Geometrical Detection of Pathways in Protein Structures Leading Among More Binding Sites

Ondřej Strnad, Vilém Šustr, Barbora Kozlíková and Jiří Sochor
Faculty of Informatics
Masaryk university, Brno, Czech Republic
Email: xstrnad2@fi.muni.cz, xsustr@fi.muni.cz, xkozlik@fi.muni.cz, sochor@fi.muni.cz

***Abstract*—In this paper, we present a novel algorithm for the detection of pathways connecting two or more specific user defined binding sites, which are deeply buried in a protein macromolecule. These pathways can play an important role in the protein reactivity and overall behavior. However, our new algorithm can be generalized and used for computation of pathways inside an arbitrary set of spheres in three-dimensional space, leading through an ordered set of user-defined sites. Our approach is based on the localized Voronoi diagram approach and the Delaunay triangulation. The greatest benefit of our approach is its independence on the size of the input data set. This is achieved by using only a subset of all atoms in the macromolecule in each phase. This substantially reduces the size of the processed space. The method can also be utilized for determination whether pathways wide and straight enough exist among determined binding sites. This information then serves as the guideline for assessing the migration of products of chemical reaction between these binding sites.**

Keywords–protein; ribosome; tunnel; channel; active site; binding site; Voronoi diagram; Delaunay triangulation

I. INTRODUCTION

Proteins are organic molecules, which are irreplaceable for every life form on our planet. They perform specific tasks and are involved in many vital processes in living organism's cells. Analysis of proteins helps to understand their complex structure and chemical principles. In the quickly evolving area of protein engineering, where new substances like drugs and inhibitors are being designed, there are strong needs to explore possibilities of protein modification via transporting ligands to the desirable spots (active sites) inside the protein structure, causing a chemical reaction. However, verification of each potential reaction by an experiment in a laboratory is extremely time-consuming. Based on this fact, heuristic methods helping to exclude impossible cases and to suggest highly probable variants are utilized for significant decrease of in vitro experiment time. Methods that have been developed so far are mostly based on the computational geometry approach. Most of them are stochastic and present an approximation to the real situation, but as proven by numerous experiments and comparisons performed by the biochemists, they still provide an essential information to the biochemical community.

These methods concentrate on analysis and detection of specific inner structures inside molecules, such as cavities, pockets, tunnels, channels, pores, etc. Long-term research of biochemical properties of proteins revealed that the protein reactivity strongly depends on the presence of specific voids (cavities) inside the molecule, called active sites [1]. These active sites subsequently represent the destination of small ligand molecule that is transported from the outside solvent via pathways called tunnels. The ligand can then react with atoms surrounding the active site and the product of such reaction can serve as a basis of new medications or other important chemical compounds. The presence of tunnels on proteins along with particular examples is thoroughly described in Chapter 17 [2].

Until now, we have concentrated on the detection of tunnels from one active site. But, some enzymes contain two or even more active sites located on separate domains or subunits and all of them can participate on the final product structure. The product of the chemical reaction between the ligand and the protein in the first active site can travel to the second active site. Here, another reaction can modify this product. The direct transfer of ligands between active sites prevents the release of labile substrates into the outer solvent or the entrance of other intermediates, which can compete with already present ligands. Moreover, using the intramolecular tunnels, biochemists are able to regulate a set of consecutive chemical reactions [3].

The first tunnel connecting distinct active sites was discovered for tryptophan synthase from Salmonella typhimurium [4], which contains two active sites. It is obvious that in this case we have to detect not only tunnels connecting the active site with the outside solvent but also intramolecular tunnels between two and more active sites. This situation demands introducing a new solution to tunnel computation, which is the main goal of this article. Intramolecular tunnels are commonly present in ammonia-transferring enzymes. Many such enzymes have been already studied in some detail [5], [6], including the following structures – carbamoyl phosphate synthetase (PDB ID of wild type (WT) 1BXR), glucosamine 6-phosphate synthase (WT 2J6H), glutamate synthase (WT 1OFD), imidazole glycerol phosphate synthase (WT 1KA9) or cytidine triphosphate synthetase (WT 1VCM).

This algorithm can also be generalized and utilized for user-driven tunnel computation, when the biochemists can, according to their prior knowledge, determine some important spots inside the protein that should the tunnel follow.

