Finding and Characterizing Tunnels in Macromolecules with Application to Ion Channels and Pores

Ryan G. Coleman[†‡] and Kim A. Sharp[†‡*]
[†]The Johnson Research Foundation and Department of Biochemistry and Biophysics, University of Pennsylvania, Philadelphia, Pennsylvania 19104; and [‡]Genomics and Computational Biology Graduate Group, University of Pennsylvania, Philadelphia, Pennsylvania 19104

ABSTRACT We describe a new algorithm, CHUNNEL, to automatically find, characterize, and display tunnels or pores in proteins. The correctness and accuracy of the algorithm is verified on a constructed set of proteins and used to analyze large sets of real proteins. The verification set contains proteins with artificially created pores of known path and width profile. The previous benchmark algorithm, HOLE, is compared with the new algorithm. Results show that the major advantage of the new algorithm is that it can successfully find and characterize tunnels with no a priori guidance or clues about the location of the tunnel mouth, and it will successfully find multiple tunnels if present. CHUNNEL can also be used in conjunction with HOLE, with the former used to prime HOLE and the latter to track and characterize the pores. Analysis was conducted on families of membrane protein structures culled from the Protein Data Bank as well as on a set of transmembrane proteins with predicted membrane-aqueous phase interfaces, yielding the first completely automated examination of tunnels through membrane proteins, including tunnels that exit in the membrane bilayer.

INTRODUCTION

Proteins adopt three-dimensional structures with complex shapes and surface topography. These topographical features, such as clefts, flaps, and tunnels, often have important functional roles. We define here the term tunnel or pore to mean a hole that goes completely through the protein, thus having two entrances or mouths. Many proteins contain tunnels or pores that are of physiological importance, the primary examples being membrane protein ion channels, pumps, porins, and transporters. Although some channels have a single simple tunnel structure, there are also more complicated structures, for example the mechanosensitive channel of small conductance (MscS) (1). Also, proteins such as the ring clamp protein (2), the ribosome (3), and other proteins involved in transcription have topological features including pores that are important for interactions with DNA strands. Spastin has a central pore that is involved in microtubule severing by pulling the end of the tubulin polypeptide through the pore (4). Some enzymes such as rubisco also have tunnels through them (5). At least one enzyme, acetylcholinesterase, has a tunnel observed under simulation with distinct exits for the two products (6). Photosystem II has three tunnels leading to the active site, theorized to be pathways for water, oxygen, and protons (7). Finding, cataloging, and measuring these tunnels are important in understanding their function. The ability to do these automatically is an important step toward automation of structural genomics or characterizing new protein structures. Although fewer than 400 high-resolution structures of transmembrane proteins are currently known, and of these only ~150 are unique (8), many advances in techniques should increase this number (9), particularly as membrane proteins become targets of large-scale structural genomics projects (10). Comparisons to the growth of globular proteins in the Protein Data Bank (PDB) suggest that ~2200 membrane protein structures will be deposited by 2025 (11). Additionally, as new examples of subclasses of membrane protein structures are found, accurate homology-modeling studies become possible (12). Tunnel analysis will increasingly be needed because these structures will no doubt include many new pumps, pores, channels, and transporters.

The seminal work in characterizing protein tunnels was the development of the HOLE algorithm (13). The algorithm has been applied very successfully to analysis of ion channels, in which the position and orientation of the pore (normal to the membrane) are known a priori, and can be used to ''prime'' the HOLE search algorithm. The algorithm is less able to deal with arbitrarily positioned tunnels or multiple pores, and it is difficult to automate because it needs some initial user guidance. Additionally when multiple tunnels are present, HOLE or variations of HOLE were not able to find the ''correct'' tunnel among several in some ribosomal structures (3). There has been some work in calculating cavities and their volumes or volumes of portions of tunnels (3,14). Additionally, CAVER functions as a three-dimensional version of HOLE in some respects, but it still needs a starting hint to find a tunnel, and it is primarily geared toward finding paths out from a pocket, not tunnels all the way through proteins (15,16). However, no further work in automatically identifying tunnels has taken place since the introduction of HOLE. This attests to the difficulty of developing a completely automated, general tunnel-finding/measuring algorithm. We present such an algorithm, which we call CHUNNEL, and then describe the principles

Submitted April 23, 2008, and accepted for publication September 16, 2008.

*Correspondence: sharpk@mail.med.upenn.edu

Editor: Klaus Schulten.

0006-3495/09/01/0632/14 $2.00

doi: 10.1529/biophysj.108.135970

of both topology and geometry on which it works. We then test CHUNNEL on a set of proteins with artificially generated pores of known path and width and on various membrane proteins with tunnels from the PDB database (8,17). Tests of the HOLE algorithm were also performed on the same test set to compare the two algorithms and show that CHUNNEL has a markedly improved ability to find tunnels automatically. We also show that CHUNNEL can be used to prime HOLE, which can then trace and characterize the pore. We also use CHUNNEL to find qualitatively new tunnels, for instance those that exit within the membrane bilayer, which have not been found or examined previously.

METHODS

General outline of the approach

The procedure developed here for finding and characterizing tunnels is an outgrowth of our previous work characterizing depths of pockets, grooves, tunnels, and other surface features in macromolecules using a measure known as ''travel depth'' (18). The travel depth of a point on the molecular surface (MS) is defined as the shortest path through the solvent to that point from a reference surface (specifically the convex hull of the protein). The shortest-paths algorithm (19), specifically the generalization we call multiple source shortest paths (MSSP) (20), is used to compute the travel depth, and it is implemented by discretizing space on a cubic grid. After the application of the MSSP algorithm, all surface points have been assigned travel depths (18). In addition, the travel depths of all solvent grid points lying between the convex hull and the MS are known.

The impetus to develop a tunnel-characterizing algorithm from this work had two sources. First, although the Travel Depth algorithm was designed to characterize pockets and clefts, an unexpected benefit is that it also measures the depth of both the lumen and the surface of a pore (18). Second, the MSSP algorithm has proven to be a general-purpose algorithm for calculating volume-avoiding, shortest-distance pathways. If the MSSP algorithm is started at the MS, and the distances are propagated outward in the solvent, then the ''Travel Out'' distance assignment will self-terminate in tunnels, forming a ''ridge'' or everted medial axis in three dimensions. These two observations suggested that by starting at a maximum in Travel Depth and Travel Out distance, and following ridges in Travel Out distance of decreasing Travel Depth in two ''opposite'' directions, one would trace out the path along the center of a pore. The Travel Out distance along this path gives the radius of the pore at each point. In practice, using just these two distance functions, it is difficult to automatically distinguish the difference between the bottom of a pocket and the center of a tunnel. It is also difficult to follow a ridge of distance in three dimensions, especially with the discretization of space required to implement any algorithm. This problem, sometimes referred to as thinning, shape skeleton, or medial axis is complicated even in two dimensions (21–23) and can be approximated only in three dimensions (24). Hence, to implement this approach, it is necessary to first ensure that the starting point is in a pore and then correctly follow the pore out in both directions. In addition, if there are multiple pores, one needs to reliably identify starting points and propagation directions for all of them. We achieve this through topological and geometric analysis of the MS.

Generation and preprocessing of the surface

We start with the generation of the MS using the algorithm in the GRASP macromolecular graphics program (25) implemented as a stand-alone program. Standard atomic radii (26) are used to generate the MS with a probe radius of 1.2 Å. This is a somewhat smaller probe radius than used previously to treat ion channels: the permeant ions can have radii less than the standard probe radius of 1.8 Å used for water. The modified GRASP surfacing algorithm first maps the molecule onto a cubic grid. It then produces a closed triangulated surface for which the vertex coordinates, vertex connectivity, triangle normals, and triangle connectivity are known. All cavities, defined as smaller disjoint sets of connected triangles, are discarded. In addition, because of the way this surface is generated, the volume inside and outside the MS is already discretized on a cubic grid whose vertices are labeled as in or out (see Fig. S1 in the Supplementary Material). The vertices of the surface triangles also lie on edges joining inside and outside vertices of the volume grid, whereas triangle edges cross the surfaces of grid cubes or lie completely within a single grid cube (Fig. S1). This well-defined relation between surface and volume discretization is key to the successful implementation of the tunnel-finding algorithm, as the latter uses both surface and volume properties. The final step in the surface generation/preprocessing is to generate the Convex hull using the Qhull algorithm (27), which also generates a closed, triangulated surface.

Enumeration and localization of pores

Triangulation of the MS (after discarding cavities) immediately provides the number of tunnels or handles present through the Euler relation:

$$V + F - E - 2 + 2N = 0, \quad (1)$$

where V, F, and E are the number of triangle vertices, faces, and edges, respectively, and N is the number of handles, so the surface is an N-torus. Although the number of tunnels is known from this topological invariant, there is no indication of their location. With a complex protein surface, it is often difficult to find them even using three-dimensional modeling graphics.

The first step to localization of the tunnels is to ''remove'' from the surface a maximal region of triangles, D, that is topologically equivalent to a disk. A triangle is picked at random to start D, and neighboring triangles are removed until it is impossible to remove another triangle and have the boundary of D remain a simple, closed, nonintersecting path (Fig. 1 *a*). The remaining triangles form a closed strip of triangles, S, one triangle wide with $2N$ loops. The loops come in N pairs of which one runs around each pore (an A-loop), and one runs through each pore (a T-loop). Fig. 1, *a* and *b*, shows a residual strip S for a torus (1-torus) and for a 2-torus.

