A Conical Representation of Hydrogen Bond Geometry for Quantifying Bond Interactions

Chesphongphach Buranasilp
Department of Computer Science
Lehigh University
Bethlehem, Pennsylvania
chb221@lehigh.edu

Brian Y. Chen
Department of Computer Science
Lehigh University
Bethlehem, Pennsylvania
byc210@lehigh.edu

Abstract

We present a new three dimensional representation of hydrogen bond donors and acceptors as spherical cones. The conical representation describes the range of bond lengths and bond angles at which a hydrogen bond can form. We hypothesized that three dimensional intersections of these cones can predict the formation of hydrogen bonds and potentially their contribution to protein-protein interactions. As a result, this representation enables a new technique for identifying similarities in bond formation and bond geometry.

I. Introduction

Algorithms for protein structure comparison often consider a wide range of biophysical mechanisms. The similarities they detect, often in binding site geometry [3], [10], [13], [20] and electric fields [11], [13], can reveal proteins that catalyze the same chemical reactions [4], [9], [16] or proteins with remote evolutionary relationships [12]. Differences in patterns of steric hindrance [5] or electrostatic complementarity [2] can point to variations that select different binding partners. Throughout, the algorithmic representation of the protein and its biophysical mechanisms performs a central role in achieving precise and accurate comparison. Nonetheless, some representations of biophysical mechanisms have been studied more than others. Many methods describe the atoms of proteins as collections of points in space (e.g. [21]), or the solvent accessible surface of proteins as triangular meshes (e.g. [1], [18]). In contrast, few techniques compare the geometry and angular tolerances of intermolecular hydrogen bonds, although bond tolerances play a critical role in techniques for assessing molecular rigidity ([17], [19]).

The specific problem addressed in this paper concerns the case where protein structures are arranged in potentially interacting positions, and it is of interest to assess the degree of compatibility between all potential intermolecular hydrogen bond donors and acceptors. We will assess compatibility of particular donors and acceptors for forming a hydrogen bond not only in that they satisfy acceptable bond length and angle limits, but also that margins exist within these limits for motion by the donor and acceptor without breaking the hydrogen bond. This novel capability, explored first in this paper, enables us to identify donors and acceptors that could be removed through mutation to modify binding preferences. We hypothesize that donor-acceptor pairs that satisfy hydrogen bonding limits with the largest margins will be the biggest contributors to binding specificity.

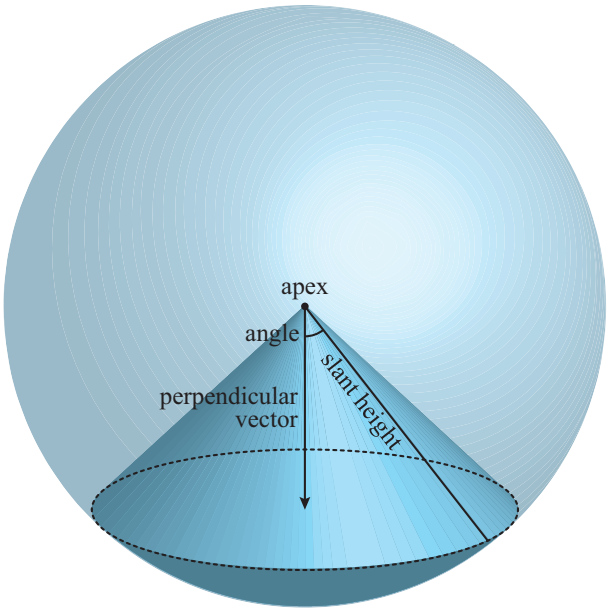


Fig. 1. **A spherical cone**.

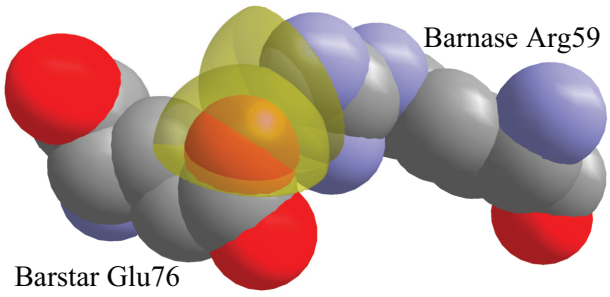


Fig. 2. **A hydrogen bond between Barnase E76 and Barstar R59 (pdb: 1brs).** Spherical cones (angle 90° and slant height 2.3 Å), illustrate potential bond angles and lengths between donor and acceptor. The hydrogen bond hydrogen is not shown, for clarity.

We approach this problem by representing the limits of bond length and angle formation around donors and acceptors using a spherical cone (Fig. 1). Our representation positions one spherical cone at the donor atom, and a second spherical cone at the acceptor atom, and we measure the volume of intersection between these cones in order to estimate the margin of hydrogen bond flexibility (e.g. Fig. 2). We determine the geometry of this intersection using an enhanced version of pClay [7], [8], a parallel algorithm we developed for computing constructive solid geometry operations (CSG) on analytically defined mathematical primitives (Fig 3a). Our enhancements enable CSG operations on a spherical cone primitive that can be made arbitrarily precise, to machine precision, and in parallel on modern multicore architectures.

Our results examine two ways in which spherical cone representations of hydrogen bond donors and acceptors can be used to analyze protein-protein complexes. First, we evaluated how effectively pairs of spherical cones can predict the presence of hydrogen bonds. Second, we evaluated how closely the volume of intersection within a spherical cone pair reflects the contribution of the resulting hydrogen bond to binding affinity. Rather than relying on general thresholds for hydrogen bond lengths and angles, we compared our findings directly to observations of hydrogen bonds from the structural literature, to ensure the most accurate evaluation of our findings. Our results point to new ways to evaluate the existence and similarity of hydrogen bonds from protein structures and models.

