):421–436, 2006.

59. A. C. Wallace, N. Borkakoti, and J. M. Thornton. TESS: a geometric hashing algorithm for deriving 3d coordinate templates for searching structural databases. Application to enzyme active sites. *Protein Sci.*, **6**:2308–2323, 1997.

60. J.M. Word, S.C. Lovell, J.S. Richardson, and D.C. Richardson. Asparagine and glutamine: using hydrogen atom contacts in the choice of side-chain amide orientation. *J. Mol. Biol.*, **285**(4):1735–1747, 1999.

61. J. Xu. Rapid protein side-chain packing via tree decomposition. In *RECOMB*, pp. 423–439, 2005.

62. J. Zhang, R. Chen, C. Tang, and J. Liang. Origin of scaling behavior of protein packing density: A sequential monte carlo study of compact long chain polymers. *J. Chem. Phys.*, **118**:6102–6109, 2003.

63. W. Zheng, S. J. Cho, I. I. Vaisman, and A. Tropsha. A new approach to protein fold recognition based on Delaunay tessellation of protein structure. In R. B. Altman, A. K. Dunker, L. Hunter, and T. E. Klein, editors, *Pacific Symposium on Biocomputing'97*, pp. 486–497. World Scientific, Singapore, 1997.

EXERCISES

1.1 For two points $x_1, x_2 \in \mathbb{R}^d$, the line through x_1 and x_2 can be written as $\{x | x = x_1 + \lambda(x_2 - x_1), \ \lambda \in \mathbb{R}\}$. Equivalently, we can define the line as

$$\{x | x = (1 - \lambda)x_1 + \lambda x_2, \quad \lambda \in \mathbb{R}\},$$

or

$$\{x | x = p_1 x_1 + p_2 x_2, \quad p_1, p_2 \in \mathbb{R}, \quad p_1 + p_2 = 1\}.$$

A closed line segment joining x_1 and x_2 is

$$[x_1, x_2] = \{x | x = (1 - \lambda)x_1 + \lambda x_2, \quad 0 \le \lambda \le 1\}.$$

Similarly, an open line segment joining x_1 and x_2 is

$$(x_1, x_2) = \{x | x = (1 - \lambda)x_1 + \lambda x_2, \quad 0 < \lambda < 1\}.$$

A set $S \subseteq \mathbb{R}^d$ is convex if the closed line segment joining every two points of S is in S. Equivalently, S is convex if for $x_1, x_2 \in S, \quad \lambda \in \mathbb{R}, \quad 0 \le \lambda \le 1$ we obtain

$$(1 - \lambda)x_1 + \lambda x_2 \in S.$$

For a nonzero vector $w \in \mathbb{R}^d, w \ne 0$, and $b \in \mathbb{R}$, the point set $\{x | x \in \mathbb{R}^n, \ w \cdot x < b\}$ is an *open half-space* in $\mathbb{R}^d$, and the set $\{x | x \in \mathbb{R}^n, \ wx \le b\}$ is a *closed half-space* in $\mathbb{R}^d$. Show with proof that:

(a) Both an open half-space and a closed half-space are convex.

(b) If $A_1, \ldots, A_n$ is a family of convex sets in $\mathbb{R}^d$, then their intersection $\bigcap_{i=1}^n A_i$ is a convex set. Specifically, the intersection of a set of half-spaces—for example, a Voronoi cell—is convex.

1.2 We can follow the dual relationship to compute the Voronoi diagram from the constructed Delaunay triangulation. In three-dimensional space, a Delaunay vertex corresponds to an atom ball, a Delaunay edge corresponds to a Voronoi plane, a Delaunay triangle corresponds to a Voronoi edge, and a Delaunay tetrahedron corresponds to a Voronoi vertex. To obtain the coordinates of a Voronoi vertex $v = (v_1, v_2, v_3) \in \mathbb{R}^3$ from a Delaunay tetrahedron formed by four atoms $b_i(z_i, r_i), b_j(z_j, r_j), b_k(z_k, r_k)$, and $b_l(z_l, r_l)$, which are located at z_i, z_j, z_k, and z_l, with radii r_i, r_j, r_k and r_l, respectively, we use the fact that the power distance

$$\pi_i(v) \equiv ||v - z_i||^2 - r_i^2$$

from v to $b_i(z_i, r_i)$ is the same as $\pi_j(v), \pi_k(v)$, and $\pi_l(v)$. Denote this power distance as R^2.

(a) Write down the set of quadratic equations whose solution will provide $r = (r_1, r_2, r_3)$ and R^2.

(b) Define functions $\lambda(v) \equiv v \cdot v - R^2$, and $\lambda(z_i) \equiv z_i \cdot z_i - r_i^2$, and define $\lambda(z_j), \lambda(z_k), \lambda(z_k)$, and $\lambda(z_l)$ similarly. Use $\frac{\lambda(v)}{2}, \frac{\lambda(z_i)}{2}, \frac{\lambda(z_j)}{2}, \frac{\lambda(z_k)}{2}$, and $\frac{\lambda(z_l)}{2}$ to simplify the system of quadratic equations into a system of linear equations, whose solution will give r and R^2.

(c) Write down the set of linear equations that determine the Voronoi line dual to a Delaunay triangle.

(d) Write down the linear equation that determines the Voronoi plane dual to a Delaunay edge.

1.3 By growing atom balls using a parameter α, we can generate a family of unions of balls, in which the size of each atom is inflated from r_i to $r_i(\alpha) = (r_i^2 + \alpha)^{1/2}$ [12,15]. We now examine the corresponding Voronoi diagrams.

    (a) In the Voronoi diagram, every point x on the separator surface for the two original atoms (z_i, r_i) and (z_j, r_j) has equal power distances $\pi_i(x)$ and $\pi_j(x)$ to the two atoms. Write down the equation for the separator surface. Is the separator surface elliptic, parabolic, or planar?

    (b) Now we inflate both atoms by α such that we have two new balls with different radii $(z_i, r_i(\alpha))$ and $(z_j, r_j(\alpha))$. Write down the equation for the separator surface.

    (c) What is the relationship between these two separator surfaces? What is the relationship between the two corresponding Voronoi diagrams?

1.4 The Voronoi diagrams can be generalized using different distance functions. When considering atoms of different radii, instead of replacing the Euclidean distance $||x - z_i||$ with the power distance $\pi_i(x)$, we can use the additive distance:

$$d_i(x) \equiv ||x - z_i|| - r_i.$$

The resulting Voronoi diagram is called the additively weighted Voronoi diagram.

    (a) Write down the equation for the separator surface formed by the set of points with equal additive distances to the two atoms (z_i, r_i) and (z_j, r_j). Is the separator surface elliptic, parabolic, or planar?

    (b) Now we inflate both atoms by α such that we have two new balls with different radii $(z_i, r_i + \alpha)$ and $(z_j, r_j + \alpha)$. Write down the equation for the separator surface. Is the separator surface elliptic, parabolic, or planar?

    (c) Is there a simple relationship between these two separator surfaces or between the two corresponding Voronoi diagrams?