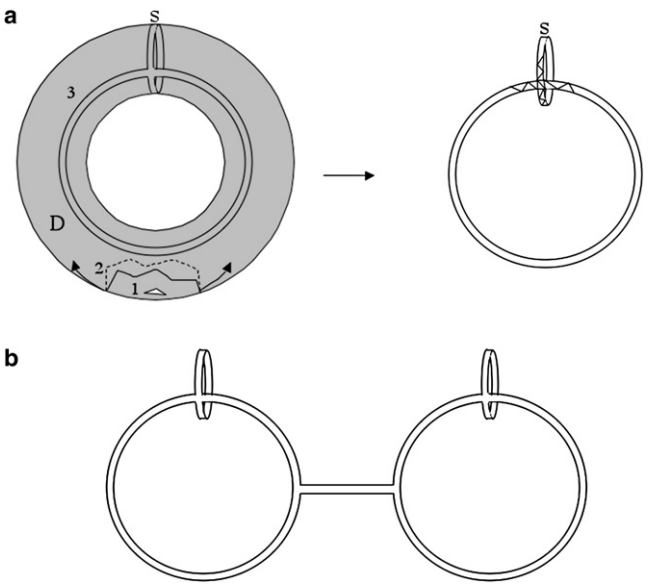

FIGURE 1 (*a*) A 1-torus showing the original starting triangle for disk region D (1), a partially expanded region D (2), and the final maximally expanded region D and the corresponding leftover minimal strip S (3). The minimal strip S is also shown separately for clarity. (*b*) A minimal strip S for a 2-torus.

On a complicated protein surface, the path of S is usually very irregular and far from minimal in length (Fig. S2). This divagation is usually great enough that one cannot at this point reliably categorize a loop as A or T just from the coordinates and orientation of the constituent triangles. In particular, there is no requirement for the A-loops to be anywhere near either the center or the narrowest part of a pore.

Obtaining a "tight" loop of triangles around a pore

The next step is to regularize or "tighten" S around the pores and then find a set of N A-loops that are topologically distinct and go around each pore in the surface. A careful combination of topology (to ensure that the A-loops found are distinct) and geometry (to ensure that such loops are tight) must be employed to accomplish this goal because neither approach by itself would work. First, the triangles of S are decomposed into $2N$ sets S_L, $L = \{1\ldots2N\}$, one for each loop (some triangles may be part of more than one loop). By use of the MSSP algorithm, neighboring triangles are sequentially added to a loop S_L (it is "fattened up") until its edges wrap around and meet at some point (Fig. 2 *a*). Because triangles are added in order of minimum neighbor distance from the original strip, one can trace back neighboring triangles from the meeting edge along the shortest path to S_L. The set of trace-back triangles forms another one-triangle-wide strip S'_L, which is the complement of S_L: If S_L is an A-loop, then S'_L is a T-loop, and vice versa. At this point one can automatically and reliably classify such a loop as A-type or T-type from its triangle surface normals by checking whether they point toward each other (A-loop) or away from each other (T-loop). A regularized A-loop runs around the narrowest part of a pore because of the shortest paths property of the MSSP, and so it more tightly delineates a pore.

Identifying two distinct directions in a pore

Now that we have generated and identified a regularized A-loop, the next step is to unambiguously define the two distinct directions from the A-loop out to the two tunnel mouths. We achieve this by building a "plug" in the A-loop starting from the strip of triangles S'_L forming the regular A-loop. This strip has two edges, G and H (Fig. 3 *a*). We collect two sets of grid points G and H such that any point in G is closer to a vertex in G than to any vertex in H, and vice versa for members of H. Additionally, any grid point g in G has at least one neighboring grid point h in H, and vice versa. The sets G and H are defined as the opposite sides of the plug. This procedure constructs an oriented "leakproof plug" across the pore circled by the regular loop S'_L. It is leakproof in the sense that there is no way to pass from one side of this region of the grid to the other while staying in the solvent without passing through at least one grid point from either side. It is oriented because we know from which edge of S'_L a plug point derived. Thus, the plug separates one side of the pore lumen from the other (Fig. 3 *a*).

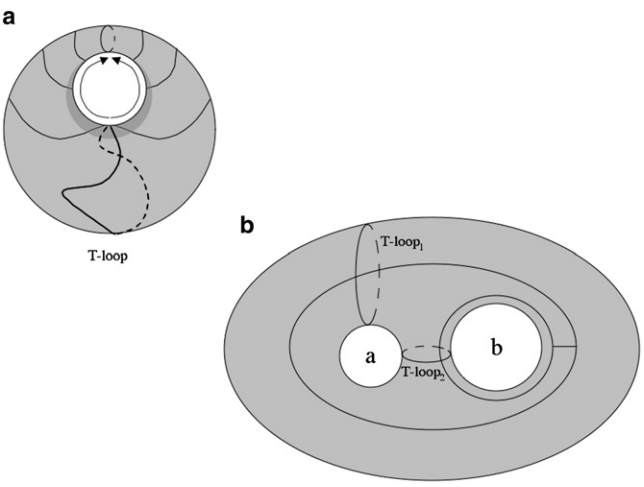

FIGURE 2 (*a*) A T-loop (bold line) whose two boundaries are sequentially advanced across the surface (light lines) to eventually meet (at arrows). Traceback according to the shortest paths algorithm (along arrows) yields a regular A-loop. (*b*) Two T-loops that both regularize to form A-loops around the same pore *a*. No A-loops are formed around pore *b* in this case, so pores must be processed and capped one at a time.

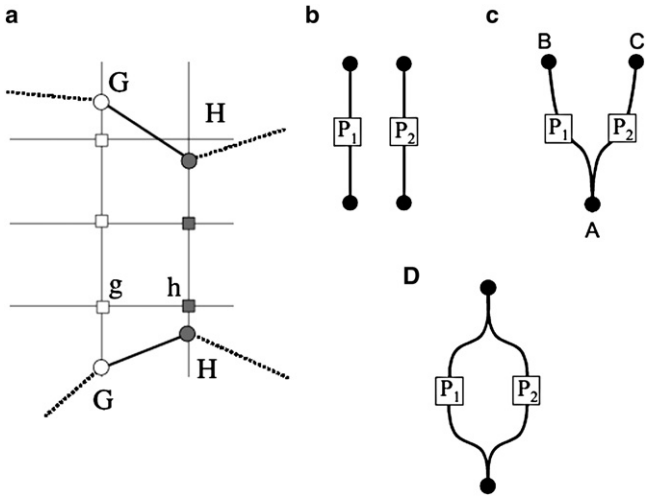

FIGURE 3 (*a*) Two-dimensional representation of a plug. Shown are the surface (*dotted and heavy lines*) and the volume grid (*light lines*). (*circles*) Bounding vertices G and H, respectively, of the regular A-loop. (*squares*) The final plug vertices, with fill indicating sides. (*b–d*) All possible topological cases for a 2-torus: (*b*) two completely separate pores, (*c*) two pores that share one endpoint, (*d*) one pore that bifurcates in the middle.

In some cases, a regular A-loop will produce a plug that extends out beyond the convex hull. This interferes with the later path-finding procedure, but this is easy to correct by generating new loops and new corresponding regular loops using a different random initial triangle. Plugs that do not extend beyond the convex hull are referred to as valid.

Ensuring a complete and nonredundant set of A-loops

Because it is possible for an unregularized T-loop to pass through two pores, or for two such T-loops to pass through the same pore, it is possible that the regularized A-loops derived from them would not completely and nonredundantly girdle the N pores. This possibility is illustrated for the simple case of a 2-torus with one narrow pore and one wider pore. If both the loops around the handles find a regularized loop around the narrower pore, the wider pore will not have a corresponding regularized loop (Fig. 2 *b*). The solution is to apply the regularization procedure recursively, "masking" off each pore as it is identified and plugged. A pore is masked off by removing the triangles S'_L of its regularized A-loop and creating two caps of new surface triangles joined along the boundary edges A and B, updating the connectivity information of the surface triangulation as necessary. The remaining surface is now an $(N - 1)$-torus. The procedure of residual strip generation, A-loop regularization, plug generation, and masking off is repeated until all N pores have been processed. We note that in practice this recursive step is the slowest step of our algorithm, as it has a quadratic dependence on the number of handles in the surface and a linear dependence on the number of grid points and surface triangles.

This set of N regularized A-loops with valid plugs contains one loop around each pore in the original surface and one valid plug in each pore. Additionally, simple checks are done to ensure that all loops are in the original surface; that is, they do not contain triangles that were added or removed in the pore-masking step.

Generating a path through a pore

Each plug is used in turn as the starting point to generate two half-paths out of the pore, one in each direction, terminating at the convex hull. The two half-paths start from plug points on opposite sides. This ensures that the complete path really traverses the pore (i.e., does not double back and emerge from the same end it started from).

First the MSSP algorithm is used to assign a Travel Depth and Travel Out distance to each solvent grid point between the convex hull and the MS. The initiating surfaces for these are the convex hull and the MS, respectively.

Next, the plug point on one side with the maximum Travel Out is identified. Starting from this point a branch-and-bound search algorithm (20) is used on the Travel Out distance, with higher distances taking precedence, leading to a path that passes as close to the center of the tunnel as possible, following the ridge of maximal Travel Out distance. The path is terminated at the first grid point encountered outside the convex hull. In cases where multiple plug grid points have the same maximum value, each path is traced out, and the one with the highest minimum value of Travel Out is kept, i.e., the one with the widest choke point. This procedure is repeated on the other side of the plug. To connect the two half-paths, the two plug grid point maxima (one from each plug side) are connected in a branch-and-bound search because this again gives a path that maintains the highest minimum Travel Out distance. We note that maximizing some minimum metric has been successfully applied to finding topological paths before (28). Our approach here is similar to the approach of CAVER (15,16). Our concept of Travel Out distance is the same as the r_{max} function from CAVER, although the methods used to compute them are different. However, in contrast to a branch-and-bound search to maximize the minimum radius of the path, CAVER uses a modified shortest paths search to find a path out, which would seem to maximize the total radius passed through; this differs from our paths.