When dealing with molecules consisting of thousands of atoms, the analysis of their structure at once is possible. However, for large macromolecular structures containing hundreds of thousands of atoms, such as ribosomes, the original approach is inapplicable. But, when dividing the area of interest into smaller regions and computing the tunnel piecewise, we can obtain acceptable and relevant results.

 ISBN: 978-1-61208-335-3

II. Related work

In this section, we concentrate on existing approaches to the computation of various structures (some of them were mentioned in the first chapter). The first group of algorithms aims to calculate the molecular surface. This apparently unrelated structure has two practical exploitations for our purpose. First of all, it can contain so called pockets. The difference between a pocket and a tunnel is that the binding site of a pocket is directly accessible from the surface, whereas the binding site of a tunnel is deeply buried inside the protein. The basic idea of the algorithm is rolling the probe of user-specified radius over the protein. The probe touches the protein boundary atoms and from all probe positions the resulting molecular surface can be constructed. This idea describes the Alpha shape theory [7] and the Reduced surfaces theory [8]. As mentioned above, a suitable probe size is able to reveal pockets on the protein surface. However, this approach is not applicable to the binding sites, which are deeply buried in the protein structures. Such binding sites cannot be accessed by the rolling probe. Some extended approaches [9] or other techniques that compute other pathways for accessing the deeply buried binding sites, e.g., tunnels or channels have to be involved.

In general, detection of tunnels in proteins can be considered as a specialized path detection algorithm. The environment is formed by a set of obstacles (atoms), which are scattered in the 3D space and may overlap. For this problem, many approaches of path planning have been introduced so far. When selecting an appropriate solution, we have to take into account its extendability by involving time changes into the input set. It corresponds to molecular dynamics, when the inner forces and outer environment causes permanent movement of atoms. Our solution of the static case, which is described in this article, was designed with respect to this knowledge.

Most of the existing path planning algorithms are mainly designed for environments, where the obstacles are static. Examples of such approaches can be the Minkowski sum [10], visibility graphs [11], or cell decomposition [12]. In 3D space, the complexity increases significantly; but, several useful algorithms have been also introduced. Again, solutions for static environments are more common, such as the spatial indexing [13] or the velocity obstacles method [14]. Algorithms for 3D dynamic environment, which would satisfy our problem with molecular dynamics, are quite rare. General methods, such as [15], can be mentioned as an example. However, these solutions contain some input constraints or were not designed for large datasets. Also in the field of protein analysis, there is an input set of initial requirements, which have to be fulfilled by the algorithms. Two sophisticated techniques for the detection of tunnels in dynamics were designed [16][17].

The first algorithm designed primarily for the detection of tunnels in static environment was based on the grid method [18]. The key idea is enclosing the whole protein into a bounding box, which is then regularly sampled into a set of cubes (grid). Thanks to the dependence on the grid resolution, the results of this algorithm may vary substantially. Also its time complexity ($\mathcal{O}(n^3)$ with respect to the number of samples) is inapplicable to larger molecular systems. So, this method was successively replaced by Voronoi diagram-based approach [19][20].

Nowadays, the leading approaches in tunnel and channel detection are based on construction of the Voronoi diagram (VD) and its dual structure, the Delaunay triangulation (DT). The construction of DT is independent on any initial settings – the complexity of the QuickHull algorithm for DT computation is expected to be $\mathcal{O}(n \cdot log(n)$ [21]. According to the duality principle, the VD is obtained from the DT in $\mathcal{O}(n))$. Thanks to this time complexity, it is applicable to molecular dynamics in reasonable computational time.

The basic idea is as follows. For the whole protein structure, the DT for all centers of atoms is computed (for example using 4D Quick-Hull algorithm [21]). Afterwards, it is converted to its dual VD. However, atoms in the protein may be of different radii. In most cases, this fact is not taken into account [20], [22] and the resulting tunnels are still biochemically relevant. Of course, involving atom radii to the computation increases the preciseness. Few recent approaches take into account the different radii of atoms – by introducing some approximation [23] or implementing the additively weighted Voronoi diagram (AWVD) (see [24]). AWVD splits the space more naturally, thus the results are more accurate. But, it also leads to higher time and memory complexity.

