Travel Depth, a New Shape Descriptor for Macromolecules: Application to Ligand Binding

Ryan G. Coleman and Kim A. Sharp*

Depth is a term frequently applied to the shape and surface of macromolecules, describing for example the grooves in DNA, the shape of an enzyme active site, or the binding site for a small molecule in a protein. Yet depth is a difficult property to define rigorously in a macromolecule, and few computational tools exist to quantify this notion, to visualize it, or analyze the results. We present our notion of travel depth, simply put the physical distance a solvent molecule would have to travel from a surface point to a suitably defined reference surface. To define the reference surface, we use the limiting form of the molecular surface with increasing probe size: the convex hull. We then present a fast, robust approximation algorithm to compute travel depth to every surface point. The travel depth is useful because it works for pockets of any size and complexity. It also works for two interesting special cases. First, it works on the grooves in DNA, which are unbounded in one direction. Second, it works on the case of tunnels, that is pockets that have no "bottom", but go through the entire macromolecule. Our algorithm makes it straightforward to quantify discussions of depth when analyzing structures. High-throughput analysis of macromolecule depth is also enabled by our algorithm. This is demonstrated by analyzing a database of protein-small molecule binding pockets, and the distribution of bound magnesium ions in RNA structures. These analyses show significant, but subtle effects of depth on ligand binding localization and strength.

Introduction

Depth is a term frequently applied to the shape and surface of macromolecules. For example, enzyme active sites are routinely described as shallow or deep. Small ligand binding sites on proteins are also frequently described in terms of depth. Depth is just one facet of the property "binding pocket shape" one would like to define quantitatively, to aid for example, in screening a large library of potential ligands, or in docking of a candidate ligand. Groove depth is one of the fundamental terms used to describe the differences in structure of the A, B and Z forms of DNA. In spite of the common use of the term depth, it is a surprisingly difficult property to define rigorously in a macromolecule. Discussions of depth in the literature, although intuitively reasonable, are usually qualitative. The concept of depth is thus difficult to subject to rigorous analysis or to extract the most information from. A large part of the difficulty in analyzing depth is due to the complexity and range of shapes adopted by macromolecules. Protein surfaces are fractal in nature, adding to the difficulty. To illustrate some of the difficulties, consider first the issue of a reference point or level. In geodesy, mountain peaks and ocean depths are referenced to the mean sea level, providing a standard reference level (although not without regional difficulties: mean sea level either side of the Panamanian isthmus differs considerably, for example). There is no equivalent to mean sea level in a molecule. Second, consider the case of deep pockets involving overhangs or that re-approach the molecule surface at some point away from their origin. Euclidean distance of the bottom of the pocket to the nearest surface, while easy to define and compute, will be a very misleading and grossly underestimating measure of depth. These difficulties are reflected...
in the fact that there are few computational tools to quantify the concept of depth, to visualize it, or analyze the results. To address this problem, we present here our notion of travel depth, simply put the physical distance a solvent molecule would have to travel from a surface point to a suitably defined reference surface. The concept of travel depth was designed to avoid the “short circuiting” error described above, and also to solve the problem of a reference level. We first define the concept of travel depth, and the reference level used by it, then present a fast, robust approximation algorithm to compute travel depth to every surface point. Selected examples using very different molecular shapes are used to demonstrate that our definition of depth works for special cases, and that it conforms to our intuition, so confirming that we have introduced a “good” definition for depth and that our approximate numerical implementation of it is reasonable. We then describe some applications of our algorithm, including a high throughput application to a small molecule binding database.

**Theory and Algorithm**

**Definition of travel depth**

Any measure of depth must start with the questions: Depth of what, and from what? Here, we are concerned with the depth of any point on the molecule’s surface. Two definitions of surface predominate for macromolecules, the solvent accessible surface, and the molecular surface. In both cases a crucial parameter is the probe radius, which is almost universally taken to be that of water (usually values between 1.4 Å and 1.8 Å are used). Many algorithms exist for computing these idealized surfaces. Most, but not all, produce a triangulated form of the surface, primarily for display using standard computer graphic routines. Our algorithm assumes a simple closed triangulated surface. The surface must be orientable and connected, though these are not strong requirements; the latter disallows only cavities. For the broadest applicability of our method, we make no other assumption about how the surface was produced, or what it should look like. In practice we use the molecular surface as generated by the algorithm in the GRASP macromolecular graphics program implemented as a stand-alone program using a probe radius of 1.8 Å and standard atomic radii. Though we test only this surface generation scheme and the resulting triangulated surfaces, our definition and algorithm generalize to any triangulated surface generation scheme.

Our definition of travel depth is that for each point on the molecular surface, the travel depth is the minimum distance a solvent probe would have to travel through the solvent from that surface point to get to the reference level. A natural and parameter-independent reference level is provided by the convex hull of the molecular surface. The convex hull is a standard construct in computational geometry. In three dimensions, the convex hull is the smallest volume convex polyhedron that contains all the surface points. In terms of molecular surfaces, the convex hull is equivalent to the molecular surface produced by an infinite solvent probe radius. Algorithms and code for convex hull computation have been well studied and are fast and reliable.

The next step is to compute the minimal distance from every surface point to the convex hull while respecting the boundary of the molecular surface. In other words, the travel path along which the distance is computed must lie outside the molecular surface in the solvent. We note that computing such a minimal distance between two points while avoiding obstacles is exactly the shortest path planning problem commonly encountered in robotics, and that an exact solution to the problem is NP-hard. Our solution, described below, is to approximate this minimal distance in such a way that it was easy to code and run in a short time so that we could establish what the depth measure would look like on real examples, and whether it would be useful in structural analysis.

**Calculation of travel depth: preprocessing**

The first step is to remove cavities, defined as completely enclosed solvent pockets in the molecular surface. The triangles that represent these cavities are removed from the surface and are not used in later calculations. Since there is no way for the solvent probe to travel from a closed cavity surface to the convex hull without passing through the macromolecule itself, travel depth does not apply to these surfaces. We note, though, that simple Euclidean distance to the nearest part of the external molecular surface would provide a satisfactory definition of the minimum depth of a closed cavity.

Two important pre-processing steps are done at this stage. First, the longest edge of any triangle in the surface is found and the length saved for later. Also, all the points on the surface are put into a two-dimensional orthogonal range search tree structure oriented along one grid axis. This helps improve the running time, as described later, but it is nonessential to the algorithm.