Building all topological paths through the pores

In cases where there is more than one pore, the set of half-paths generated by the branch-and-bound algorithm may be combined in different ways to form alternative full paths (Fig. 3 *b*). For example, a Y-shaped or branched tunnel has three entrances, A, B, and C, and one can define three full paths A-B, A-C, and B-C, which share segments (Fig. 3 *c*). Finding one path per entrance/exit combination is not sufficient to get all topologically distinct (nonlooping) paths. A path is defined uniquely only by the entrance, exit, and plug maxima through which it passes. Therefore, all plug-to-mouth half-paths are added to a tree, which is then reprocessed to get individual full paths. This reprocessing attempts to connect all combinations of points in the tree by all possible noncycling paths. This gives all the possible topological paths of interest. The potential number of such pathways grows exponentially with the number of pores in the protein surface; however, most structures do not have the maximum number of pathways—in fact, many have only one pathway per pore, for instance, when none of the pores intersect.

Checking that paths traverse pores

An important final step in the path generation approach is to check each potential path to ensure that it passes through an actual topological pore in the protein. This prevents false positives. This is accomplished by using the set of tight A-loops, S'_L. If a path passes through at least one of these loops, then it passes through a pore in the protein. Starting with the loop of connected triangle vertices forming one border of a loop strip, A (Fig. 3 *a*), it is triangulated by arbitrarily selecting one point as the common base point, creating triangles using the other points, and then checking whether each path segment intersects with any of these triangles. An odd number of intersections means this path goes through this loop and, therefore, through a pore of the protein surface. We note that in theory a path could pass through more than one pore before encountering the convex hull. Currently only one passage is reported, although all passages could be reported with slight additional processing.

In summary, the above procedure results in a complete list of topologically distinct paths. Multiple paths can then be prioritized based on several geometric properties described below.

Test set of protein pores

Having a set of protein structures with realistic and known pores created in them is desirable for two reasons: first, to check the accuracy of the algorithm; and second, to test the algorithm without accessing the limited number of real pore and ion channel structures in the training phase. For this purpose, we created a set of ''punctured'' or drilled structures. Starting with larger structures (>100 residues) from the PDBbind database (29,30), pores were punctured from one side of the protein to the other by moving a sphere in a biased random walk (using a Von Mises Distribution (31)) from one side of the convex hull to another, removing all atoms overlapped by the sphere at any point. The radius of the sphere at each step was picked randomly from a Gaussian distribution and restricted to be between 2 Å and 4 Å. The bias for the Von Mises Distribution was set to either 2/3 or 2, which creates relatively straight or somewhat winding paths, respectively. This procedure was conducted a few times for each protein, then the resulting punctured structures were examined by hand to weed out some pathological cases; 86 relatively straight and 55 winding punctured structures were produced. Of this total of 141 known pore cases, a randomly chosen set of 100 was used during the development of the algorithm to identify errors and make improvements. The remaining 41 were reserved until the final version of the algorithm had been developed to provide an unbiased estimate of accuracy.

It should be noted that these structures have a reasonable exterior and a reasonable channel through them, but the composition of the interior side chains is severely disrupted by this puncturing process. Characterizing the pores using residue identities or other structural motif methods would not make sense. Because the algorithm presented here relies only on gross topological and geometric features and uses atoms, not residues, to create the surface, it is acceptable to use these punctured structures for training and testing. A probe radius of 1.2 Å was used in making the MS for these structures. This is much lower than the minimum radius of the created pores, to ensure that some additional pores would be present. Also 1.2 Å should be small enough for most real ion channel use, so this value was used throughout training and testing and in all further analyses except where noted. However, this radius could be changed in future applications, as nothing in the training or testing procedure is materially dependent on this parameter.

Quantifying and checking pores

A pore is fully characterized geometrically by the locus of the pore center and the width at each point (the maximum radius sphere that can be placed at this point). Other properties that are of interest include the length, the minimum radius over the entire length, the first minimum radius found from each end, and the maximum radius between those two minima (13). Additional geometric metrics are also computed because different properties may play roles of varying importance depending on the physiological function of the protein. To get some estimate of the uniformity of the path, the number of local minima is determined. The maximum travel depth is also computed, providing an alternative measure of path length. To estimate how direct a route the path takes, its length is divided by the distance ''as the crow flies'' between the ends, which will be 1 for a perfectly straight route and >1 for a route that takes a more circuitous path. This is called the winding metric. Given the path and its radius at each point, it is straightforward to identify residues lining the path or any particular subsection such as a choke point by identifying residues within the pore radius plus some additional distance threshold. The threshold of 4 Å was used for all analysis presented here, but this cutoff is under user control. CHUNNEL calculates and outputs each of these metrics for each pore, along with a listing of each tunnel's entrance and exit, and the plug(s) each path passed through, which together uniquely identify the tunnels in a multiple-tunnel structure.

Finally, CHUNNEL sorts the list of tunnels in order of decreasing minimum radius.

For test cases with known pore paths and radii, we designed several measures to check how closely a computed path matched the known path. Because paths are drilled and found by independent algorithms, each with a finite path point resolution, there is not necessarily a one-to-one mapping between points on the known and computed paths. In the following measures, for any pairwise comparison, each computed point is mapped to the closest known point.

1. Root mean-square deviation between known and computed paths. This was computed using either equal weighting (P_{rms}) or weighting by 1 over the radius of the known path (W_{rms}). W_{rms} weights the narrow sections of the tunnel over the typically wider mouths, as the former are usually more important to get right.
2. Span. We first determine all the points on the known path that are mapped onto by at least one computed path point. The two extremal mapped points are identified, and the span is defined as the fraction of the known path that lies between these two points.

By examining these measures, we can show how closely the paths computed by CHUNNEL are to the known paths in the drilled training and test structures; additionally, we can compare the performance of CHUNNEL to the performance of HOLE.

Computational requirements

The overall algorithmic complexity of finding tunnels is quadratic in terms of the topological complexity, linear in terms of the number of grid points, and quadratic in terms of the number of triangles. Outputting all possible paths is exponential in terms of the topological complexity because there are potentially that many possible paths; however, in most cases there are far fewer paths than this. To give an estimate of the practical runtimes involved, we performed some timings using one processor of a dual-processor machine (Intel 3.06 GHz chip, 6094 BogoMIPS, 2 gigabytes RAM) running GNU/Linux Fedora Core 4. The results are shown in Table 1. The relation between topological complexity and total processing time can be seen. Although no formal computational space analysis was performed, many hundreds of megabytes of RAM were often in use. Our code currently writes output files compatible with PyMOL, although customization for other programs is possible. Software is available at http://crystal.med.upenn.edu/.

RESULTS

Verification and accuracy of the algorithm

The CHUNNEL algorithm was developed on the drilled training set of proteins with known pores. The goal here was to reserve all real structures and the drilled test set of known pores for analysis only after the algorithm was completely developed and we could successfully identify the known pores in the training set. We note that of the 100 training cases, only 10 had a single tunnel. Multiple tunnels commonly arise during drilling when, as atoms overlapping a drill sphere are removed, an additional exit is created. These extra mouths are no different qualitatively from the known tunnel, except their exact path is not known. The CHUNNEL algorithm finds all tunnels, but for purposes of testing the algorithm, we focus on how accurately the single known path is found. Identifying which of the computed tunnels is the correct one for comparison with the known tunnel is straightforward either from visual inspection or by its significantly lower P_{rms}.

To interpret the accuracy of CHUNNEL, it is necessary to know what different values of the measures described previously (P_{rms}, W_{rms}, span) actually mean in terms of deviations between computed and known paths. In Fig. 4, a montage of nine examples from the training set is shown. In each image, the tunnel is shown via the surface, which has been clipped for visibility, along with both the known and calculated path. The examples were chosen to represent three ranges of P_{rms} values. The first row highlights computed paths that are essentially perfect; they are very close to the known paths from end to end. The P_{rms} values are less than 1 Å. In the second row, three examples with P_{rms} values of ~1.9 Å are shown. In the leftmost of these, both ends are slightly incorrect; in the other two examples, one end is moderately incorrect. However, these inaccuracies are in the mouths of tunnels, where the lack of a well-defined pore makes it harder

TABLE 1 Representative timings and algorithm statistics

	Sample A	Sample B	Sample C
Number of atoms	388	2,148	4,380
Number of handles	1	7	15
Number of triangles	5,564	37,520	63,832
Number of nodes	16,943	207,703	479,422
Number of paths found	1	11	156
Count handles (s)	0.001	0.005	0.008
Travel out (s)	1.1	58.5	222.7
Get loops and plugs (s)	2.1	249.0	1454.5
Find paths (s)	0.001	0.2	2.0
Total including I/O (min)	0.6	16.1	239.4