After the VD or AWVD is constructed, it is converted into a graph. In this graph, each Voronoi vertex is represented by a node and connections of Voronoi vertices form edges. Every edge is then evaluated by a number representing the radius of maximal sphere with the center located on this edge and non-colliding with any atom of the protein (Figure 1). In the last phase, a search algorithm (such as Dijkstra or A*) is initiated. As a result, a set of collision-free paths is obtained. Note that the additional search criteria resulting in the detection of the shortest or the best-evaluated path can be applied (in our case the widest tunnels).

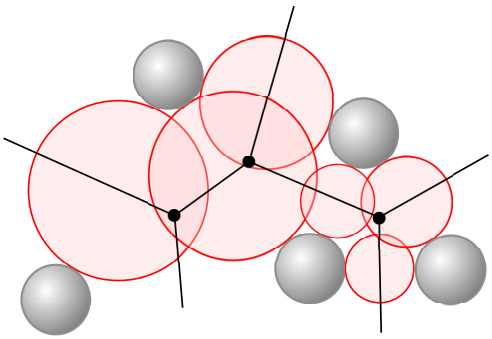

Figure 1: Description of the empty space inside the protein expressed by spheres. Spheres with maximized radius not intersecting any of atoms are centered on the edges of Voronoi diagram.

The main advantage of this VD-based method lies mainly in its precision. But still, the running time and memory consumption increase rapidly when being applied on large structures, such as ribosomes. In these cases, our newly designed approach is able to divide the structure into smaller parts which are then processed individually.

To our best knowledge, the above described methods were by now applied to the computation of tunnels leading from a

 ISBN: 978-1-61208-335-3

binding site to the outer solvent. However, our solution aims to detect intramolecular tunnels connecting several deeply buried binding sites.

III. ALGORITHM

Our new method presented in this paper focuses, generally, on the detection of pathways of guaranteed minimal width connecting sites of interest inside a set of points. To demonstrate the applicability of this general approach, we customized it for the detection of tunnels among more active sites in large macromolecular structures, e.g., ribosomes.

The input set consists of points in 3D space (representing the centers of atoms). Users have to define points of interest – starting and ending points of each tunnel local segment (e.g., chemically relevant binding sites inside the ribosome). Previously described algorithms were designed for searching for arbitrary paths from the active site to the molecular surface. Users had no control over the trace of the computing tunnels. They could only choose the most suitable and relevant tunnels from the computed set. Using this new approach, users can control the trace of the tunnel by determining the desired spots, through which the tunnel should traverse.

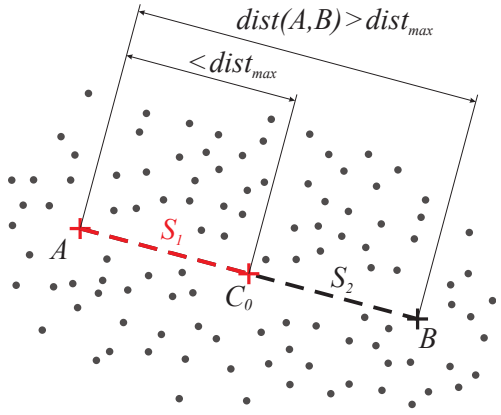

Figure 2: A plan for start-to-end connection, pivot segments S_1, S_2

By default, the criteria of our algorithm are set for searching "straight" tunnels (more straight tunnels are often also more biochemically relevant). However, this requirement can be easily customized by users and easily adapted to compute the widest tunnels or tunnels fulfilling other properties or their combination.

Require: points A, B, distance $dist_{max}$
  if $dist(A, B) > dist_{max}$ **then** {split long segment}
    $S \leftarrow \{[A, C_0], ..., [C_n, B]\}$
  else
    $S \leftarrow \{[A, B]\}$
  end if
  return S

Figure 3: Preprocessing phase: construction of the set of pivot segments

In the preprocessing phase (Figure 3), the algorithm evaluates the assignment and prepares a set of individual tasks for searching paths along shorter segments. Afterwards, this set is iteratively processed in four phases and pathways along the determined pivot segments are computed. Finally, individual paths are connected to form the resulting pathway – tunnel.