II. Methods

A. Definitions

A spherical cone (Fig. 1) is a subset of a sphere defined by the intersection of the sphere and an infinite cone whose apex is at the center of the sphere. The **base** of the spherical cone is a circle at the intersection of the sphere's surface and the cone's surface. We define each spherical cone by its **apex** $a = (a_x, a_y, a_z)$, the apex of the cone, its **slant height** l, a line from the apex to any point on the base, its **perpendicular vector** $\overrightarrow{v} = v_x\hat{\imath} + v_y\hat{\jmath} + v_z\hat{k}$, a vector pointing from the apex to the center of the base, and its **angle** α, the angle between the perpendicular vector and the slant height.

CSG operations approximate set theoretic computations defined on closed sets in three dimensions, which in this are always unions, intersections, or differences. While we define spherical cones as mathematically exact geometric solids, the final output from a CSG operation is an approximation of the exact set theoretic operation, and it is returned as a triangular mesh, which we visualize later in this paper. Multiple CSG operations can be performed at once, to avoid the accumulation of error from this mesh translation.

B. Requirements for CSG operations

We compute CSG operations on spherical cones using pClay [8], a parallel, variable-precision implementation of marching cubes [14] that performs CSG operations on mathematically exact geometric primitives. To make this possible, we defined a spherical cone primitive, which requires three specific functions for the primitive to be implemented. They are containsPoint(), intersectSegment(), and findSurfaceCubes(). These functions are defined as follows:

containsPoint(p) begins with any point p in three dimensions and determines if p is inside or outside the solid. When p is on the surface of the spherical cone, it is considered inside. intersectSegment(s) begins with a line segment s and finds all points where s intersects the surface of the operand, as well as the interior or exterior state of each subsegment between intersections. Finally, given a cubic lattice l that surrounds the cone, findStartingCubes(l) finds a few cubes of the lattice where at least one corner of the cube is inside and one corner is outside the cone. Cubes that satisfy this property are called *surface cubes*. findStartingCubes() supports a function called findAllSurfaceCubes(), which is implemented once for all primitives. This process begins with the surface cubes identified by findStartingCubes, and, following a parallel breadth first search, it walks the lattice along the surface of a cone in order to identify all surface cubes. To accelerate intersections, a bounding box method, GetInternalBoundingBoxVertices(), defines the bounding box of a spherical cone; it is used to avoid marching cubes computations that are not necessary.

C. Marching Cubes on Spherical Cones

While pClay supports multiple geometric primitives and CSG operations, we leave the general description to earlier work [8]. Here, we paraphrase the process as it would be applied to compute the intersection of two cones. First, we define a cubic lattice l, oriented to the x,y, and z axes. Each cube in l will have sides equal to r. The boundary of l is selected so that there are an integer number of cubes in each of the three dimensions, and that both operands are contained completely within l (Fig. 3c).

Once we have defined the l, we call findStartingCubes(l) on both of the input cones, finding several surface cubes on each cone (Fig. 3d). Next, for each set of starting cubes c, we call findAllSurfaceCubes(c), in order to identify all cubes of the lattice that are surface cubes for each cone (Fig. 3e). Using containsPoint(), this process determines the interior/exterior state of the corners of these cubes in relation to specific solids (Fig. 3f). We then compute the interior/exterior state of these points in relation to all other solids in an embarrassingly parallel manner. Once this information is computed, a logical operation based on the desired CSG operation determines whether the corner of each cube is interior or exterior to the final output region (Fig. 3g). This process identifies the subset of cubes identified by the findAllSurfaceCubes() functions that will be surface cubes of the output region.

On the resulting cubes, we find all segments s that will connect a corner that is interior to the output region to a

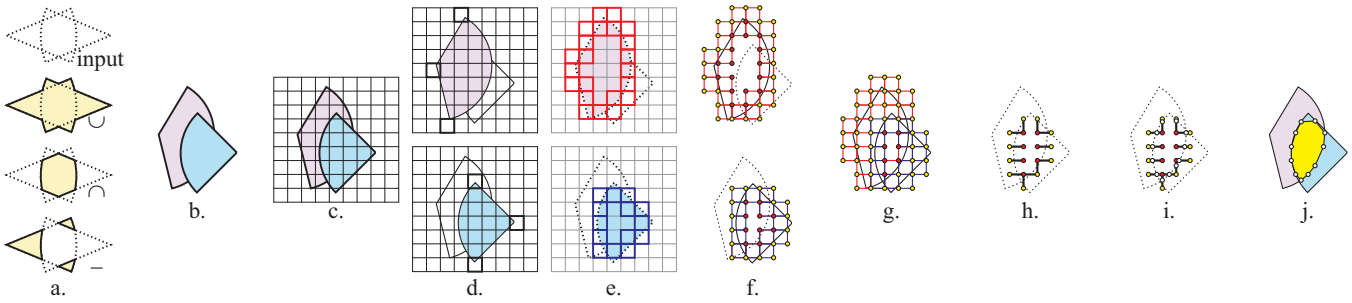


Fig. 3. **CSGoperations on spherical cones. a.** CSG union (∪), intersection (∩), and difference operations (−) performed on input solids (white, dotted outlines) and their outputs (yellow). **b.** two spherical cones. **c.** The lattice l (gray grid). **d.** starting cubes for the red cone (top, black squares), and the blue cone (bottom, black squares). **e.** All surface cubes for the red cone (top, red squares), and blue cone (bottom, blue squares). **f.** Cube corners for both cones, with interior corners shown as red dots and exterior corners shown as yellow dots. **g.** After the logical operation, corners interior to the output intersection are shown as red dots and exterior corners are shown as yellow dots. **h.** segments that connect an interior corner to an exterior corner (heavy black segments). **i.** Segment intersections with the output surface (white dots). **j.** Final output surface generated from segment intersections.