**Calculation of travel depth: mapping onto grid**

The macromolecule and a region of the surrounding solvent are embedded in a cubic grid of dimensions $K \times L \times M$. For convenience, the grid extends to one grid cube beyond the minimum and maximum coordinate of the molecular surface in each orthogonal direction, so that the border is completely outside the surface. The default grid spacing used in our algorithm is 1 Å, however the algorithm and code generalize to any spacing. The only consideration is for the spacing to be small enough to approximate well the topology of the
given molecular surface. For instance, when a probe radius $P=1.8$ Å is used, as in our surfaces, the maximum concavity of any section of the molecular surface is limited to that of the probe radius. From this, a maximum allowable grid spacing, $G$, can be calculated from the formula:

$$G = 2P/\sqrt{3}$$  \hspace{1cm} (1)

This grid spacing ensures that any concave depression in the surface is represented by at least one grid center. Using the same formula with the smallest atom radius used to construct the molecular surface leads to another bound on the grid spacing to guarantee that any convex protrusion is represented by at least one grid point. Again, 1 Å is well within this limit for most commonly used radii for heavy atoms (the smallest such atom commonly found in biomolecules is oxygen, with a radius of 1.4 Å). This assumption ignores problems caused by a very coarse surface, though this assumption is relaxed and a solution to problems caused by this in our algorithm are discussed later.

The next step of the algorithm is to find the convex hull of the molecular surface. There are many $O(n \log n)$ algorithms for computing the convex hull in three dimensions. We use available code from Qhull or Quickhull, an optimized and robust package.\textsuperscript{13}

**Calculation of travel depth: classifying grid points**

After construction of the convex hull each point lying at the center of each grid cube must be checked to see whether it lies inside or outside the convex hull and inside or outside the molecular surface. The convex hull can be represented as a list of outward facing triangles. A sufficient check for being outside the convex hull is to check the point against each triangle and surface normal to see which side it is on. A point that is outside any convex hull triangle is outside the entire convex hull. Doing this check for each point in the grid is sufficient to determine which points are outside the convex hull and which are inside. Next, points inside the convex hull are assigned to either the outside or the inside of the molecular surface. This step is the most time consuming portion of the entire algorithm. The problem is that determining whether a point is inside or outside a general triangulated surface requires global information. It is not sufficient to check a point against every surface triangle. However, an appropriate geometric property can be used to solve this problem quite efficiently: any line drawn completely through the molecular surface will intersect an even number of triangles. Lines are constructed in one orthogonal direction of the grid such that they each pass through a set of grid points. Moving from grid point to grid point along this line from one side of the grid until the first triangle is met assigns all those points to the outside. Each time a triangle is encountered, the inside/outside assignment switches. This procedure is continued until the opposite side of the grid is reached. In this manner, when a complete set of lines through the grid in one direction have been processed, the correct assignment has been made for all the points. In practice, since a line of grid points is used in this step, all their inside or outside checks can be done at once. Each triangle from the surface can be checked to see if it intersects this line, and to find the point of intersection if it exists. After this, the previously described procedure can be used to determine on which side of the surface each point on that line lies.

Naively, each triangle could be tested against each line. However, a more efficient procedure, which drastically cuts down the number of intersection checks, uses both of the preprocessing steps mentioned earlier. After picking a dimension along which the lines will be constructed, the other two dimensions are chosen as the orthogonal directions to construct a two-dimensional orthogonal range search tree from all the surface points. This polynomial time construction allows queries that consist of any orthogonal, or grid-aligned, rectangle which return all the points in that area.\textsuperscript{14} This is used in conjunction with the precomputed longest edge length, to quickly find all triangles that possibly intersect the line of interest, by querying the square centered around the line’s axis plus and minus the longest edge length. Only triangles that have all three points in this square possibly intersect the line of interest, and this test quickly reduces the number of triangle intersection checks that must be done. Though these checks each take constant time, they can be very slow, as they involve evaluating several matrix determinants.

To unambiguously determine inside and outside, our algorithm assumes that these lines will not intersect a triangle across its face, or through a single vertex. These special cases, if they occur, are easy to detect and the points can be slightly perturbed until the ambiguity no longer occurs.

At this point each grid cube has been classified into one of four categories based on the location of its center and whether it contains any molecular surface points. Either outside the convex hull (class O), between the convex hull and molecular surface (class B), inside the molecular surface but containing a molecular surface point (class S) or finally inside the molecular surface but containing no molecular surface points (class I). A small example is shown in the left panel of Figure 1. Class I cubes are ignored in the rest of this work, as no depth needs to be calculated for them.

**Calculation of travel depth: assignment of travel depth to grid points**

It remains to approximate the minimum distance that a probe sphere would need to travel to get from each surface point to the convex hull. This travel depth is assigned to class B and S points recursively, as follows. All grid cubes of class O are assigned a travel depth of zero. All cubes of class B and S are initially assigned an unreachablely large value, e.g.
that have endpoints in the two grid cubes, neighbors defined by edges of the molecular surface. Additionally, cubes of class S can have additional neighbors defined by edges of the molecular surface that have endpoints in the two grid cubes, \( i \) and \( j \). Their distance is also the Euclidean distance \( d(i, j) \) between the grid cube centers, 1, \( \sqrt{2} \), and \( \sqrt{3} \) grid units, respectively.

The distance to these three types of adjacent cube are the Euclidean distances between cube centers, 1, \( \sqrt{2} \), and \( \sqrt{3} \) grid units, respectively. Additionally, cubes of class S can have additional neighbors defined by edges of the molecular surface and the convex hull as well as the grid cubes \( I \) that contain surface points. Although the travel depth assignment of points between the convex hull and molecular surface is not used in the applications of travel depth described here, it is a property that may prove useful in future applications like docking.

\[
d_i = \min(d_j + \text{dist}(i, j))
\]

where \( j \) ranges over all neighbors of \( i \). This procedure is repeated until no new depth assignments are made.

A key requirement to correctly propagate depth with respect to the topology of the molecular surface is the appropriate definition of neighboring cubes in equation (2). For a class O or B cube, any of the 26 immediately adjacent cubes of class O, B or S are considered neighbors. Additionally molecular surface edges that have an endpoint in a class O or B cube and another endpoint in a class O, B, or S cube make those two cubes neighbors. For class S grid cubes, any adjacent cube of class O or B is a neighbor. However, for a class S cube only class S grid cubes that are connected to it by a molecular surface edge are considered neighbors, even if the two class S cubes are adjacent. There may be adjacent class S grid cubes that do not have a molecular surface edge between them, for example when two distant parts of the molecular surface approach each other very closely without meeting. It is important not to propagate the travel depth across this gap.