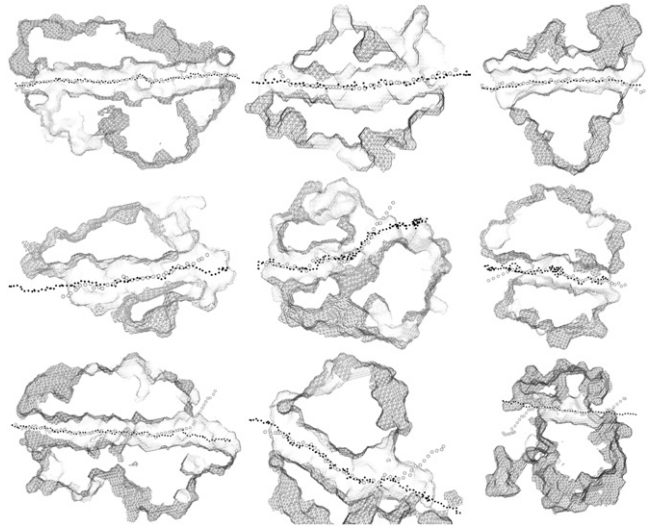

FIGURE 4 A montage of nine (of 100) sample training set cases. The known path is shown in black, the best path according to the lowest P_{rms} is shown in light gray (almost white) spheres; the surface is shown in cutaway. The top three cases have $P_{rms} < 1$ Å. The second row of three all have P_{rms} of 1.9 Å. The third row shows two examples with P_{rms} of 4.7 Å and then (*on the right*) a P_{rms} of 6 Å. Figures were produced using customized PyMOL (68).

to completely and accurately follow the entire length of the path. In the bottom row are examples chosen from the worst performance on the training set. The leftmost two examples have P_{rms} values of 4.7 Å, and in both cases, the computed path deviates from the known path in one mouth. Again, these inaccuracies can be attributed to wide mouths, and because the paths are still in the correct mouth, they are not a cause for concern. The rightmost example on the bottom row has a P_{rms} of 6 Å, and there are inaccuracies in both mouths. The W_{rms} for all these examples is lower than the P_{rms}: the top row is in the range of 0.4 Å to 0.7 Å, the middle row's range is 1 Å to 1.7 Å, and the bottom row's range is 2.5 Å to 3.4 Å. The range of values for span on these examples goes from the nearly perfect upper left example with 0.97 to the middle left example with a value of 0.60. The span values for the bottom row are all greater than this worst case value of 0.60.

With an understanding of the meaning of specific values for the various measures, we can examine the performance on the training and test sets of known paths. In Fig. 5, we show the best P_{rms} and W_{rms} values for the training and test sets. Most P_{rms} values are less than 2 Å, and most W_{rms} values are less than 1.5 Å, indicating that they have almost the entire path correct. There are, however, a number of cases where wide mouths cause the computed path to have high P_{rms} and W_{rms} values from the known path. In Fig. S3, the span values across the training and test sets are shown. Again, most paths are found with high accuracy. Those that are less accurate have inaccuracies in one or two mouths, but the central part of the path is found correctly in all cases, indicated by span values >58% in all cases. There are no significant differences in average P_{rms}, W_{rms}, and span between the training and test sets for our method, indicating that CHUNNEL was not overtrained to perform well only on the training set.

In Fig. 6 we compare the performance of our method with that of HOLE (13). HOLE in many cases performs poorly, often giving W_{rms} values of 6–10 Å and even $W_{rms} > 10$ Å, values that indicate partial or complete failure to find the path, respectively. In contrast, CHUNNEL gives $W_{rms} \leq 2$ Å for the majority of cases, indicating that the entire path is correct or that there is at most a small error in one of the mouths. In all other cases, CHUNNEL gives 2 Å $< W_{rms} \leq 7$ Å, usually from an error in following the wide mouths. In running, CHUNNEL identifies the plug points with the maximum Travel Out depth, i.e., point in the middle of a narrow part of each tunnel. Illustrating a possible way to combine both CHUNNEL and HOLE, these plug positions were used to initialize the latter. With this hint, HOLE produces values of $W_{rms} \leq 3$ Å for most of the paths. However, the results are no better than using CHUNNEL for both initiation and generation of paths. In summary, HOLE can perform well when given a hint from the plug generation from CHUNNEL, but in fact, getting to this point is really the bulk of the CHUNNEL algorithm. Once a good starting point is found for the tunnel, HOLE and CHUNNEL follow the paths out with similar accuracy.

Application to the porin membrane protein family

A likely use for our method is to predict the paths of tunnels in membrane proteins. The number of structures of membrane proteins determined through experimental methods, like those of the PDB database in general, is on the rise. The difficulties in obtaining structural data for membrane proteins are being overcome by various methods, and membrane proteins will likely become the focus of future structural genomics projects (32). We used part of a hand-collated database of membrane proteins (8), which, on October 1, 2007, had 278 structures representing 132 unique proteins. In this database, structures are broken down into groups based on fold and known function, which aids closer analysis. One such subgroup contains the porins, which provide the molecular basis for membrane permeability. These porins are found in bacteria and allow promiscuous or specific

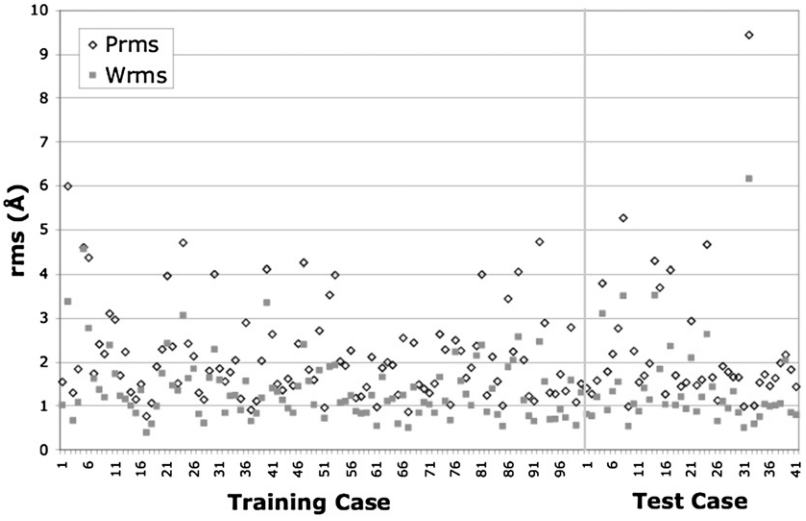

FIGURE 5 The best P_{rms} (*open diamond*) and W_{rms} (*gray square*) found by CHUNNEL for the 100 training cases and 41 test cases in the known pore set.

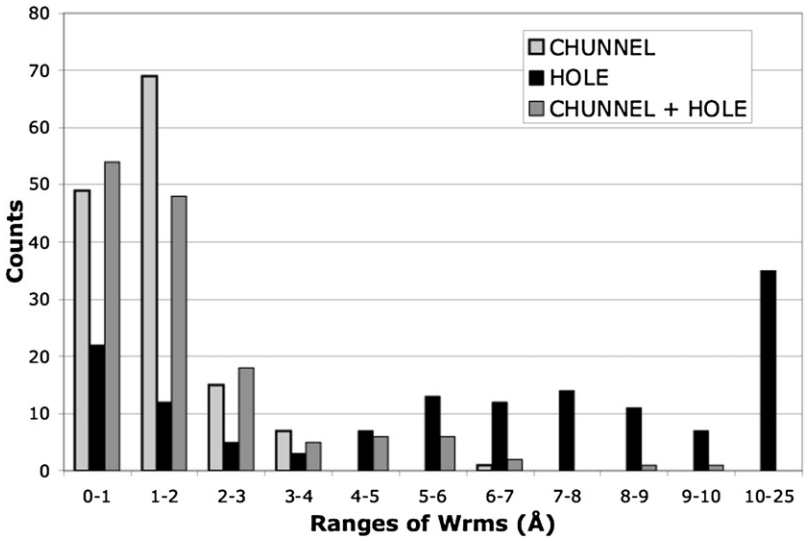

FIGURE 6 A histogram of weighted pore path error, W_{rms}, between CHUNNEL and HOLE using the combined known-pore training and test sets. Minimum P_{rms} path from CHUNNEL (*light gray*). HOLE, no hint (*black*). Minimum P_{rms} path from HOLE given several plug points with maximal Travel Out distance found using CHUNNEL (*medium gray*). Note that above 10 Å the results are put into a single bin.

transport through the outer membrane (33). CHUNNEL was used to analyze the porin family, as defined by the β-barreled porin fold (34). We examined the subset comprised of homotrimers plus structurally related monomers. In each case, the complete biological unit was examined. Overall, we examined 17 structures (8), including two structures that were analyzed with bound ligands and then again with the ligands removed, for a total of 19 cases (35–47). In five of these cases, the physiologically relevant tunnel was blocked by either a structural rearrangement, a peptide, or a ligand. Either no paths were found by CHUNNEL or nonphysiological paths were found with a very small minimum radius and length, instances where small adventitious pores are created by particular side-chain conformations near the surface of the protein. In the other 14 cases, the path with the largest minimum radius, ranked first by CHUNNEL, was the physiologically relevant and significant tunnel. Most of the structures are homotrimers, so there are three ''correct'' tunnels, which are all found by CHUNNEL.