corner that is on the exterior (Fig. 3h), and find the point along these edges that intersects the output surface, using intersectSegment(s) (Fig. 3i). This process is computed in parallel and stored in a datastructure that avoids duplication, ensuring that once it is computed for one cube, it will never be computed again for the adjacent cubes that share the segment.

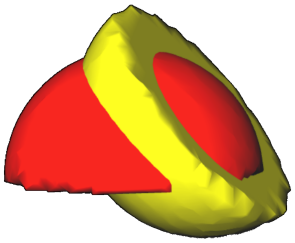


Fig. 4. **Two superposed spherical cones.**

Fig. 5. **Cone intersection.**

Finally, once all edge intersections have been compled in the manner above, we generate triangles within each surface cube of the output region (Fig. 3j). Triangle topology is defined using a lookup table based on the corners that are interior or exterior to the cube, even though the exact position of triangle corners depends on the input data. The set of all triangles generated in this manner produces a closed triangular mesh that approximates the output region (Fig. 4,5). What remains is then to describe how containsPoint(), intersectSegment(), and findSurfaceCubes() is defined for a spherical cone primitive.

D. ContainsPoint(p)

Given a point $p = (p_x, p_y, p_z)$. To determine if p is in the spherical cone, we have to show that p is in both the sphere and the cone. Thus, it is sufficient to show that the euclidean distance from a to p is less than l and the angle of $\overrightarrow{ap}$ and $\overrightarrow{v}$ is less than α. Suppose $\overrightarrow{ap} = m_x\hat{\imath} + m_y\hat{\jmath} + m_z\hat{k}$. Thus, the point p is in the spherical cone if p satisfies the following inequalities.

$$(a_x - p_x)^2 + (a_y - p_y)^2 + (a_z - p_z)^2 < l^2 \quad (1)$$

$$\frac{m_x v_x + m_y v_y + m_z v_z}{\sqrt{m_x^2 + m_y^2 + m_z^2}\sqrt{v_x^2 + v_y^2 + v_z^2}} > cos(\alpha) \quad (2)$$

Equation 1 restricts the point p to being within the sphere of proper radius by using the euclidean distance (Fig. 6), and equation 2 restricts the point p to being within the proper angle by using the dot product (Fig. 7).

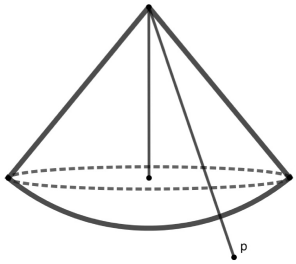


Fig. 6. **p violates equation 1**

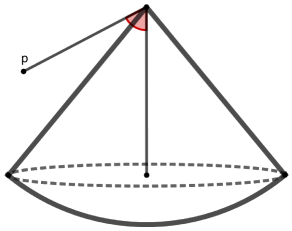


Fig. 7. **p violates equation 2**

E. IntersectSegment(S)

Let S be a line segment parameterized over $t \in [0, 1]$. Suppose $S = \{(s_x + d_x t, s_y + d_y t, s_z + d_z t) \mid t \in [0, 1]\}$. Since intersection points of S and the spherical cone are on either slant surface or spherical surface, it is sufficient to find all points where S intersects the sphere and the cone, and then remove all points those are not contained in the spherical cone. To find intersection points on the slant surface, we have to find all points $p = (p_x, p_y, p_z)$ such that the angle between $\overrightarrow{ap}$ and $\overrightarrow{v}$ is equal α, and the euclidean distance from a to p is less than l. Also, to find intersection points on the spherical surface, we have to find all points $p = (p_x, p_y, p_z)$ such that the angle between $\overrightarrow{ap}$ and $\overrightarrow{v}$ is less than α, and the euclidean distance from a to p is equal l. Suppose $\overrightarrow{ap} = m_x\hat{\imath} + m_y\hat{\jmath} + m_z\hat{k}$. Thus, the point p is an

intersection of S and the spherical cone if p satisfies either one of the following systems.

System I: Intersection on Slant Surface

$$(a_x - p_x)^2 + (a_y - p_y)^2 + (a_z - p_z)^2 < l^2 \quad (3)$$

$$\frac{m_x v_x + m_y v_y + m_z v_z}{\sqrt{m_x^2 + m_y^2 + m_z^2}\sqrt{v_x^2 + v_y^2 + v_z^2}} = cos(\alpha) \quad (4)$$

System II: Intersection on Spherical Surface

$$(a_x - p_x)^2 + (a_y - p_y)^2 + (a_z - p_z)^2 = l^2 \quad (5)$$

$$\frac{m_x v_x + m_y v_y + m_z v_z}{\sqrt{m_x^2 + m_y^2 + m_z^2}\sqrt{v_x^2 + v_y^2 + v_z^2}} > cos(\alpha) \quad (6)$$

In the system I, equation 3 restricts the point p to being within the sphere of proper radius by using the euclidean distance, and equation 4 restricts the point p to being at the surface of the cone by using the dot product. In the system II, equation 5 restricts the point p to being at the surface of the sphere by using the euclidean distance, and equation 6 restricts the point p to being within the proper angle by using the dot product. In these equations, we substitute m_x, m_y, m_z with $p_x - a_x, p_y - a_y, p_z - a_z$. Again, we substitute p_x, p_y, p_z with $s_x + d_x t, s_y + d_y t, s_z + d_z t$. The equation in each system is reduced to a quadratic equation in term of t. The inequality in each system is used to verify solutions.