The neighbor distances in equation (2) are defined as follows: Each grid cube has six adjacent cubes that share one face, 12 adjacent cubes that share only an edge and eight adjacent cubes that share only a vertex. The distances to these three types of adjacent cube are the Euclidean distances between cube centers, 1, \( \sqrt{2} \), and \( \sqrt{3} \) grid units, respectively.

Starting from the class O grid cubes with depth 0, the neighboring grid cubes are assigned a depth according to equation (2), then the neighbors of the neighbors are assigned and so on. In this way the depth propagates in towards the molecular surface, and into the class S cubes, but it does not propagate through the macromolecule since the depth assignment is not propagated into class I cubes. This is illustrated in the right panel of Figure 1. After the assignment phase terminates, the depth is converted from grid units into a physical distance by multiplying by the grid spacing. This results in a calculation of the shortest paths from the class O cubes to all class B and class S cubes, given the neighbor and distance definitions above.

The depth assignment phase of the algorithm is speeded up by using Dijkstra’s algorithm for shortest paths on a graph and using available code that implements a key component of that algorithm, a priority heap. Dijkstra’s algorithm keeps track of the vertices in the graph (grid cubes) that have already been assigned a travel depth, and the shortest path from these assigned grid cubes to the rest of the grid cubes. The priority heap keeps track of the unassigned grid cubes that can be assigned a travel depth, and efficiently updates and finds the current shortest travel depth grid cube that has yet to be processed. In practice, we use a priority heap that has reasonable amortized performance and was compatible with the rest of our code (Eppstein, D.†). At this stage, all that remains is to assign each surface point a depth based on the grid cube it is located in, resulting in a computed travel depth for each point on the surface. The travel depth is also computed for all the grid cubes B between the molecular surface and the convex hull as well as the grid cubes \( I \) that contain surface points. Although the travel depth assignment of points between the convex hull and molecular surface is not used in the applications of travel depth described here, it is a property that may prove useful in future applications like docking.

† http://aspn.activestate.com/ASPN/Cookbook/Python/Recipe/117228
Presentation of results

To visualize the results of our algorithm, we used the PyMOL package.17 The triangulated molecular surface can easily be read into this program, along with travel depth values, and a red-green-blue color gradient assigned to each point of the surface based on travel depth. Red represents a travel depth of zero, with increasing depth indicated as the color changes from green to blue. The depth represented by blue is set either to the maximum value for that molecule, or to a fixed value to compare of a set of molecules. Color values at each point along each edge and triangle are interpolated using the standard approach to produce a smooth visualization of depth.17 Further refinements, such as displaying only surface in a certain range of depth may be useful for particular applications, and are straightforward with our algorithm.

Robustness, errors and timing analysis

Depending on the size of the macromolecule and the resolution at which the molecular surface is generated, the input surface to the travel depth algorithm might be quite coarse. In this case, regions of the surface may not conform well to the estimates of maximum concavity. This may result in small crevices or tunnels that violate the maximum concavity assumption. These errors are accounted for by the molecular surface edges that define grid cube adjacencies. The only level of coarseness that may cause a problem is where two parts of the surface approach each other very closely, less than the grid spacing. In these cases, the travel depth would propagate between these surfaces when it should not. However, to violate this assumption requires a violation of the maximum convexity assumption, which corresponds to a severe underestimation of the size of an atom or adjacent atoms forming such a barrier.

There are two sources of error in our approximation algorithm, each of which can be reduced at the cost of increasing the running time of the algorithm. The first source of error comes from the grid orientation. The approximate distance can be overestimated if a significant part of the path traveled is diagonal with respect to the grid axes. The worst case is when the actual distance should be down two grid units and over one grid unit in both other directions, the path length here is $\sqrt{6}$, while the approximation gives one grid unit down and then one diagonal step of length $\sqrt{3}$. This type of error leads to an error factor at most $(1 + \sqrt{3})/\sqrt{6}$, or roughly 1.11 times the actual shortest path length. Rotating the grid axes and re-running the algorithm and taking the minimum computed in either orientation would reduce this error, although we found that for the applications described here it has not been necessary.

The second source of error lies in the discretization of the distance, again from the use of the grid cubes to approximate the distance. Using smaller grid cubes, at a cost of increasing the running time, can reduce this error. In practice, there is little reason to get an extremely accurate measure of this distance, as there are already sources of uncertainty regarding the travel depth property, and indeed in the molecular surface construct itself. It would be hard to argue that differences of some small travel depth distance like 1 Å had any real physical meaning.

Our algorithm has both a reasonable asymptotic running time when the complexity is analyzed, and a reasonable running time in practice. Also, following the philosophy of keeping the code as simple as possible, time spent coding and debugging was minimized, available pieces of code like PyMOL17 and a priority heap (Eppstein, D.‡) were used when possible.

We have highlighted the practical runtime issues throughout the description of the algorithm. The algorithm also has a reasonable running time when analyzed asymptotically.16 Without the orthogonal range search tree speedup mentioned, the running time is:

$$O(p \log p + c(d^3 + (t + d)d^2 + (d^3 + e)\log d^3)) \quad (3)$$

where $p$ is the number of points on the molecular surface, $c$ is the number of triangles on the convex hull, $t$ is the number of triangles on the molecular surface, $d$ is the number of grid cubes in any dimension, and $e$ is the number of edges, which is linear in terms of $t$ and $d^3$. The first term in equation (3) comes from the convex hull construction, the second term from the checks for each grid cube to see if it is inside or outside the convex hull. The third term comes from the checks to see if each grid cube is inside or outside the molecular surface. The fourth term is the cost of the propagation step using the shortest path algorithm and amortized time cost priority heap.

With the orthogonal range search tree speedup in place, there are two additional components to consider, the $O(t)$ steps to find the longest triangle edge, the $O(t \log t)$ steps to build the orthogonal range search tree (faster algorithms exist, but are harder to code14). The $O((t + d)d^2)$ term to check each grid cube becomes $O((\log^2(t) + k + d)d^2)$ step to do a range search query and then $k$ checks must be done, where $k$ is the number of triangles returned from the range search. Also, it should be noted as was later revealed by our timing analysis that the orthogonal range search idea should probably be applied to the inside/outside convex hull routine, changing the $O(cd^3)$ time into $O(c(\log c + (\log^2(c) + d)d^2))$ as the time for the convex hull checks now outweighs the time for the molecular surface checks as we have it implemented.