As the input, the algorithm requires a set of points in 3D P, starting A and ending B points. All those points have to be located inside the convex hull of the input set P. To overcome memory limitations and to avoid the calculation of large Voronoi diagram, large pivot segments are split into shorter segments. The distance $dist(A, B)$ between points A and B is compared to the user defined value $dist_{max}$. If $dist(A, B)$ is larger, it is split to the set of shorter *pivot segments* $S_i, i = 1, ..., m$ of equal lengths, see Figure 2. Each *pivot segment* S_i is represented by its starting point $A(S_i)$ and ending point $B(S_i)$, i.e., for the case when $|C_j| = 0$ there is only one *pivot segment* $S_1 = [A, B]$, when $|C_j| = 1$ there are two segments $S_1 = [A, C_0]$ and $S_2 = [C_0, B]$ and so on. The algorithm finds a path connecting $A, C_0, ..., C_n, B$. Because the connection between every two subsequent points is computed similarly, in the following paragraphs we concentrate only on the computation of the path between A and B ($|C_j| = 0$).

Phase 1: Selection of relevant points

Require: set of points P, pivot segment S_i, user-defined distance r
  $P_i \leftarrow$ empty
  for $p \in P$ **do**
    if dist$(p, S_i) \leq r$ or dist$(p, A(S_i)) \leq 2r$ or dist$(p, B(S_i))$ $\leq 2r$ **then**
      $P_i \leftarrow p$
    end if
  end for
  return P_i

Figure 4: Phase 1: Selection of relevant points

The iteration process of the algorithm starts with selecting of relevant atoms (Figure 4). During the computation of a partial path along pivot segment S_i, only the set of relevant atoms P_i is taken into account. Relevant atoms are those having the distance to the segment S_i lower than r or the distance from $A(S_i)$ or $B(S_i)$ lower than $2r$, where $r > 0$ is a user-defined distance (Figure 5). Such a construction assures that the starting or ending points of each segment are always enclosed inside the set P_i. At the same time, according to the current value of parameter r, the massive reduction of input data is achieved. On the other hand, the lower the ratio r, the narrower the resulting pathway could be or it may not even be detected.

Phase 2: Computation and refinement of the Delaunay triangulation

From the set P_i using the Quick Hull algorithm [21], the Delaunay triangulation $T(P_i)$ is computed (Figure 6). In the three dimensional space, the triangulation is formed by tetrahedra. Since we compute the triangulation only for a limited set of atoms P_i, there may be differences between the triangulation computed for this subset and the triangulation

 ISBN: 978-1-61208-335-3

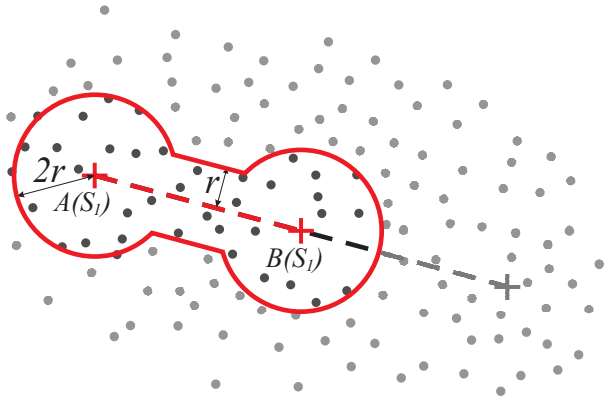

Figure 5: Selection of relevant points P_i.

T computed for the whole protein. The differences occur mainly on the boundaries, where narrow tetrahedra forming the hull appear. These narrow tetrahedra would misrepresent the resulting tunnel. Therefore, the triangulation $T(P_i)$ must be refined to include only tetrahedra that are also contained in $T(P)$. All boundary tetrahedra from $T(P_i)$ must be checked whether they satisfy the Delaunay condition in the context of the whole protein. I.e., for a tangent sphere constructed for every triple from P_i no other point from P is allowed to lie inside such a sphere (Figure 7). If a point lies inside the tangent sphere, the old tetrahedron has to be replaced by two newly constructed tetrahedra. The refinement is processed incrementally and to prevent the worst case – traversing all points from P – we stop it when there is no unchecked atom within the distance lower than $3r$ from the segment S_i. The rest of tetrahedra that do not fulfill the Delaunay condition are simply removed from the triangulation.