It is interesting to note that when viewing the van der Waals representation of the homotrimeric porins, is the viewer sees a small gap in the middle of the trimer interface that appears to be a tunnel. However, because of the size of the solvent probe, there is no tunnel in the MS, and therefore, CHUNNEL does not find any paths through this middle region. The first tunnel found in each of the 14 successful cases has a minimum radius of between 1.4 Å and 4.3 Å. The low end of this range is PDB code 2O4V, a porin adapted to phosphate transfer, with the bound phosphate removed (47). This makes sense because phosphate is the smallest specific ligand bound to any of the porins that are not promiscuous transporters. The other bound ligands, once removed, have paths with larger minimum radii of 1.93 Å for glucose in 2MPR (44), 1.93 Å for malate in 2FGR (39), and 2.4 Å for sucrose in 1A0S (46). Of course the minimum tunnel radius is obviously not the only factor contributing to specificity in these cases, as many other nonspecific porins have tunnels with similar radii. The two cases of PDB codes 2IWV and 2IWW represent a pH-dependent folding change that blocks the pore (43). When it is unblocked, the minimum radius is 2.25 Å; when blocked, two paths formed by side chains on the exterior are found, but no paths are found through the pore.

To further illustrate the ease with which our code allows paths of related proteins to be compared, we compare three of these homotrimeric porins with a small minimum radius (1.9 Å) (38), a medium minimum radius (3.1 Å) (40), and a large minimum radius (4.3 Å) (36). The first found path for each is shown in Figs. S4 and S5. In Fig. S4, the radius is graphed against the distance from the beginning of the path, and the minimum point is easy to recognize. In Fig. S5, the structures with the found paths are shown in increasing size of minimum radius from top to bottom.

As a final example from the porin set, we analyzed the makeup of residues lining the entire tunnel and each choke point using the 14 nonblocked structures. A distance threshold of 4 Å from the radius of the pore was used to define lining residues. The enrichment factor for each residue was calculated as the fractional occurrence of that residue lining the path divided by its fractional occurrence over the entire 14 porin set. This is shown in Fig. S6. There is the expected enrichment of polar residues lining the pores, along with a notable enrichment of Arg, Tyr, Glu, and Pro residues at choke points.

Application to aquaporin

We also examined the integral membrane protein aquaporin (which is not a member of the porin family) using CHUNNEL, as this protein presents a challenge for structural analysis of this type because of its complexity and the small width of the water pores. Each of the four units has

a tunnel, and there is a central tunnel created among them (48). It is debatable whether or not the central tunnel has physiological importance, so it is important to catalog and compare all the tunnels. We used the aquaporin structure, PDB code 1J4N (49). In the analysis we found that because the water channels are very small, they are missed using the default CHUNNEL probe radius of 1.2 Å for surface generation. Hence, we used a smaller probe radius of 1.0 Å. However, this creates many small adventitious tunnels where side chains just barely touch, particularly on the cystoplasmic face of the structure, and a surface with 37 pores results. Many of the 37 pores result from the alternate mouths for all five important tunnels on the cytoplasmic side of the protein. Because of the hole-ridden cytoplasmic face of the surface and the different exit/plug combinatorics, one can generate hundreds of alternative pore-transiting paths from the half-paths produced by CHUNNEL. The central channel, formed by tetramerization, has a minimum radius of 1.97 Å. The four water channel paths found by CHUNNEL have minimum radii of 0.74 Å. Note that this minimum radius is lower than the probe radius used to construct the surface because of the finite resolution of the surface and volume discretization. These five paths are shown in Fig. S7.

Application to other transmembrane proteins

As a final application for CHUNNEL, we analyzed a larger set of transmembrane proteins. To do this, we used a set of 192 structures from the OPM database (17). These transmembrane structures were gathered from the PDB, and their positions within the membrane bilayer were calculated computationally and compared with experiment when possible (50). We chose the OPM database and methodology because it included not just α-helices but β-barrels as well, unlike some metrics, which were designed for helical transmembrane proteins only (51). We accessed this database and used the 192 transmembrane structures available on January 28, 2008. We removed waters and hetero atoms from the PDB files, which contain complete biological units (17). Our goal was, first, to generate all pore paths using CHUNNEL; second, to identify the subset of CHUNNEL paths that pass exactly once through the membrane bilayer, using the bilayer boundary information of Lomize et al.; and third, to identify tunnels that exit within the membrane bilayer. We presume that the bilayer-transiting pores would be of greatest physiological importance. The OPM data set also contains many structures for which no physiological path is expected to be found using the CHUNNEL method, including those involved in proton channels or proton pumps as well as GPCRs.

No information on the placement of these structures in the lipid bilayer is used in the CHUNNEL algorithm. This information is used to sort the found paths only after processing is complete. Note that, although the OPM methodology is limited to flat symmetric membranes, our analysis could be repeated for more general definitions of membrane barriers, for instance, by the use of elastic theory to define the lipid/water interface (52).

After processing of the OPM database with CHUNNEL, 284 membrane-transiting putative physiological paths were found in 52 unique structures, indicating that multiple tunnels are the rule rather than the exception (Table 2). After degeneracy of paths as a result of multimeric proteins is taken into account, there are 175 unique membrane-transiting paths in 52 unique monomers/proteins. In 28 of these structures, there is a single unique path per monomer. The mean length of these putative physiological paths is 126 $\pm$ 51 Å, much greater than the width of the membrane bilayer (usually 25–30 Å). There are two reasons for this. First, the paths must pass through not just the lipid barrier but the whole protein to reach the convex hull of the protein. Second, the paths are usually not straight, the data set having a mean winding

TABLE 2 Numbers of tunnels of various types in the OPM database

		Entire OPM	α-Helical	β-Barrel	NR25A[§]	NR25B[§]	Entire OPM, radius > 1.8	α-Helical, radius > 1.8	β-Barrel, radius > 1.8
	Total structures	192	140	52					
Putative physiological*	Paths	284	173	111	121	51	82	40	42
	Structures	52	26	26	19	14	35	19	16
One side exit[†]	Paths	1232	1199	33			284	274	10
	Structures	73	69	4			30	29	1
Two side exits[†]	Paths	446	415	31			87	84	3
	Structures	51	49	2			19	18	1
Side branch[‡]	Paths	108	86	22			63	55	8
	Structures	13	12	1			10	9	1

*Membrane-transiting.
[†]One or both ends of tunnel exit within bilayer.
[‡]Branch off a membrane-transiting path that exits within the bilayer.
[§]Nonredundant set with maximum 25% sequence similarity of proteins with α (NR25A) or β (NR25B) motif.

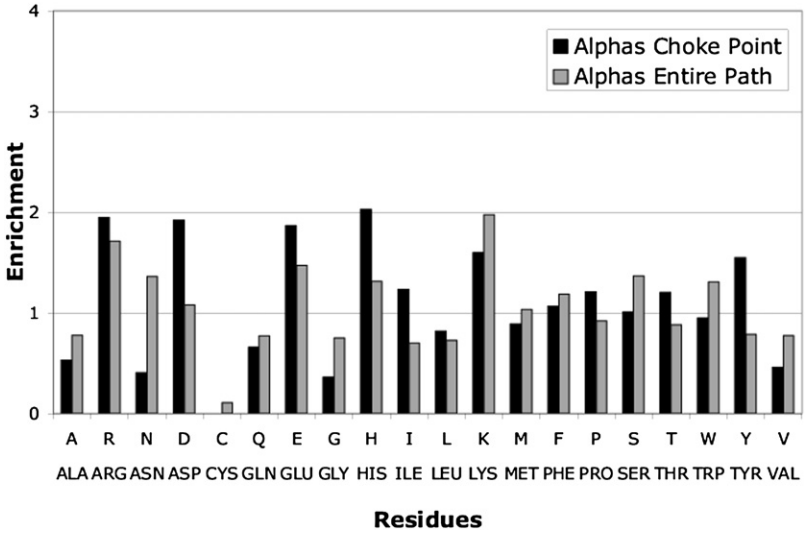

FIGURE 7 Residue enrichment for pores and choke points of α-helical motif proteins of the OPM database (17), pruned to 25% sequence similarity using PISCES (53).

metric of 1.68 $\pm$ 0.5. The path width minima over the set have a mean of 1.35 $\pm$ 1.8 Å, which is within the expected physiological range considering that 1.2-Å probes were used to construct these surfaces. Enrichments for residues found near the choke point and near the entire path were calculated relative to the residue composition of the entire OPM transmembrane database. These enrichments are shown in Fig. S8. There is an overall enrichment of the charged amino acids, particularly Arg, Glu, and, to a lesser extent, Lys, and an enrichment of the polar aromatic residue Tyr. For a finer analysis, the structures were split into either α-helical or β-barreled classes and pruned to a maximum of 25% mutual pairwise sequence identity using PISCES (53). The results are shown in Figs. 7 and 8. Removal of sequence-homologous duplicates insures that these graphs reflect real pore amino acid preferences, not just sequence conservation. The same four or five residues show enrichment, but interestingly, the degree of enrichment is much greater in the β-barrel class than the α-helical class.