At the heart of this analysis is the fact that if we halve the grid spacing used, our algorithm gets

‡http://aspn.activestate.com/ASPN/Cookbook/Python/Recipe/117228
worse by a factor of 8, since there are twice as many grid cubes in each dimension. This is one reason grid distances smaller than 1 Å are never considered. Though they could be calculated they are impractical. Fortunately this analysis shows us that the overall speed of the slowest steps in practice, that is checking whether each grid cube is inside or outside the various surfaces, can be made to grow only with the squared logarithm of the number of triangles, plus the factor $k$ representing how many triangles are returned from an average range query. Though an initial penalty must be paid, this provides an overall faster approach as the number of triangles increases. This allows us to use very fine triangulated surfaces and still maintain reasonable runtimes, or use very coarse triangulated surfaces to get good exploratory results.

To give some estimate of the processing time involved, we provide the following timing analysis, conducted using one processor of a dual processor machine (Intel 2.4 GHz chip, 4797 BogoMIPS, 1 gigabyte RAM) running RedHat Linux 9.0. Different parts of the algorithm were timed separately. Two test Protein Data Bank (PDB) files were used, a representative small protein, cyclic bovine pancreatic trypsin inhibitor, PDB code 1K6U. To represent larger more complicated proteins, the six chain tic trypsin inhibitor, PDB code 1K6U. To represent representative small protein, cyclic bovine pancrea-

### Results

The first tests of the travel depth algorithm were designed to see if the definition conformed to one’s qualitative intuition about depth in macromolecules. In other words, is the definition of travel depth reasonable and useful? We used a variety of structures that had qualitatively different surface topographies. The first is duplex DNA, to which the term groove depth is commonly applied. We evaluated the depth of the major and minor grooves in $A$, $B$ and $Z$ canonical forms of DNA. 15 base-pairs of A-T were generated with the routine NUCGEN in canonical $A$ form, crystal structures 1BNA and 3ZNA were used for the $B$ and $Z$ forms, respectively. It should be noted that the structure 3ZNA was constructed by duplicating base-pairs present in the crystal to achieve the length shown, and is therefore considered a theoretical model in the PDB. Our travel depth algorithm gives intuitively reasonable results, shown in Figure 2. All surfaces are colored from red (travel depth 0 Å) to green (travel depth 7 Å), then finally to blue (travel depth 14 Å). It is clear that the major and minor grooves of the $B$-form are nearly the same depth, whereas the major groove of the $A$-form is much deeper than the minor groove of $A$-form or either groove of the $B$-form. Also, what would usually be the minor groove has turned into a very deep groove in the $Z$-form, and the major groove has almost no depth. We summarize these results quantitatively in Table 2. This is in good agreement with the standard description of these grooves. Specifically, “in $A$-DNA the helix axis passes by the major groove side of each base-pair, making that groove very deep, the minor groove shallow…” Also, B-DNA is described: “This means that major and minor grooves are of comparable depth…” Finally, Z-DNA is described: “With the helix axis passing down the minor groove, that groove is extremely deep, whereas the major-groove edge of each base-pair is pushed out to the perimeter of the helix, giving the groove zero depth”.

To further illustrate that our algorithm is intuitively correct, we show three other examples. First, a simple well-known pocket was analyzed, that of streptavidin bound to biotin (PDB code 1MK5). The result is shown in Figure 3. This clearly illustrates that travel depth can quantify a pocket near the surface. Next, a tunnel is shown in Figure 4 from the FAB fragment (PDB code 1A0Q). The travel depth algorithm works well in this case. Despite the fact the
tunnel has no bottom the middle of the tunnel is correctly identified as the deepest point. Also, the tunnel in Figure 4 is additionally characterized by the maximum distance for which a solid connected ring of surface points exists all the way around the tunnel, which is 18 Å. Finally, horseradish peroxidase (PDB code 1ATJ) is shown, which has a very deep active site. Figure 5 shows the result, which illustrates a case where a purely Euclidean distance algorithm would fail, as the deepest part of the pocket is closer to the other side of the protein than the one the substrate must enter from. A summary of various features on these previous six examples is shown in Table 2.

As an example of a high throughput data base application of the travel depth algorithm, we examined the small molecule binding structural database PDBbind. All 900 structures were used from the 2003 refined set. The proteins each bind a single small molecule ligand, and have binding data associated with the complex, as well as separate files for protein and ligand. Binding data for this set are from either the dissociation constant ($-\log K_d$) or competitive inhibitor concentration ($-\log K_i$), both referred to here for brevity as $-\log K$. This database was chosen over other available options because the structures and binding data had been hand checked and gathered from original sources, and the structure files were easily accessible, downloadable in modified, in clean form within one archive file. This allowed us to perform the analysis with only minor conversion of data formats, and no further editing or checking of input files. We note that 13 of these structures had ligands completely enclosed in cavities, inaccessible to solvent, and therefore only 887 structures were used whenever the ligand site was analyzed. The protein atom coordinates were used to construct the molecular surface at a medium setting of surface coarseness. We assume that sampling the travel depth at these surface points gives us an accurate and representative picture of the depth of the protein, or of a ligand binding site for instance. Under this assumption, averaging the travel depth across the surface points is an acceptable way to measure the overall travel depth of a protein, as is done later.

To test the hypothesis that ligands are in deeper pockets rather than shallower pockets, the protein surface points were divided into two classes, those near ligand atoms representing the binding site, and the rest. For each atom in the ligand, the single nearest surface point was found and included in the

![Table 2](image)

**Table 2.** Travel depths of selected macromolecular features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Major groove max. depth</th>
<th>Minor groove max. depth</th>
<th>Major groove average depth</th>
<th>Minor groove average depth</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-DNA</td>
<td>13.7</td>
<td>5.0</td>
<td>8.6</td>
<td>3.6</td>
</tr>
<tr>
<td>B-DNA</td>
<td>10.0</td>
<td>9.1</td>
<td>4.6</td>
<td>5.6</td>
</tr>
<tr>
<td>Z-DNA</td>
<td>4.2</td>
<td>10.8</td>
<td>2.0</td>
<td>6.5</td>
</tr>
<tr>
<td>Binding site average depth</td>
<td>Binding site max. depth</td>
<td>Deepest tunnel point</td>
<td>Ring around tunnel depth</td>
<td></td>
</tr>
<tr>
<td>Tunnel (1A0Q)</td>
<td>10.6</td>
<td>18.0</td>
<td>23.0</td>
<td>18.0</td>
</tr>
<tr>
<td>Horseradish peroxidase (1ATJ)</td>
<td>18.8</td>
<td>24.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Streptavidin-biotin (1MK5)</td>
<td>8.5</td>
<td>10.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Depths are in Å.
binding site if it was within an arbitrary threshold of 4 Å. This method gave a simple way of partitioning the surface into the binding site and the rest of the surface, erring on the side of including too few surface points in the binding site. The results are shown in Figure 6. In the Figure, for each protein the average depth of the binding site surface, $d_b$, is plotted against the average depth of non-binding site surface, $d_n$. The vast majority of the points lie above the $x = y$ line indicated on the Figure, demonstrating that the binding site is almost always a pocket, as expected.