There are three possible cases of the number of solutions:
(1) If the systems has two solutions, it concludes that S intersects the spherical cone normally (Fig. 8,9).
(2) If the systems has one solution, it concludes that S is too short or S is tangent to the spherical cone's surface (Fig. 10,11).
(3) If the system has no solution, it means there is no intersection of S and the spherical cone.

F. findStartingCubes()

There are two significant cubes to be chosen as starting cubes containing (1) the top of the cone which is a cube containing an apex of the cone, and (2) the bottom of the cone which is a cube containing an intersection point of the perpendicular vector and the spherical surface. Since we know the coordinate of the apex, then the cube containing the apex is defined. To find the intersection of the perpendicular vector and the spherical surface, given $\overrightarrow{ap} = m_x\hat{\imath} + m_y\hat{\jmath} + m_z\hat{k}$, then we need to find a point $p = (p_x, p_y, p_z)$ and such that

$$(a_x - p_x)^2 + (a_y - p_y)^2 + (a_z - p_z)^2 = l^2 \quad (7)$$

$$\frac{m_x v_x + m_y v_y + m_z v_z}{\sqrt{m_x^2 + m_y^2 + m_z^2}\sqrt{v_x^2 + v_y^2 + v_z^2}} = 1 \quad (8)$$

The cube containing the intersection of the perpendicular vector and the spherical cone is a cube containing p.

Equation 7 restricts the point p to being at the surface of the sphere by using the euclidean distance, and equation 8 restricts the point p to being on the perpendicular vector by using the dot product.

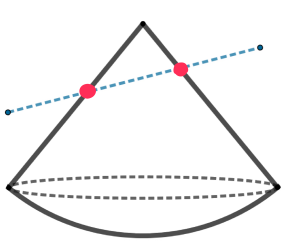


Fig. 8.

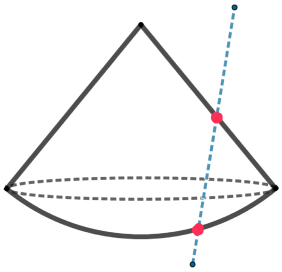


Fig. 9.

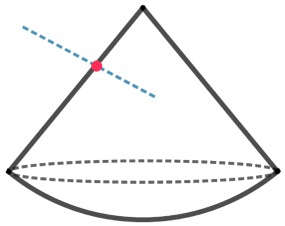


Fig. 10. **s is too short**

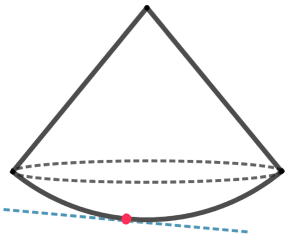
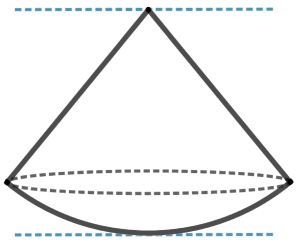


Fig. 11. **s is tangent**

G. GetInternalBoundingBoxVertices()

Define the bounding box $B = (l_x, l_y, l_z, h_x, h_y, h_z)$ as the region within the interval $[l_x, h_x]$ on the x-axis, within the interval $[l_y, h_y]$ on the y-axis, and within the interval $[l_z, h_z]$ on the z-axis.

Since a spherical cone is a subset of a sphere, B can be roughly approximated as the bounding box of the sphere which is $(a_x - r, a_y - r, a_z - r, a_x + r, a_y + r, a_z + r)$ when r is the radius of the sphere. Unfortunately, this inefficient approximation will dramatically increase the running time of the marching cube algorithm, especially when the angle of the spherical cone is small. However, if we move a plane along its axis, we will see that the highest point and the lowest point at which the plane intersects the spherical cone are the tangent points of the plane and the spherical cone. Thus, if we determine these tangential points, we will produce a more exact bounding box.



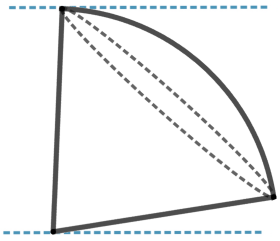


Fig. 12. **tangent at apex and sphere** Fig. 13. **tangent at apex and base**

There are five possible tangent points: (1) the apex (Fig. 12,13), (2) the highest point of the base (Fig. 13), (3) the lowest point of the base, (4) the highest point of the sphere, (5) the lowest point of the sphere (Fig. 12). The highest point

and the lowest point of these points to the associated axis will be used to define the bounding box. The apex, the highest point and the lowest point of the sphere are known, but the highest point and the lowest point of the sphere will be not considered if it is not in the spherical cone. Next we have to find the highest point and the lowest point of the base. Since $p = (p_x, p_y, p_z)$ is on the circumference of the base if the angle between $\overrightarrow{ap}$ and $\overrightarrow{v}$ is equal α and the euclidean distance from a to p is equal l. Suppose $\overrightarrow{ap} = m_x\hat{\imath} + m_y\hat{\jmath} + m_z\hat{k}$. Thus, p is on the circumference of the base if p satisfies two equations.