Additionally, there are a surprisingly large number of paths, 4879, that do not pass through both membrane barriers once. This shows the power and importance of the membrane barrier data of Lomize et al. (17) in analyzing membrane protein pores. From this set of paths, we analyzed three interesting subsets: 1), those that start on one side of the membrane bilayer and emerge within the bilayer; 2), those that start and end within the bilayer; and 3), the branches of membrane-transiting putative physiological tunnels that terminate within the bilayer. Other classes of paths, such as those that lie entirely within a region on one side of the membrane, were not analyzed. Because we are also interested in paths that could potentially contain water, we separately identified tunnels whose minimum radius is greater than 1.8 Å, the commonly accepted upper limit on the size of a water molecule. The numbers of such tunnels and what kind of structures they are found in (α-helical or β-barrel) are summarized in Table 2. When we examine the data graphically, we notice that when side exits lie very close to the membrane surface,

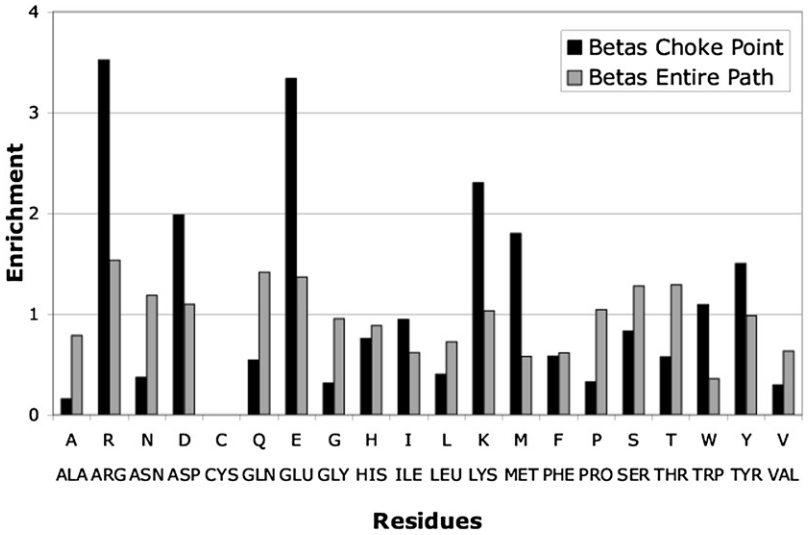

FIGURE 8 Residue enrichment for pores and choke points of β-barrel motif proteins of the OPM database (17), pruned to 25% sequence similarity using PISCES (53).

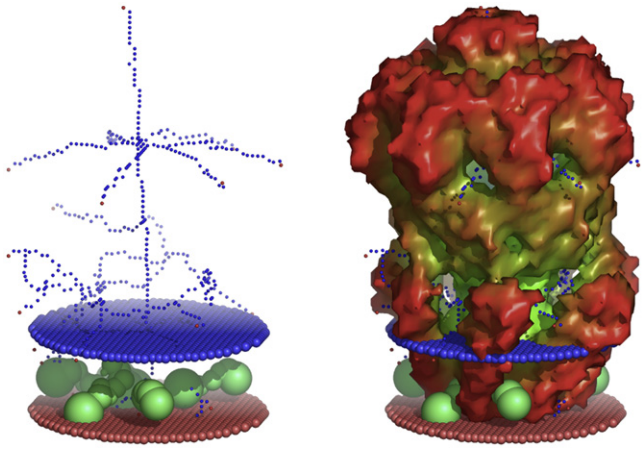

FIGURE 9 The MscS, PDB code 2OAU, shown with the membrane barriers in red and blue disks. The complete tree of paths is shown in blue spheres, the endpoints in red spheres. Some of the branched tunnels are shown in green. At left, no protein is shown for clarity; at right, the Travel Depth surface is shown.

they may exit the protein outside the membrane but reach the convex hull at a point inside the membrane, in which case they are classified as exiting inside the bilayer. The reverse situation also occasionally occurs. This introduces some ambiguity into the classification of intramembrane side exits and some degree of uncertainty in the numbers tabulated in Table 2. In specific proteins of interest, the ambiguity is easily resolved using graphic analysis.

The overall message from the data in Table 2 is that complicated tunnel topologies, defined as multiple membrane-transiting paths, paths with intramembrane exits, and branches with intramembrane exits, are not rare. For example, side tunnels and branched tunnels, although not ubiquitous, are quite common. Of particular interest is that they are much more common in α-helical domains than in β-barrel domains. A good example of a complicated tunnel structure is given by the MscS (1) in Fig. 9, which shows the complete tree structure of the tunnels and some of the intramembrane branched tunnels as well.

Preferences for residues lining intramembrane exiting and side-branching tunnels were also examined. The most interesting case appears to be the paths and choke points of the tunnels that branch off physiological tunnels that exit inside the membrane. Strikingly, a strong, fivefold enrichment for Trp is shown (Fig. 10). Even in the residue composition of the protein regions just within the membrane barriers, the enrichment of Trp in these branch paths is still over twofold, and near choke points, it is still almost 3.5-fold. It has been noted that in many membrane protein structures tryptophan is often found near the polar head group, and head-group/acyl chain interface regions of bilayers (51,54). Together, these data imply that side branches preferentially exit in this polar/apolar transition region of the membrane. Significant amounts of water within the membrane are also observed in the head-group/acyl chain interface region (55). It is thus likely that these side branch exits are accessible to some water.

DISCUSSION AND FUTURE WORK

We have presented here the implementation and testing of a new algorithm, CHUNNEL, to automatically find and characterize pores in proteins. The main contribution of CHUNNEL is its ability to identify and catalog all the tunnels through a given surface, which neither HOLE (13), CAVER (15,16), nor other work (3,14) could accomplish automatically. Although CHUNNEL is markedly slower than HOLE because of its complicated geometric and topological computations, the results are worth it for various applications. Moreover, complete automation is necessary for analyzing more than a handful of structures and for the membrane protein databases. These databases are growing at a steady pace, in part because of structural genomics projects (10). Our analysis of the transmembrane portion of the

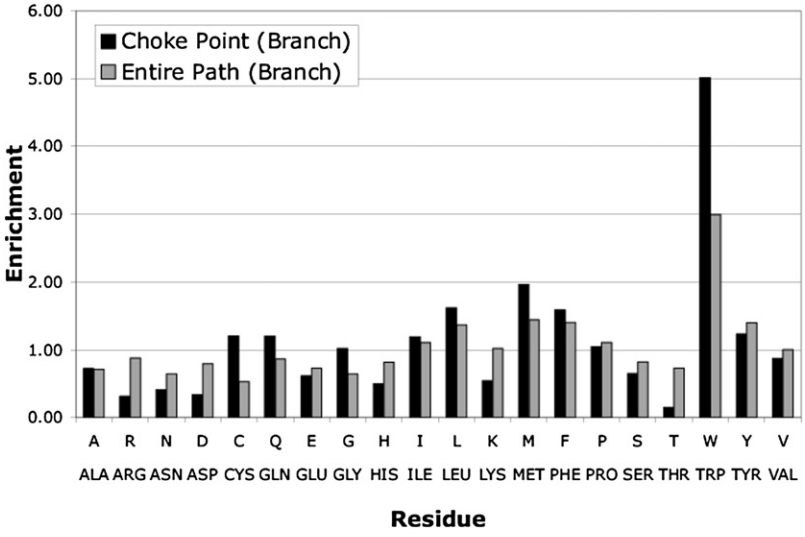

FIGURE 10 Enrichment of residues near the branches of putative physiological tunnels that exit into the membrane bilayer.

OPM database (17) is the first large-scale, automated analysis of channels that pass through the membrane barrier.

A second contribution of CHUNNEL is its ability to easily analyze structure and residue composition of the pores. Some studies on smaller classes of transmembrane proteins have been conducted, for instance, on aquaporins and related proteins (12,56). These studies highlighted the arginine/aromatic selectivity filter. Our results on a much larger OPM data set confirm this pattern of residue enrichment: Both arginine and tyrosine are highly enriched at choke points in the larger set, shown in Fig. S8. Arginine is also highly enriched in the choke points of the unrelated outer membrane porin family, shown in Fig. S6. We also partitioned membrane proteins of the OPM transmembrane database into the two α-helix and β-barrel motif subsets. The analysis of the residue enrichment shows significant differences between these two motifs (Figs. 7 and 8, respectively). The β-barrel motif has a less uniform distribution, showing stronger preferences for Arg, Glu, Lys, and Met than the α-helix motif. In other words the α-helix subset seems to favor a wider variety of amino acids in choke points than the β-barrel subset. The reasons for this marked difference in amino acid preferences with structural motif are unknown at this time. Possible factors include evolutionary and environment constraints because the β-barrel transmembrane proteins are found only (so far) in the outer membrane of bacteria. Because there are still a small number of nonhomologous proteins with transmembrane paths in either class (14 β-barrels, 19 α-helices), these difference may result in part from normal statistical fluctuations. As the database expands in the future, this question can be easily revisited thanks to the automated nature of CHUNNEL.

Another striking finding is the sheer number of tunnels and tunnel branches in membrane proteins, both membrane transiting and nontransiting. Although additional channels in the extramembrane portions of membrane proteins have been noted, to our knowledge, the analysis here is the first to draw attention to and analyze the multitude of intramembrane exiting channels. In part this is a consequence of HOLE's intrinsic design for finding linear tunnels: these side or branched tunnels would not be found with previous methods. Regarding the physiological importance of these additional tunnels and branches, this can be systematically evaluated based on the tunnel type:

1. Both exits in the aqueous phase and transiting the membrane once. This is the ''classical'' tunnel of putative physiological function, subject of numerous analyses. Presumably at least one such channel must exist in the open state for the protein to function. The exception is for proton or electron transport across the membrane, which can occur through ''wires'' or chains of donors and acceptors. Here, because of the small size of the permeant entity, no actual tunnel may exist.
2. Both exits in the aqueous phase, not transiting the membrane, i.e., confined to the extramembrane region on one side of the membrane. This is not likely to have any functional importance.
3. A branch off a membrane-transiting tunnel, with the exit in the aqueous phase. If the selectivity filter, or highest energy barrier controlling the flux is in the common part of the tunnel, before the branch, then the extra mouth is likely to have a small effect; otherwise, an extra branch would create a ''short-circuit.'' The extra entrance may, however, increase the probability of the substrate finding the channel, which at low concentrations could increase the rate. Multiple entrances may also play a role if multi-substrate interactions, such as ion-ion interactions, are important in conduction (57).
4. A branch off a membrane-transiting tunnel with the exit in the membrane interior. For an ionic or polar substrate, presumably the solvation penalty for exiting in the membrane, compared to the aqueous phase, is so high that conductance is minimal. Effectively the apolar part of the membrane plugs such leaks. This may explain why such tunnels are relatively common, as there is little evolutionary pressure for a protein to evolve a structure that is completely leakproof alone. However, there is a propensity for such tunnels to exit in the transitional region between acyl tails and head groups, where there is a significant amount of water. Thus, a sufficient degree of hydration to allow leakage currents cannot be ruled out. The existence of such water-filled side tunnels also has implications for the interpretation of membrane structure-probing experiments such as Cys-labeling and spin-labeling mapping of water-accessible and -inaccessible regions (58–60). Regions may be accessible to the probes but inside the membrane. Finally, because any such tunnels with a minimum radius of 1.8 Å or greater are presumably filled with water, this may play a role in the energetics and dynamics of substrate permeation, first, by providing an additional reservoir of water in the interior of the channel that could help hydrate ions. Because of the long-range nature of the electrostatic interaction, this water need not actually be touching the ion or even be in the main channel to be energetically significant. The energetic effects need not be limited to the permeant ion. Voltage sensing of channels requires that charge elements be moved in the membrane, and the energy of this would be affected by nearby water (61). Second, it may play a role in the energetics and dynamics by allowing water to flow in or out in response to substrate movement. In many cases the main channel is narrow enough that substrates and waters must move in file, requiring concerted movements and limiting conductance (57). Additional water passages ahead or behind the substrate could facilitate motion.
5. One or both exits inside the membrane region. These could play a role in allowing the interaction between membrane-soluble carriers and channel-permeant

species. Examples of the former include the apolar quinones that interact with the bc(1) complex (62).

Clearly more analysis of such epiphytic channels needs to be done in specific cases to investigate their functional importance.

Future work in this area includes calculation of additional metrics and pore properties, with the aim of possibly distinguishing nonphysiological tunnels from ion channels and pores from the structure in the absence of relevant experimental data. Although the influence of some geometric properties on various properties of tunnels, particularly ion channels (63), has been conducted, there is still much work to be done in this area, in part because the databases are still developing, in part from lack of fully automated, reliable pore finding. A single metric used here, the largest minimum radius, correctly identified the physiological tunnels in the porin set. However, a complete set of geometric features, as well as other physical features, will no doubt be necessary if we are to identify physiological tunnels of other classes of protein. In this regard, we point out that CHUNNEL, like HOLE and CAVER, does not provide much assistance in finding the paths of proton channels. Proton channels function in a different manner than ion channels in that the proton is not necessarily transferred through an open tunnel (64). Thus, reducing the probe radius is of no help.

CHUNNEL uses a set probe radius, chosen in advance to be smaller than the smallest permeant species of the channel(s) being analyzed. An interesting alternative is to use methods taken from the α-shape-filter idea (65). This would allow one to find a probe radius where the first topological tunnel emerges. However finding additional tunnels would require complete recomputation of the CHUNNEL procedure as the α-shape filter changes, effectively decreasing the probe radius and changing the entire surface. This is currently beyond practical computational capabilities. For this reason as well as reliance on previous code for surface generation, we currently implement a fixed, user-controlled probe radius parameter rather than an automated method.

Further work on both the algorithm and the implementation remains to be done. The quadratic dependence of the algorithmic complexity on the number of holes is acceptable but should be improved because the program can take hours to run if the surface has many holes. The worst combination is an extremely large complicated structure and a very small probe radius; these prove to be impractical to run on desktop workstations. Improvement here may also make the automated probe radius option discussed above feasible.

The methods developed here to find a topologically complete and geometrically distinct set of loops could prove useful in other applications. The ability to remove the handles from an N-torus and turn it into a topological sphere is a powerful method in many fields of computational geometry, for instance, to use spherical harmonic methods (66). Because our removals are done to cap tunnels roughly at their narrowest point, the caps are geometrically well placed. For other applications, it may be better to remove handles by cutting the handles at their narrowest point or possibly a mix of cutting handles and capping tunnels. For instance, removing each handle by doing the fewest changes would result in the closest thing to a topological sphere for a given protein surface, which would be useful for algorithms that only work on topological spheres, for instance mapping complicated topological spheres to geometric spheres (67).

In summary, we introduce a method, CHUNNEL, that automatically finds starting points and paths for all possible topological tunnels through a macromolecular surface. This improves on the mostly, but not completely automated methods of HOLE (13) and CAVER (15,16). Starting points found using our method can be used by these other methods as well; in fact, a hybrid approach may be advantageous for some applications. We show that we can find all known paths in a constructed data set of drilled tunnels and show examples and some overall analysis from a set of transmembrane proteins (17), including automatic identification of residues found near the tunnels or in the choke points.

SUPPLEMENTARY MATERIAL

Eight figures are available at http://www.biophysj.org/biophysj/supplemental/S0006-3495(08)00008-8.

R.G.C. thanks Michael A. Burr for discussions on checking whether a path passes through a given topological loop, and many others for discussions at several preliminary talks given on this work. Both authors wish to thank Drs. Carol Deutsch, Zhe Lu, Clay Armstrong, Kevin Foskett, and Qingyi (Joy) Yang for many useful discussions, as well as the reviewers for helpful comments. R.G.C. received partial funding for this project from the NIH Structural Biology Training Grant GM008275 and the Computational Genomics Training Grant T32HG000046 from the National Human Genome Research Institute. Funding for K.A.S. and R.G.C. was provided by NIH GM48130.

REFERENCES

1. Bass, R. B., P. Strop, M. Barclay, and D. C. Rees. 2002. Crystal structure of *Escherichia coli* MscS, a voltage-modulated and mechanosensitive channel. *Science*. 298:1582–1587.
2. Moarefi, I., D. Jeruzalmi, J. Turner, M. O'Donnell, and J. Kurlyan. 2000. Crystal structure of the DNA polymerase processivity factor of T4 bacteriophage. *J. Mol. Biol.* 296:1215–1223.
3. Voss, N. R., M. Gerstein, T. A. Steitz, and P. B. Moore. 2006. The geometry of the ribosomal polypeptide exit tunnel. *J. Mol. Biol.* 360:893–906.
4. Roll-Mecak, A., and R. D. Vale. 2008. Structural basis of microtubule severing by the hereditary spastic paraplegia protein spastin. *Nature*. 451:363–367.
5. Taylor, T. C., and I. Andersson. 1997. The structure of the complex between rubisco and its natural substrate ribulose 1,5-bisphosphate. *J. Mol. Biol.* 265:432–444.
6. Gilson, M. K., T. P. Straatsma, J. A. McCammon, D. R. Ripoli, C. H. Faerman, et al. 1994. Open ''back door'' in a molecular dynamics simulation of acetylcholinesterase. *Science*. 263:1276–1278.