Next, the distribution of depths of ligand binding sites was compared to the distribution of the overall surfaces, across the entire 887 complexes. For comparison, a small control dataset with proteins not known to bind any ligands was analyzed as well. We note that 1TGF was left out of the control dataset since it is no longer in the PDB. We removed the water molecules, ions and buffers found in these control structures for the analysis. The histograms showing the depths of the surface points in each category over the entire dataset are shown in Figure 7. The since the number of surface points in each set is so different, the data has been normalized so that the area under each curve is equal. The Figure shows that there is a clear but not complete preference for deeper points to be near a ligand binding site. Interestingly, the width of the histogram for proteins that bind a ligand is greater than that for the control, “non-binding” proteins. This indicates that binding proteins tend to have a rougher, or more corrugated surface. This raises the possibility that some proteins are intrinsically more “bindable” than others due to the kind of surface topography that they have.

Finally, to calculate the statistical significance that the ligands had some bias to be near deeper surface points, a permutation $p$-value test was conducted. For each protein, the complete set of surface points was assembled, including both the ligand binding site and the rest of the surface. From this set, a random selection of points equal in number to those in the ligand binding site was taken, and the average depth found. This random selection was repeated five million times. The $p$-value is the number of times this selection had greater than the average depth of the true ligand binding site $d_b$ divided by the number of random sets. With five million random permutations, the lower bound on the possible $p$-value is $2 \times 10^{-7}$. This test gives a good measure of whether the ligand bound to each protein is bound in a deep pocket more often than random. A more complete estimate would use all the potential ligand binding sites on the surface, and calculate the average depth for each. However, generating all the possible ligand binding sites is a rather complicated problem, one which is usually solved by only sampling some of the possible binding sites.

The complete results of the permutation tests on the PDBbind dataset are given as Supplementary Data, along with the average depth of the binding site. To summarize the results, 13 of 900 structures contained ligands that were completely enclosed in cavities, inaccessible to the outside solvent. Excluding those in cavities, 48 of 887 structures had a $p$-value greater than 0.05, so the remaining 839 structures had ligands that were in significantly deep pockets under this criteria. Under the strictest requirement tested, that of having a $p$-value less than $2 \times 10^{-7}$, 688 of 887 structures had ligands buried in these significantly deep pockets.

In Figure 8 we examine the relationship between protein size and average surface depth using the PDBbind data set. As a robust measure of protein...
size that could easily be computed for the entire data set we used the total number of heavy atoms. Assuming very similar packing densities for all proteins, number of heavy atoms should be proportional to protein volume. Depth data were plotted against the cube root of the number of heavy atoms since for a largely globular set of proteins this metric should scale well with the linear dimension of the protein. Indeed, for the mean surface depth averaged over the entire protein surface there is an excellent linear correlation ($R^2 = 0.84$). Thus average depth increases linearly with protein size. Not surprisingly, larger proteins can have deeper pockets, but for the average depth to increase with protein size larger proteins must also have more pockets of significant depth, i.e. be rougher. The scaling law indicates that average travel depth is an indicator of overall surface roughness, and a good reflection of the fractal nature of the protein surface, as was discovered previously by analysis of surface area.4

The fact that the fractal nature of the protein surface also emerges from a quite different analysis based on depth provides additional validation of the concept of travel depth. Looking at depth data from just the ligand binding sites in Figure 12, there is still some correlation with protein size, but the significantly smaller variance in travel depth is explained by protein size ($R^2 = 0.47$). This may include effects from the smaller amount of averaging involved in using a small subset of the protein surface.

A straightforward question to answer with the binding affinity data from the PDBbind dataset is whether binding affinity of the ligand ($-\log K$) correlates with the average travel depth at which the ligand is bound. A priori, one might expect deeper pockets to have greater affinity, based on the idea that a deeper pocket would make more interactions with the ligand. On the other hand, the amount of surface area that one could bury or interactions one could make when binding a small

---

Figure 4. An example tunnel color coded by travel depth. This is a FAB fragment from PDB code 1A0Q. The top view looks down on the tunnel, the bottom view is a side view that has been cutaway through the tunnel.
ligand is limited by the ligand size. For a given amount of surface burial or number of interactions, one might expect deeper pockets to be less favorable as the long range electrostatic desolvation penalty would be greater. In anticipation of these effects, we computed the change in solvent accessible surface area upon ligand binding, $\delta A$, for each of the 887 complexes in the database, in addition to travel depths. The change in surface area was obtained from the surface area of the entire complex minus that of the protein and the ligand alone, calculated using the program SURFCV. Binding affinity, buried surface area and travel depth data for the set of 887 complexes is given in the Supplementary Data. The average travel depth of the binding pocket, $d_b$, is plotted against the binding affinity in Figure 9 for the entire binding data set. A linear fit regression was conducted, and the $R^2$ values were very close to 0, indicating that

Figure 5. An example deep pocket color coded by travel depth. Two views of horseradish peroxidase, taken from PDB code 1ATJ. The bottom view is a cutaway showing one view of the pocket with a maximum depth near the ligand of 24 Å. A straight line Euclidean metric from the deepest point of this pocket would travel through the protein to the wrong side.