$$(a_x - p_x)^2 + (a_y - p_y)^2 + (a_z - p_z)^2 = l^2 \tag{9}$$

$$\frac{m_x v_x + m_y v_y + m_z v_z}{\sqrt{m_x^2 + m_y^2 + m_z^2}\sqrt{v_x^2 + v_y^2 + v_z^2}} = cos(\alpha) \tag{10}$$

Equation 9 restricts the point p to being at the surface of the sphere by using the euclidean distance, and equation 10 restricts the point p to being at the surface of the cone by using the dot product. In these equations, we substitute m_x, m_y, m_z with $p_x - a_x, p_y - a_y, p_z - a_z$. To find the highest point and the lowest point of x-axis, we have to modify these equations into two quadratic equations in term of p_x. Since the highest point and the lowest point of the base are the tangential points of the associated axis to the base, then each quadratic equation in term of p_x is supposed to have exactly one solution or repeated roots. Thus, the discriminant of both quadratic equations are zero and then we will get another two quadratic equations in term of p_y, p_z. We solve p_y, p_z and then plug them back to equation 9 or equation 10 to find the value of p_x. To find the highest point and the lowest point of y, z-axis, we repeat the same idea.

For each axis, there are two possible cases of the number of solutions.
(1) If the system has two solutions, it concludes that the plane is tangent to the base at two points. The greater solution refers to the highest point of the base, and the smaller solution refers to the lowest point of the base (Fig. 14).
(2) If the system has one solution, it concludes that the base is parallel to the plane. The highest point and the lowest point are the same (Fig. 15).

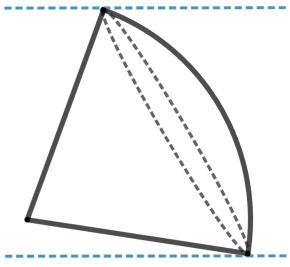


Fig. 14. **system has two solutions**

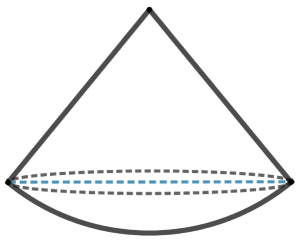


Fig. 15. **system has one solution**

H. Dataset Construction

We tested our method on two well studied protein-protein interfaces. These were the barnase-barstar complex (pdb: 1brs, chains A,D) and the RAP-RAF complex (pdb: 1c1y, chains A,B). We selected these complexes because extensive mutational studies reveal the effect of mutation on bonds and affinity on these complexes.

To prepare these structures, we first removed all atoms unrelated to the chains in our datasets. Next, we removed all hydrogens, and then replaced the hydrogens using the reduce module from MolProbity [6], with the -build option turned on. Intermolecular hydrogen bond donors and acceptors were then identified using distance and angle thresholds from HBPlus [15]. Spherical cones were then generated with apexes centered on donor atoms and acceptor antecedant atoms, with perpendicular vectors oriented towards the donor hydrogen and the acceptor atom, respectively.

I. Implementation Details

CSG operations were performed on Sol, a 34 node computing cluster with 780 2.3Ghz intel processor cores and an average of 5.5 GB of system memory per core. pClay is implemented in C++ and parallelized using Intel Threading Building Blocks. The average runtime for determining and computing CSG intersections (using the bounding box) between all 30,000 possible pairwise intersections of spherical cones with angle 90, slant height 23, resolution 0.25 was 341 seconds.

Intersection volumes of CSG intersections were computed using the Surveyor's Formula, which we described earlier [5]. This method accurately measures the volume inside a closed triangular mesh, which is the output of all CSG operations described here.

The defined ranges make it impossible for an atom to interfere with the hydrogen bond, so it is not necessary to check for interfering atoms within the intersection of the two cones.

III. Results

To evaluate how accurately the intersection of spherical cones can predict the presence of hydrogen bonds and their impact on binding, we computed the intersection of cone pairs generated from complementing donors and acceptors in each complex in the dataset. First, we evaluated how accurately a range of cone parameters could predict intermolecular hydrogen bond formation. Second, we measured the relationship between cone intersection volume and contribution to binding affinity.

A. Intersections between Spherical Cones

To assess the accuracy of spherical cone intersections for predicting the presence of intermolecular hydrogen bonds, we generated all donor and acceptor cones in our dataset at a range of parameters. Since hydrogen bonds have been found to exist between 2.3 Å and approximately 4.5 Å , we generated all cones with slant height ranging from 2.3 Å to

1.1 Å in increments of .3 Å. This range fully encompasses the range of expected intermolecular bonds, because, at the longest slant heights, hydrogen bonds that occur between the most distant donors and acceptors can be detected, and at the shortest slant heights, only the closest hydrogen bonds will be detected. We also generated all cones with a range of angle parameters. Since hydrogen bonds have been found to exist with bond angles approaching 90 degrees between the donor-donor hydrogen vector and the acceptor-acceptor antecedent vector, we generated cones with angles from 22.5° to 90°, in increments of 22.5. We then computed the CSG intersection between all intermolecular donor-acceptor cone pairs with the same radius and angle measurements, and measured the volume of the resulting region.

TABLE I
AVERAGE INTERSECTION VOLUME OF SPHERICAL CONES AND NUMBER OF INTERSECTING SPHERICAL CONES FOUND AT A RANGE OF VOLUMES.