7. Murray, J. W., and J. Barber. 2007. Structural characteristics of channels and pathways in photosystem II including the identification of an oxygen channel. *J. Struct. Biol.* 159:228–237.
8. White, S. H. 2007. Membrane proteins of known structure. http://blanco.biomol.uci.edu/Membrane_Proteins_xtal.html.
9. Hankamer, B., R. Glaeser, and H. Stahlberg. 2007. Electron crystallography of membrane proteins. *J. Struct. Biol.* 160:263–264.
10. Walian, P., T. A. Cross, and B. K. Jap. 2004. Structural genomics of membrane proteins. *Genome Biol.* 5:215.
11. White, S. H. 2004. The progress of membrane protein structure determination. *Protein Sci.* 13:1948–1949.
12. Bansal, A., and R. Sankararamakrishnan. 2007. Homology modeling of major intrinsic proteins in rice, maize and Arabidopsis: comparative analysis of transmembrane helix association and aromatic/arginine selectivity filters. *BMC Struct. Biol.* 7:27.
13. Smart, O. S., J. G. Neduvelil, X. Wang, B. A. Wallace, and M. S. P. Sansom. 1996. HOLE: A program for the analysis of the pore dimensions of ion channel structural models. *J. Mol. Graph.* 14:354–360.
14. Kelyweght, G. J., and T. A. Jones. 1994. Detection, delineation, measurement and display of cavities in macromolecular structures. *Acta Crystallogr. D Biol. Crystallogr.* 50:178–185.
15. Damborský, J., M. Petřek, P. Banáš, and M. Otyepka. 2007. Identification of tunnels in proteins, nucleic acids, inorganic materials and molecular ensembles. *Biotechnol. J.* 2007:62–67.
16. Petřek, M., M. Otyepka, P. Banáš, P. Košinová, J. Koča, et al. 2006. CAVER: A new tool to explore routes from protein clefts, pockets and cavities. *BMC Bioinformatics.* 7:1–9.
17. Lomize, M. A., A. L. Lomize, I. D. Pogozheva, and H. I. Mosberg. 2006. OPM: orientations of proteins in membranes database. *Bioinformatics.* 22:623–625.
18. Coleman, R. G., and K. A. Sharp. 2006. Travel depth, a new shape descriptor for macromolecules: application to ligand binding. *J. Mol. Biol.* 362:441–458.
19. Dijkstra, E. W. 1959. A note on two problems in connexion with graphs. *Numerische Mathematik.* 1:269–271.
20. Cormen, T. H., C. E. Leiserson, R. L. Rivest, and C. Stein. 2001. Introduction to Algorithms, 2nd ed. McGraw-Hill Higher Education, New York.
21. Feldman, J., and M. Singh. 2006. Bayesian estimation of the shape skeleton. *Proc. Natl. Acad. Sci. USA.* 103:18014–18019.
22. Larn, L., S.-W. Lee, and C. Y. Suen. 1992. Thinning methodologies-a comprehensive survey. *IEEE Trans. Patt. Anal. and Mach. Intel.* 14:869–885.
23. Reinders, F., M. E. D. Jacobson, and F. H. Post. 2000. Skeleton graph generation for feature shape description. *Eurographics-IEEE TCVG Symposium on Visualization. 73–82.*
24. Foskey, M., M. C. Lin, and D. Manocha. 2003. Efficient computation of a simplified medial axis. *J. Comput. Inf. Sci. Eng.* 3:274–284.
25. Nicholls, A., K. Sharp, and B. Honig. 1991. Protein folding and association: insights from the interfacial and thermodynamic properties of hydrocarbons. *Proteins Struct. Funct. Genet.* 4:281–296.
26. Bondi, A. 1964. van der Waals volumes and radii. *J. Phys. Chem.* 68:441–451.
27. Barber, C., D. Dobkin, and H. Huhdanpaa. 1993. The Quickhull algorithm for convex hull. *In:* Geometry Center Technical Report GCG53. University of Minnesota, Minneapolis.
28. Coleman, R. G. 2004. Finding knotted and linked vorticity lines in 3D vector fields. Master's Thesis. Tufts University, Medford, Massachusetts.
29. Wang, R., X. Fang, Y. Lu, and S. Wang. 2004. The PDBbind database: collection of binding affinities for protein-ligand complexes with known three-dimensional structures. *J. Med. Chem.* 47:2977–2980.
30. Wang, R., X. Fang, Y. Lu, C.-Y. Yang, and S. Wang. 2005. The PDBbind database: methodologies and updates. *J. Med. Chem.* 48:4111–4119.
31. Mardia, K. V. 1975. Statistics of directional data. *J Roy Stat Soc B Method.* 37:349–393.
32. Lundstrom, K. 2006. Structural genomics for membrane proteins. *Cell. Mol. Life Sci.* 63:2597–2607.
33. Nikaido, H. 2003. Molecular basis of bacterial outer membrane permeability. *Microbiol. Mol. Biol. Rev.* 67:593–656.
34. Koebnik, R., K. P. Locher, and P. Van Gelder. 2000. Structure and function of bacterial outer membrane proteins: barrels in a nutshell. *Mol. Microbiol.* 37:239–253.
35. Weiss, M. S., and G. E. Schulz. 1992. Structure of porin refined at 1.8 Å resolution. *J. Mol. Biol.* 227:493–509.
36. Kreusch, A., A. Neubüser, E. Schiltz, J. Weckesser, and G. E. Schulz. 1994. Structure of the membrane channel porin from Rhodopseudomonas at 2.0 Å resolution. *Protein Sci.* 3:58–63.
37. Dutzler, R., G. Rummel, S. Albertí, S. Hernández-Allés, P. Phale, et al. 1999. Crystal structure and functional characterization of OmpK36, the osmoporin of. *Klebsiella pneumoniae. Structure.* 7:425–434.
38. Zeth, K., K. Diederichs, W. Welte, and H. Engelhardt. 2000. Crystal structure of Omp32, the anion-selective porin from *Comamonas acidovorans*, in complex with a periplasmic peptide at 2.1 Å resolution. *Structure.* 8:981–992.
39. Zachariae, U., T. Klühspies, S. De, H. Engelhardt, and K. Zeth. 2006. High resolution crystal structures and molecular dynamics studies reveal substrate binding in the porin Omp32. *J. Biol. Chem.* 281:7413–7420.
40. Cowan, S. W., T. Schirmer, G. Rummel, M. Steiert, R. Ghosh, et al. 1992. Crystal structures explain functional properties of two *E. coli* porins. *Nature.* 358:727–733.
41. Baslé, A., G. Rummel, P. Storic, J. Rosenbusch, and T. Schirmer. 2006. Crystal structure of osmoporin OmpC from *E. coli* at 2.0 Å. *J. Mol. Biol.* 362:933–942.
42. Subbarao, G. V., and B. van der Berg. 2006. Crystal structure of the monomeric porin OmpG. *J. Mol. Biol.* 360:750–759.
43. Yildiz, O., K. R. Vinothkumar, P. Goswami, and W. Kühlbrandt. 2006. Structure of the monomeric outer-membrane porin OmpG in the open and closed confirmation. *EMBO J.* 25:3702–3713.
44. Meyer, J. E., M. Hofnung, and G. E. Schulz. 1997. Structure of maltoporin from Salmonella typhimurium ligated with a nitrophenyl-maltotrioside. *J. Mol. Biol.* 266:761–775.
45. Schirmer, T., T. A. Keller, Y. F. Wang, and J. Rosenbusch. 1995. Structural basis for sugar translocation through maltoporin channels at 3.1 Å resolution. *Science.* 267:512–514.
46. Forst, D., W. Welte, T. Wacker, and K. Diederichs. 1998. Structure of the sucrose-specific porin ScrY from *Salmonella typhimurium* and its complex with sucrose. *Nat. Struct. Biol.* 5:37–46.
47. Moraes, T. F., M. Bains, R. E. Hancock, and N. C. Strydnadka. 2007. An arginine ladder in OprP mediates phosphate-specific transfer across the outer membrane. *Nat. Struct. Mol. Biol.* 14:85–87.
48. Hedfalk, K., S. Törnroth-Horsefield, M. Nyblom, U. Johanson, P. Kjellbom, et al. 2006. Aquaporin gating. *Curr. Opin. Struct. Biol.* 16:447–456.
49. Sui, H., B. G. Han, J. K. Lee, P. Walian, and B. K. Jap. 2001. Structural basis of water-specific transport through the AQP1 water channel. *Nature.* 414:872–878.
50. Lomize, A. L., I. D. Pogozheva, M. A. Lomize, and H. I. Mosberg. 2006. Positioning of proteins in membranes: a computational approach. *Protein Sci.* 15:1318–1333.
51. Senes, A., D. C. Chadi, P. B. Law, R. F. S. Walters, V. Nanda, et al. 2007. Ez, a depth-dependent potential for assessing the energies of insertion of amino acid side-chains into membranes: derivation and applications to determining the orientation of transmembrane and interfacial helices. *J. Mol. Biol.* 366:436–448.
52. Choe, S., K. A. Hecht, and M. Grabe. 2008. A continuum method for determining membrane protein insertion energies and the problem of charged residues. *J. Gen. Physiol.* 131:563–573.
53. Dunbrack, R. L., Jr., and G. Wang. 2003. PISCES: a protein sequence culling server. *Bioinformatics.* 19:1589–1591.

54. Yau, W.-M., W. C. Wimley, K. Gawrisch, and S. H. White. 1998. The preference of tryptophan for membrane interfaces. *Biochemistry*. 37:14713–14718.

55. Wiener, M. C., and S. H. White. 1992. Structure of a fluid dioleoylphosphatidylcholine bilayer determined by joint refinement of x-ray and neutron diffraction data. III. Complete structure. *Biophys. J.* 61: 434–447.

56. Wallace, I. S., and D. M. Roberts. 2004. Homology modeling of representative subfamilies of arabidopsis major intrinsic proteins. classification based on the aromatic/arginine selectivity filter. *Plant Physiol.* 135:1029–1068.

57. Hille, B. 1992. Ionic Channels of Excitable Membranes. Sinauer Associates, Sunderland, Massachusetts.

58. Chen, H., F. Sesti, and S. A. N. Goldstein. 2003. Pore- and state-dependent cadmium block of IKs channels formed with MinK-55C and wild-type KCNQ1 subunits. *Biophys. J.* 84:3679–3689.

59. Yellen, G., D. Sodickson, T. Y. Chen, and M. E. Jurman. 1994. An engineered cysteine in the external mouth of a K^+ channel allows inactivation to be modulated by metal binding. *Biophys. J.* 66:1068–1075.

60. Gross, A., and W. L. Hubbell. 2002. Identification of protein side chains near the membrane-aqueous interface: a site-directed spin labeling study of KcsA. *Biochemistry*. 41:1123–1128.

61. Pathak, M., L. Kurtz, F. Tombola, and E. Isacoff. 2004. The cooperative voltage sensor motion that gates a potassium channel. *J. Gen. Physiol.* 125:57–69.

62. Akiba, T., C. Toyoshima, T. Matsunaga, M. Kawamoto, T. Kubota, et al. 1996. Three-dimensional structure of bovine cytochrome bC1 complex by electron cryomicroscopy and helical image reconstruction. *Nat. Struct. Biol.* 3:553–561.

63. Beckstein, O., and M. S. P. Sansom. 2004. The influence of geometry, surface character and flexibility on the permeation of ions and waters through biological pores. *Phys. Biol.* 1:42–52.

64. Nagle, J. F., and H. J. Morowitz. 1978. Molecular mechanisms for proton transport in membranes. *Proc. Natl. Acad. Sci. USA*. 75:298–302.

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

66. Kazhdan, M., T. Funkhouser, and S. Rusinkeiwicz. 2003. Rotation invariant spherical harmonic representation of 3D shape descriptors. *Symp. Geom. Proc.* 167–175.

67. Rahi, S. J., and K. A. Sharp. 2007. Mapping complicated surfaces onto a sphere. *Int. J. Comput. Geom. Appl.* 17:305–329.

68. The PyMOL Molecular Graphics System. DeLano Scientific, San Carlos, California.