Figure 6. Average travel depth of entire protein surface plotted against average travel depth of just the binding site, $d_b$ for each structure in the PDBbind dataset. The $y=x$ line is shown on the Figure.
none of the variation in binding affinity can be explained by the depth. Even using only those ligands binding in a very significantly deep pocket as judged from the p-value being less than 2 × 10^{-7}, no clear relationship is seen. However, if the data are restricted to those ligands that bury less than 500 Å² of surface area, there is a significant positive correlation between depth and affinity of $R = 0.23$. Restricting the area burial still further to <400 Å² and then <350 Å² increases the positive correlation to $R = 0.34$ and $R = 0.47$, respectively, with a positive slope of about 2.5 (inset, Figure 9). Although the amount of data at lower areas is sharply reduced, the trend is clearly that affinity depends on depth when the area buried by the ligand is low. This indicates that there is no simple dependence of binding affinity upon depth, because of factors such as surface area burial and no doubt other factors alluded to above. To extract the broad trend in binding affinity, the affinity data were modeled as a two variable function of buried area and depth, $-\log K = f(\delta A, d_b)$. Using the two-dimensional data smoothing function in Origin, a piecewise linear approximate $f(\delta A, d_b)$ was constructed using a 10x10 interpolation matrix with weighted averaging. The resulting function $f(\delta A, d_b)$ is plotted as a surface in the three dimensions of $-\log K$, $\delta A$ and $d_b$ in Figure 10. For additional clarity the $f(\delta A, d_b)$ is also depicted as a gray scale contour plot in the upper projection of the Figure. Considering first the effect of buried surface area, at small to medium depth the affinity shows an initial approximately linear increase, followed by a plateau. This general trend follows closely that seen in earlier broad surveys of the effect of ligand size. At the greater travel depths, there are very few compounds, and the range of observed buried areas is sharply restricted to small values so no trend is discernable. Considering now the effect of travel depth, for small burial areas greater travel depth does appear to increase the binding affinity. For larger amounts of buried area, the affinity is insensitive to the average binding pocket depth. Overall, the highest binding affinities occur at comparatively low buried surface area and high travel depth, though there is not much data in this region. These results are of course broad trends which “average out” the effects of specific interactions, shape effects, etc. in each complex, but the analysis demonstrates the kind of questions one can now examine quantitatively with a good measure of depth. Considering further the argument that binding pocket depth would primarily affect the polar desolvation contribution to binding, this contribution would be larger for charged ligands such as ions than for neutral ligands. For charged ligands, the desolvation term would be larger for more highly charged ligands. This implies that if such an effect of depth on binding affinity exists, it would be more important for divalent ion ligands than for monovalent ion or neutral ligands. Most RNA structures require the divalent ion Mg^{2+} to fold and maintain a stable structure, and the growing number of RNA structures with bound magnesium ions allows one to analyze where the magnesium is binding, and how such binding might be related to surface depth. For these reasons we next investigated magnesium binding to RNA structures. We first extracted all RNA PDB entries with magnesium bound. After careful checking of the structures, and eliminating cases where the magnesium ions were bound to non-RNA molecules in the complex, we were left with a set of 29 structures. No pruning by similarity was performed: there are several tRNA structures, several dimerization site initiation points, and several pseudoknots, for instance. The following PDB codes were in our final set: 1EVV, 1F27, 1FIR, 1ITJ, 19V, 1K9W, 1KXK, 1O9Z, 1TN2, 1TRA, 1XP7, 1XPE, 1XPF, 1Y73, 1Y95, 1Y99, 1YKV, 2B8R, 2B8S, 301D, 310D, 3TRA, 430D, 462D, 468D, 469D, 470D, 471D, 4TNA. These structures contained 249 magnesium ions that were

![Figure 7](image-url) Figure 7. A histogram comparing travel depths of different structure subsets: a control set with no known ligands, the PDBbind set, and just the binding pockets of the PDBbind set. The control and binding pocket curves have been normalized so that the area under each curve equals that of the PDBbind set.

![Figure 8](image-url) Figure 8. Average travel depth of the entire protein (△) and just the binding site (○) plotted versus the cube root of the number of heavy atoms, for proteins in the PDBbind dataset. Lines show linear least squares fits for the entire protein ($y = 0.55x - 2.72$, $R^2 = 0.84$) and for the binding site ($y = 1.54x - 8.13$, $R^2 = 0.47$).
bound to RNA, or closer to RNA than to other molecules in the PDB structure. We broke down surface points on the RNA structures into five types, based on the nearest atom. The five types are phosphate, sugar, and three nucleotide groups, the major groove, minor groove, and other. Non-standard nucleotides found commonly in tRNA were grouped in with the other group in the analysis, since they do not have standard hydrogen bonding patterns and usually do not conform to the major/ minor groove distinction.

Figure 11 shows an example tRNA structure with one magnesium bound, PDB code 1FIR. The color scale on this runs to 17.6 Å in depth at the deepest blue. Figure 12(a) shows the distribution of surface depths where magnesium ions are bound within 4 Å of surface points, broken down into the five categories. Figure 12(b) shows the relative frequency of binding at each depth and category data by normalizing the curves in Figure 12 (a) by the number of surface points at each travel depth in each of the five categories. The normalized data are cut off at depths >13 Å since more than half the category/structure combinations either had no data representing that level, or the number of points was so small as to be statistically insignificant. The major feature from this analysis is the significant amount of Mg$^{2+}$ binding near major groove atoms at a depth of 9 Å. This is apparent even without normalization.

The relatively high frequency of magnesium binding to the phosphate backbone category at depths ≈4–12 Å seen in Figure 12(b), appears somewhat significant in the context of the RNA

**Figure 9.** Mean ligand binding site travel depth, $d_b$, plotted against experimental binding affinity for the PDBbind dataset. Only ligands bound significantly deep ($p$-value $<2 \times 10^{-7}$) are shown in this analysis. Line shows the linear least squares fit, with ($y = -0.39x + 16.5$, $R^2 = 0.0199$). Inset shows the same plot for complexes that bury less than 500 Å$^2$, 400 Å$^2$, and 350 Å$^2$, respectively.

**Figure 10.** The binding affinity from the PDBbind dataset, $-\log K$, described as a function of two variables, average travel depth of the binding pocket, $d_b$, and surface area buried upon binding, $\delta A$ as $-\log K =$ $f(\delta A, d_b)$. $f(\delta A, d_b)$ is plotted as both a gray scale colored surface in 3-D and as a gray scale 2-D contour plot in the upper part of the Figure. Determination of $f(\delta A, d_b)$ is described in the text.
structural database available at this time (Figure 12(a)). Looking at the overall frequency/depth distribution (Figure 12(a)) without regard to category, one can see a fairly uniform distribution of depths from 0–12 Å (Figure 12(a)), though the three peaks, two for phosphate regions and one for the major groove, seem significant. The relatively uniform distribution, with ions occurring quite frequently at depths up to 10 Å indicates that there is little depth dependency to the desolvation penalty. This probably follows from the fact that RNA structures tend to be quite open and highly solvated compared to protein structures, even in deep pocket or groove regions, as exemplified by tRNA in Figure 11. As described previously, the irregular shape of the molecular surface influences the electrostatic potential, creating pockets away from the immediate vicinity of the phosphate groups where cations are likely to bind.32 Our analysis supports the idea that these pockets occur relatively often in major grooves, and they are distributed around a specific travel depth of about 10 Å.