Cones with Angle 90, slant height 2.3Å	1brs	1c1y
average cone volume (Å^3)	4.66	4.83
cones with volume >10.0	11	7
5 <volume <= 10.0	3	8
0 <volume <= 5	29	24

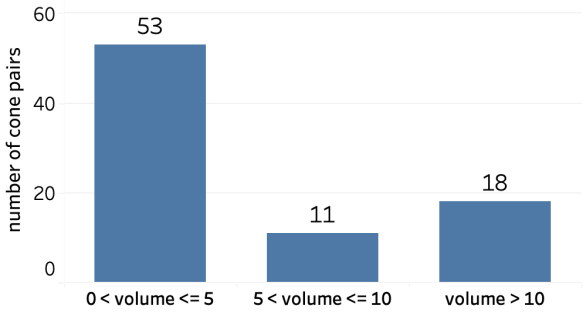


Fig. 16. number of cone pairs, generated with angle 90° and slant height 2.3Å, grouped by volume

Table I illustrates that cones generated at the largest volumes, with angle 90 and slant height 2.3, produced cone intersection volumes averaged between 3.53 and 4.66 Å^3. However, substantial and non-monomodal variations could be observed in cone intersection volumes. 53 out of 82 intersecting pairs had intersection volumes less than 5 Å^3, but 18 had intersection volumes above 10.0 Å^3 (Fig. 16). While our selection of volume thresholds is arbitrary, it demonstrates that, at least on this small dataset, there is a considerable subset of cone intersections with large volumes relative to the other intersections.

B. Predicting Hydrogen Bonds

We hypothesized that intersecting pairs of donor and acceptor cones generated at 90 degrees, with slant height 2.3Å and intersection volume greater than 10.0 Å^3 would correspond to actual hydrogen bonds. We identified the atoms and amino acids that supply the hydrogen bond donor and acceptor

barnase-barstar (1brs)

slant height	90°	67.5°	45°	22.5°
2.3Å	10	10	10	9
2.0Å	10	10	10	9
1.7Å	10	10	10	6
1.4Å	10	10	10	3
1.1Å	6	6	5	0

rap-raf (1c1y)

slant height	90°	67.5°	45°	22.5°
2.3Å	7	7	7	6
2.0Å	7	7	7	5
1.7Å	7	7	7	3
1.4Å	7	7	7	1
1.1Å	4	4	3	0

TABLE II
PREDICTED HYDROGEN BONDS THAT ARE VERIFIED IN THE LITERATURE, PREDICTED WITH A RANGE OF CONE PARAMETERS.

and verified whether these bonds are known to exist in the structural literature. 11 cone pairs in the barnase-barstar complex and 7 cone pairs in the rap-raf complex fulfilled these properties.

All 18 cone pairs, except one in the barnase-barstar complex, correctly identified hydrogen bonds established in the literature, demonstrating a positive predictive value (PPV) of 94.4% (true positives / positive predictions). Unfortunately, if the literature does not mention a hydrogen bond, this does not mean that a hydrogen bond does not exist, so false positives and true negatives cannot be assessed. Due to the large number of intersections with volumes less then 10.0 Å^3, we did not count false negatives, where small intersections nonetheless correspond to a bond in the literature.

We also considered whether the same bonds were predicted by intersecting cones with more limited parameters, expanding our study to cones with angle from 90 to 22.5 degrees and slant height from 2.3 Å to 1.1 Å. When cones were generated with smaller angles and/or smaller slant heights, the number of hydrogen bonds that were correctly predicted dropped only subtly, except at the smallest cone angles, the smallest slant heights, or combinations of both (Table II).

C. Predicting Hydrogen Bond Contributions to Affinity

We hypothesize that beyond simply detecting the presence of hydrogen bonds based on their angle and distance constraints, that intersections between spherical cones could estimate bond strength. Specifically, larger cone intersections correspond to margins within bond angle and bond length limits that might tolerate more motion in donor and acceptor atoms than cone pairs that have smaller intersections. These tolerances may of course be mitigated by other flexibilities in sidechain and backbone conformation.

Using cones generated with angle 90 and slant height 2.3 Å, we plotted the intersection volumes of all donor-acceptor pairs with greater than 10Å^3 against the experimentally reported effect on binding affinity caused by mutating one or both sides of the donor-acceptor pair. Affinity measurements from mutants where both amino acids were changed to alanine were preferred over other mutations, such as charge flipping

mutations, and mutations of the amino acids supplying both the donor and the acceptor were preferred over mutations of only the donor or only the acceptor.

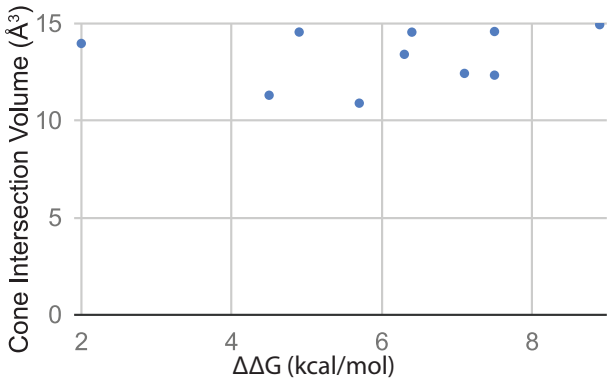


Fig. 17. A plot of cone intersection volume versus the change in Gibbs free energy (ΔΔG, kcal/mol) induced by mutating one or both sides of a hydrogen bond. Blue circles indicate a hydrogen bond donor-acceptor pair that was mutated to remove one or both sides of the hydrogen bond.