As a final example of the use of travel depth we consider the problem of automatically identifying ligand binding pockets. This is a difficult problem, and the latest methods, which rely to a large extent on pocket volume, still encounter problems identifying extremely buried pockets.33 Deeply buried pockets are often not the largest by volume. A different way of solving this problem involves clustering surface points within some distance of the centroid of the protein atoms,34 which is another way of incorporating depth information. The centroid method requires careful selection of the appropriate domain or subset of atoms to give sensible results, and so it is not straightforward to apply in large, multi-domain proteins. We examine the case of the FS4 cluster ligand binding site in PDB entry 1H2R, which is reported as problematic.35 There are five separate ligands: three different iron-sulfur clusters, a nickel-iron active center and a magnesium ion. This protein’s overall average

![Figure 11. An example tRNA structure (PDB code 1FIR). A magnesium ion is shown as a purple sphere.](image)

![Figure 12. Histograms of depth of magnesium binding surface in RNA structures. (a) Frequency of each of five classes of surface points at given travel depths. Points are counted if within 4 Å of a magnesium ion. (b) As for (a), except each point was normalized by the overall number of surface points of that class at that travel depth. Data are only shown where at least half the classes/structures had data at that depth, in other words equal to or below 13 Å.](image)
depth of molecular surface is 8.7 Å. Examining this depth in terms of our earlier analysis of protein size against average travel depth, this is more than an angstrom deeper than the trendline. This means, for a protein of this size a rugged surface. Our travel depth algorithm gives average depths of the ligand binding sites as 27.6 Å, 22.9 Å, 38.5 Å, 18.6 Å, and 42.1 Å respective to the five ligands indicated in the structure by the abbreviations: FS3, FS4_1, FS4_2, MG, NFE. Figure 13 shows this structure and the travel depth colored surface. All ligands except magnesium are significantly deeper than the average depth, and they would clearly demark these regions in a blind search. However it is notable that the deepest pocket contains no ligand. This example suggests that combining depth and volume would significantly improve binding site identification in cases where pocket volume alone fails.

Discussion

We have introduced here a quantitative, robust and useful definition of the depth of any region of a triangulated surface of a molecule. We have also implemented this definition with an approximate, though sufficiently accurate and fast algorithm. This implementation is suitable for quantitatively analyzing individual molecules or large databases of molecules. The algorithm satisfactorily quantifies binding pockets in proteins as intended. Interestingly, travel depth also works for two difficult cases

Figure 13. A sample case where pocket volume does not correlate well to ligand binding sites. PDB code 1H2R, NiFe hydrogenase is shown in two views. In one, gray ribbons are shown with the four ligands and magnesium ion in light blue. The other view, from the same perspective shows the surface colored by travel depth, only the surface with travel depth greater than 16 Å is shown, cavities have also been removed for clarity.
for which it was not specifically designed. The first is for grooves in DNA, which present an interesting case since the grooves are unbounded in one direction. Second, our algorithm works in the case of tunnels, that is pockets that have no bottom, but go through the entire macromolecule.

Our definition of travel depth differs significantly from depth measures used in previous work. Other definitions have been proposed based on the difference between molecular surfaces of varying smoothness. GRASP\cite{10} has a macro called Molecular Difference, which produces the difference between the normal molecular surface and a molecular surface generated with a probe radius of 10 Å (Malcolm E. Davis, © 10/31/1995). APROPOS used a smoothed Euclidean difference between two surfaces generated with a probe radius of 10 Å. Both the GRASP macro and APROPOS program compute a simple Euclidean distance, ignoring the complicated surface structure of the macromolecule. The travel depth defined here, in contrast, uses a non-Euclidean, macromolecule-avoiding distance. While a Euclidean distance agrees with our definition in cases where the paths to the convex hull are simple non-macromolecule intersecting straight lines, it differs when the macromolecule contains overhanging and narrow tunnels to interior binding sites. This occurs frequently in proteins: over all surface points in the PDDBbind dataset used in our analysis, about 52% of the surface points had a higher travel distance than Euclidean distance, and 5% of the surface points had a difference above 5 Å. Moreover, most of the large errors occur in pockets, which are the regions of most interest. The APROPOS definition of depth is also not taken from molecular surface points, and has been highly tuned to detect binding sites. Our method is more general than these; it can calculate depths for any molecular surface, it works for pockets of any size and complexity, it can also calculate depths for the volume between the convex hull and the molecular surface.

Additionally, our definition is quite different from the notion of extreme elevation.\cite{35} The extreme elevation is a height distance between any two points on the surface, and the algorithm finds all points that are local maxima of such a function. These pairs of points that maximize the elevation could be used in some applications, however it does not define a general notion of depth for every surface point, as a point could be in several pairs. The extreme elevation method also has a high asymptotic complexity, but has been shown to help generate possible poses in docking applications.\cite{36}

Our algorithm for solving the shortest path problem is significantly different from previous work. The related shortest path planning problem in three dimensions was shown to be in the class \text{NP}-hard with respect to the obstacle complexity,\cite{37} however some approximation algorithms have been proposed.\cite{38,39} The previous approximation algorithms usually subdivide each edge of the polyhedral obstacle into smaller pieces, then compute visibility maps among vertices. The code required to compute visibility maps in three dimensions is complex. Moreover, the computation time is large. Other previous approaches, which we did not use, are conformal or constrained meshing, fast marching methods, and proximity depth. If a reasonable quality conformal mesh could be generated between the molecular surface and the convex hull it would be easy to apply multiple source shortest paths to compute minimum travel distances.\cite{40} There is a large amount of previous work on fast marching methods, algorithms to grow expanding boundaries. Again, the algorithms and code to implement these approaches are complex.\cite{41} Also, producing the intermediate conformal mesh given an arbitrary triangulated molecular surface may be difficult unless constraints are applied to the latter. Such constraints may impose undesirable compromises on the type and resolution of molecular surfaces that could be handled. In this context, our minimal travel distance algorithm can be viewed as a discretized fast marching method, or an approximation to a conformal mesh generator constrained to a cubic lattice, although for our purposes it need not fully implement either of these intermediate constructs. Finally, our work is most similar to the notion of proximity depth, which has been developed for various proximity graphs where edges exist between close points.\cite{42} In contrast we generate our points and weighted connectivity in different and explicit ways, and specifically exclude certain edges, those passing through the macromolecular surface.

In its actual implementation, our travel depth approximation algorithm has several advantages: It works on any orientable and connected triangulated surface, it is relatively easy to code, it has polynomial complexity and it has practical computation times. The result is at most a constant multiplicative factor worse than the true travel depth and this constant can be controlled to a degree. In application to several different kinds of macromolecular surface, including analysis of DNA groove depth, our measure agrees with previous qualitative descriptions, and with one’s intuition when looking at structures.

The significance of having only polynomial complexity and practical computation time is that the algorithm is practical for high-throughput analysis for large macromolecules. Our analysis of the PDDBind database of 900 protein–ligand complexes required less than a week of computation time on a single computer, encompassing all aspects of the computation: surface generation, travel depth computation, and all statistical analyses.

In analyzing the protein database, the travel depth algorithm revealed two important features. First, proteins with known ligand binding sites have a different depth distribution profile than those without known binding sites. Those with ligand binding sites have a wider profile at low depths with a higher
tail at high travel depths. Second, protein size as a function of the cube root of the number of atoms explains a lot of the variation in overall travel depth from protein to protein; a clear linear correlation between depth and size is present.

The travel depth analysis was used to show that ligands tend to bind in deeper pockets. Moreover, when this analysis is combined with surface area calculations, it shows that binding in deeper pockets has a significant effect on binding affinity when surface area burial is low. Though the picture is not yet completely clear, this specific two factor effect has not been suspected before. As is apparent from the contour depiction of $f(\delta A, d_b)$ in Figure 10 there are no complexes with both large buried surface area and great depth. Again this is unanticipated, and it may reflect some intrinsic constraints for good ligand binding in proteins. Although the PDBBind database is quite large, with diverse ligands and protein sizes, the lack of large $\delta A$ plus $d_b$ could also reflect limitations of the PDBBind dataset. If complexes with this combination of $\delta A$ and $d_b$ are discovered, along with more structures with low buried surface area and high travel depth, the relation between $\delta A$, $d_b$ and binding affinity could be better understood.

In addition, we analyzed the influence of travel depth on magnesium binding to RNA structures. Our preliminary conclusion is that there is little effect of desolvation in deeper ion binding sites. Also, perhaps surprisingly, a significant number of Mg ions bind closer to the major groove than to the phosphate groups, in contrast to one’s naive expectation based on charge complementarity. Since the RNAstructural database at present is rather small compared to that of proteins, more data are needed to make the picture clearer.

It has been well established that the substrate binding and enzyme active site of a protein is commonly located in the pocket with the largest or second largest volume. However, there are some cases where neither of the largest pockets by volume contain the enzyme active site. Some of these cases are peptide recognition sites that are commonly spread across the surface of the enzyme. However, other cases where the ligand lies in a deep, small pocket may benefit from taking depth of the pocket into account. Our general analysis showing that many structures have ligands that bind in significantly deep pockets reinforces this conclusion. These issues are part of the larger problem of automatic identification of ligand binding sites. We considered one difficult example in this area as an illustration of how the travel depth analysis can help. In addition, travel depth may help in the further problem of discriminating between different kinds of active sites.

Having a quantitative definition of travel depth also now allows one to combine this property of the surface with other features for analysis. Other surface features include volume, surface area, curvature, and chemical features like electrostatics. Together with sequence properties like conservation, the combined analysis of all these features may allow for excellent overall prediction of ligand binding site location. A recent example of this kind of analysis shows the importance of having a good fast quantitative definition of depth. Depth analysis would also be useful in structural genomics projects where little functional information is known about a new structure.

Future directions

The algorithm that we develop here for measuring depth of macromolecular surfaces is a discretized approximation of a multiple shortest paths (MSP). The paths are initiated at the convex hull, and terminated at the molecular surface. We chose the convex hull as a natural and practical reference point to initiate the MSP, but we note that other choices are possible. Among the class of convex surfaces, an ellipsoidal or spherical surface completely enclosing the molecule is another possibility for a reference level. Some choice must be made of how far outside the molecule is this surface, however, which introduces another arbitrary parameter. One expects that an ellipsoidal initiation surface suitably chosen and aligned to the molecule’s axes of inertia would provide much the same rank ordering of depths as the convex hull. Alternatively, as a non-convex shape, one could use a molecular surface created with a large probe radius as the reference level.

Additionally, varying the probe radius used to generate the molecular surface would be another straightforward variant of our method. Our algorithm works for any triangulated surface with reasonable constraints. A larger probe radius might mimic, for example the larger size of a ligand molecule group compared to water. Indeed, the default use of water-sized probes to create the molecular surface is a standard caveat in this area. For instance, minimal distance paths may travel through water tunnels to active sites, though the path a larger ligand may travel would be different, and probably longer.

More generally, our travel distance implementation of MSP could be used as a measure of shortest avoiding distance in a variety of applications to macromolecules. For example: (1) In cases where a particular protein pocket is accessible by more than one pathway or “tunnel”, traveling back along the steepest descent of travel depth values, or simply recording the last step taken to arrive at each grid cube, would provide the shortest “escape” route and its length. (2) Examining the union of all such escape routes from a ligand binding site could also give interesting information, for instance, examining the number of grid cubes with varying depth from 0 upwards could yield information about the steepness or width of the tunnel. (3) Taking the molecular surface as the initiation surface, and propagating the travel distance inside the molecular surface until it self terminates (when all the grid points inside the surface have been assigned) the algorithm would assign a depth, with respect to the nearest surface point,
every part inside the molecule. Applications of this depth include quantifying the depth of burial of side-chains in a protein core. This analysis would be similar to the notion of atom depth⁴⁹ and likely yield chains in a protein core. This analysis would be depth include quantifying the depth of burial of side...

Acknowledgements

We thank the following for many useful discussions: Miller Lee, Dan Kulp, John Skinner, Nathaniel Nucci, Jennifer Dashnau, Dr Ninad Prabhu, Qingyi Yang, and the many others who gave useful feedback on preliminary talks given on this subject. We thank Dr Diane Souvaine for reading preliminary versions of the manuscript. Support from the NSF (MCB02-35400) is gratefully acknowledged.

Supplementary Data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.jmb.2006.07.022

References