Of the two complexes in our dataset, barnase-barstar was the subject of a greater number of mutational studies. Every high intersection cone pair in this complex, except one, had been considered in mutational studies, four by mutating one side and six by mutating both sides of the complex. All mutations were to alanine, ensuring that no new hydrogen bond was formed by the side chain of the same amino acid. The relationship between cone intersection volume and mutational effect on binding affinity, measured in ΔΔG, is shown in Figure 17. On this complex, there is a limited relationship between intersection volume and ΔΔG, suggesting that larger intersection volumes are weakly related to stronger single-bond contributions to affinity.

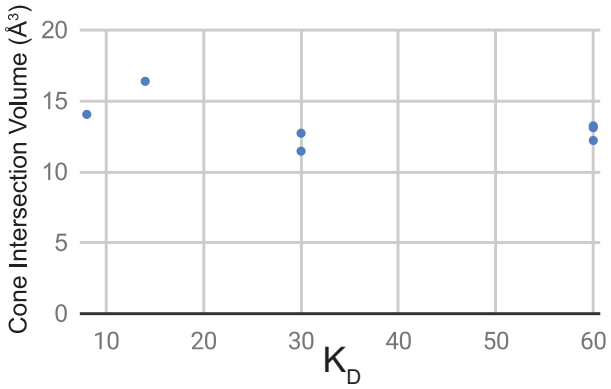


Fig. 18. A plot of cone intersection volume versus the dissociation constant K_D induced by mutation. Blue circles indicate a hydrogen bond donor-acceptor pair that was mutated on the RAF side of the hydrogen bonds to prevent hydrogen bond formation.

Studies of the RAP-RAF complex report binding affinities from mutations that removed hydrogen bond donors and acceptors from RAF, but not from RAP. Four of the mutations performed were to alanine, and the remaining three were to leucine, ensuring that the modified RAF sidechains did not form a hydrogen bond. However, since RAP amino acids were not modified, it may be the case that these amino acids form alternative hydrogen bonds with other amino acids of RAF. In these experiments, affinity was reported in terms of K_D. Three mutations involving R89L from RAF impaired binding so much that it could not be measured. The other three mutations showed a weakly negative relationship between increasing cone intersection volume and K_D (Fig. 18), suggesting that larger intersection volumes are weakly related to decreases in contributions to affinity.

IV. Discussion

We have presented a novel solid geometric representation of hydrogen bonds as spherical cones, inspired by the distance and angle constraints on hydrogen bond formation. This representation is the first to enable the comparison of hydrogen bonding capabilities based on bond formation tolerances, even if only one half of the bond is considered. CSG intersections could observe, for example, that two aligned protein structures reveal hydrogen bond donors in similar orientations, even if a partner molecule is not present. These capabilities point to the capacity to detect patterns of hydrogen bonding that are conserved among proteins with different folds.

In this study, we first evaluated whether the spherical cone representation would be effective for predicting the presence of hydrogen bonds in protein-protein complexes. Using two complexes that have been extensively examined in the structural literature for the presence and function of hydrogen bonds, we showed that 94.4% of spherical cones defined with large angle and slant height parameters that also had large intersection volumes corresponded to hydrogen bonds documented in the structural literature. Indeed, spherical cones are quite sensitive for this application, because even when they are generated with lower angles and slant heights, the same bonds are still identified.

We also evaluated whether or not the intersection volume between a donor cone and an acceptor cone would reflect the contribution of the cone to protein-protein binding affinity. Since the cones themselves represent the angle and distance range at which the hydrogen bond can form, the intersection is in some ways related to the geometric limits that the bond can be subjected to without breaking hydrogen bond limits. Citing the extensive mutational studies performed on our dataset proteins, we did not observe a strong relationship between cone intersection volume and reductions in binding affinity upon mutating the amino acids associated with the cones.

Overall, the spherical cone representation creates a new capability to detect hydrogen bonds as well as similarities in hydrogen bond compatibility. This capability points to new applications in predicting mutations that remove hydrogen bonds and in identifying similarity in hydrogen bonding patterns even when binding partners are unknown.

ACKNOWLEDGEMENTS

This work was funded in part by NIH Grant R01GM123131 to Brian Y. Chen.

REFERENCES

[1] T Andrew Binkowski, Patrick Freeman, and Jie Liang. pvsoar: detecting similar surface patterns of pocket and void surfaces of amino acid residues on proteins. *Nucleic acids research*, 32(suppl_2):W555–W558, 2004.

[2] Brian Y Chen. Vasp-e: Specificity annotation with a volumetric analysis of electrostatic isopotentials. *PLoS computational biology*, 10(8):e1003792, 2014.

[3] Brian Y Chen, Drew H Bryant, Viacheslav Y Fofanov, David M Kristensen, Amanda E Cruess, Marek Kimmel, Olivier Lichtarge, and Lydia E Kavraki. Cavity-aware motifs reduce false positives in protein function prediction. In *Computational Systems Bioinformatics*, pages 311–323. World Scientific, 2006.

[4] Brian Y Chen, Viacheslav Y Fofanov, Drew H Bryant, Bradley D Dodson, David M Kristensen, Andreas M Lisewski, Marek Kimmel, Olivier Lichtarge, and Lydia E Kavraki. The mash pipeline for protein function prediction and an algorithm for the geometric refinement of 3d motifs. *J. Comput. Biol.*, 14(6):791–816, 2007.

[5] Brian Y Chen and Barry Honig. Vasp: a volumetric analysis of surface properties yields insights into protein-ligand binding specificity. *PLoS computational biology*, 6(8):e1000881, 2010.

[6] Vincent B Chen, W Bryan Arendall, Jeffrey J Headd, Daniel A Keedy, Robert M Immormino, Gary J Kapral, Laura W Murray, Jane S Richardson, and David C Richardson. Molprobity: all-atom structure validation for macromolecular crystallography. *Acta Crystallographica Section D: Biological Crystallography*, 66(1):12–21, 2010.

[7] Georgi D Georgiev, Kevin F Dodd, and Brian Y Chen. pclay: A precise parallel algorithm for comparing molecular surfaces. In *19th International Workshop on Algorithms in Bioinformatics (WABI 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

[8] Georgi D Georgiev, Kevin F Dodd, and Brian Y Chen. Precise parallel volumetric comparison of molecular surfaces and electrostatic isopotentials. *Algorithms for Molecular Biology*, 15:1–20, 2020.

[9] Vladimir Gligorijević, P Douglas Renfrew, Tomasz Kosciolek, Julia Koehler Leman, Daniel Berenberg, Tommi Vatanen, Chris Chandler, Bryn C Taylor, Ian M Fisk, Hera Vlamakis, Ramnik J. Xavier, Rob Knight, Kyunghyun Cho, and Richard Bonneau. Structure-based protein function prediction using graph convolutional networks. *Nature communications*, 12(1):1–14, 2021.

[10] Chittibabu Guda, Sifang Lu, Eric D Scheeff, Philip E Bourne, and Ilya N Shindyalov. Ce-mc: a multiple protein structure alignment server. *Nucleic acids research*, 32(suppl_2):W100–W103, 2004.

[11] Kengo Kinoshita and Haruki Nakamura. Identification of protein biochemical functions by similarity search using the molecular surface database ef-site. *Protein Science*, 12(8):1589–1595, 2003.

[12] S Sri Krishna and Nick V Grishin. Structural drift: a possible path to protein fold change. *Bioinformatics*, 21(8):1308–1310, 2005.

[13] Daniel Kuhn, Nils Weskamp, Stefan Schmitt, Eyke Hüllermeier, and Gerhard Klebe. From the similarity analysis of protein cavities to the functional classification of protein families using cavbase. *Journal of molecular biology*, 359(4):1023–1044, 2006.

[14] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987.

[15] Ian K McDonald and Janet M Thornton. Satisfying hydrogen bonding potential in proteins. *Journal of molecular biology*, 238(5):777–793, 1994.

[16] Mark Moll, Drew H Bryant, and Lydia E Kavraki. The labelhash algorithm for substructure matching. *BMC bioinformatics*, 11(1):1–15, 2010.

[17] Hunter Read, Dylan Carpenter, Sam Herr, and Filip Jagodzinski. Petra: Protein-ligand complex engineering through rigidity analysis. In *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, pages 568–573, 2019.

[19] Michael Siderius and Filip Jagodzinski. Mutation sensitivity maps: Identifying residue substitutions that impact protein structure via a rigidity analysis in silico mutation approach. *Journal of Computational Biology*, 25(1):89–102, 2018.

[20] Alexander Stark, Shamil Sunyaev, and Robert B Russell. A model for statistical significance of local similarities in structure. *Journal of molecular biology*, 326(5):1307–1316, 2003.

[21] Andrew C Wallace, Neera Borkakoti, and Janet M Thornton. Tess: a geometric hashing algorithm for deriving 3d coordinate templates for searching structural databases. application to enzyme active sites. *Protein science*, 6(11):2308–2323, 1997.

SUPPLEMENTARY MATERIALS

TABLE III
HYDROGEN BOND DONORS AND ACCEPTORS IDENTIFIED FROM THE BARNASE-BARSTAR COMPLEX (PDB: 1BRS) WITH INTERSECTION VOLUME GREATER THAN 10 Å^3

Donor Atom	Donor Res.	Donor Res. #	Donor Chain	Accept. Atom	Accept. Res.	Accept. Res. #	Accept. Chain	Intersec. Vol.
NZ	LYS	27	A	OG1	THR	42	D	10.900755
N	ARG	59	A	OD1	ASP	35	D	13.41674
NH2	ARG	59	A	OE1	GLU	76	D	14.56397
N	LEU	34	D	OE2	GLU	60	A	13.97747
N	GLU	60	A	OD2	ASP	35	D	11.309484
NH2	ARG	83	A	OD1	ASP	33	D	14.559535
NH2	ARG	83	A	O	GLY	43	D	11.063205
NH2	ARG	87	A	OD2	ASP	39	D	12.435039
N	GLY	31	D	ND1	HIS	102	A	12.344065
ND2	ASN	33	D	O	HIS	102	A	14.592471
NE2	HIS	102	A	OD2	ASP	39	D	14.944932

TABLE IV
HYDROGEN BOND DONORS AND ACCEPTORS IDENTIFIED FROM THE RAP.GMPPNP-CRAF1 COMPLEX (PDB: 1C1Y) WITH INTERSECTION VOLUME GREATER THAN 10 Å^3

Donor Atom	Donor Res.	Donor Res. #	Donor Chain	Accept. Atom	Accept. Res.	Accept. Res. #	Accept. Chain	Intersec. Vol.
NH1	ARG	59	B	OE1	GLU	37	A	12.728163
NE	ARG	59	B	OE2	GLU	37	A	11.462316
N	VAL	69	B	O	GLU	37	A	14.056369
NH2	ARG	89	B	OD1	ASP	38	A	13.115738
N	SER	39	A	O	ARG	67	B	16.388207
NE	ARG	89	B	O	SER	39	A	12.217045
NH2	ARG	89	B	O	SER	39	A	13.24629