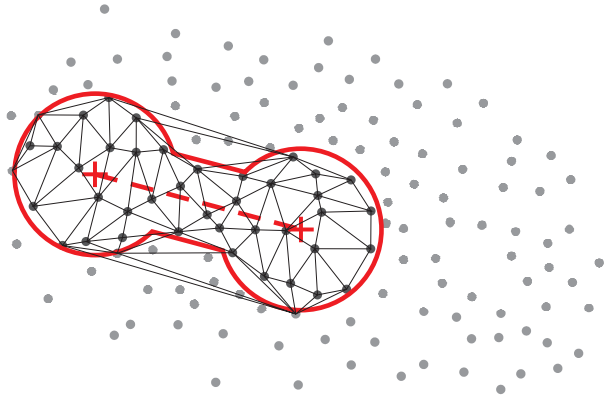

Figure 6: Local triangulation of $T(P_i)$.

Phase 3: Conversion to graph and evaluation

When the triangulation $T(P_i)$ is computed, it is converted into a graph $G(P_i)$ in the same way as described in [20]. The duality between the Delaunay triangulation (DT) and the Voronoi diagram (VD) is employed, i.e., every tetrahedron represents a Voronoi vertex and every shared face of two tetrahedra represents a Voronoi edge. The graph $G(P_i)$ is then constructed in the following steps. Every node N in graph $G(P_i)$ represents a Voronoi vertex from triangulation $T(P_i)$ and stores a reference to its dual tetrahedron $tetra(N)$. For each two nodes whose referenced tetrahedra share a face, a new edge e_k is added into the graph $G(P_i)$. In the last step, the weight $w(e_k)$ of every edge e_k is computed as

$$w(e_k) = \frac{length(e_k)}{dist(e_k)^3} \quad (1)$$

where $length(e_k)$ is the geometrical length of the edge e_k and $dist(e_k)$ is the distance from e_k to the surface of the nearest atom from P_i. The power of 3 was determined experimentally and it means that shorter and wider edges in graph are preferred to longer and narrower ones.

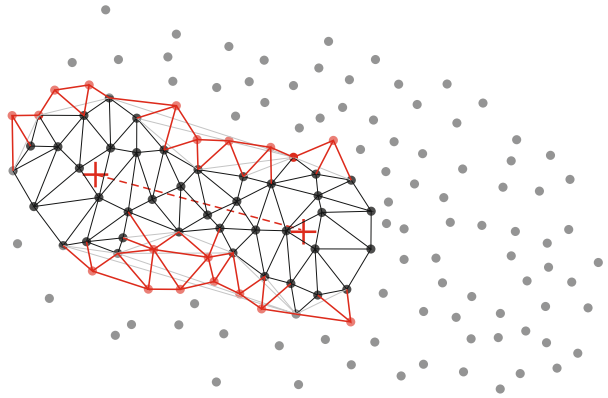

Figure 7: Local triangulation of P_i refinement – every tetrahedron has to satisfy the Delaunay condition.

Phase 4: Traversing the graph

Before the start of the searching algorithm, the starting nodes $A(N)$ and ending nodes $B(N)$ from $G(P_i)$ have to be determined. Firstly the nodes $A(N)$ or $B(N)$ whose referenced tetrahedra are enclosing the starting point $A(S_i)$ or $B(S_i)$ are selected. However, the weight $w(e)$ of all edges originating from the node $A(N)$ or $B(N)$ may be low, i.e., causing the starting or ending site inaccessible for the probe of a selected size. To prevent these cases, in places where the path is completely blocked at the starting or ending node, nodes $A(N)$ and $B(N)$ are selected (optimized) in the following way. Let Y be a set of nodes whose referenced tetrahedra $tetra(N)$ are accessible ($tetra(N)$ and $tetra(A(N))$ share a face) from $tetra(A(N))$ at most in 3 hops. From the set Y a node, from which the edge e_k with the greatest $w(e_k)$ originates, is selected as $A(N)$. The ending node $B(N)$ is determined similarly.

The optimization is not performed, when $A(S_i) \in \{C_0, ..., C_n\}$ or $B(S_i) \in \{C_0, ..., C_n\}$. In other words, $A(S_i)$ or $B(S_i)$ can be optimized only if it was created when splitting the long pivot segments by $createSegments$ method and not set by the user. This ensures that if the path is found it always leads through the user specified points of interest. However, this restriction may be loosen to allow more flexible path searching.

 ISBN: 978-1-61208-335-3

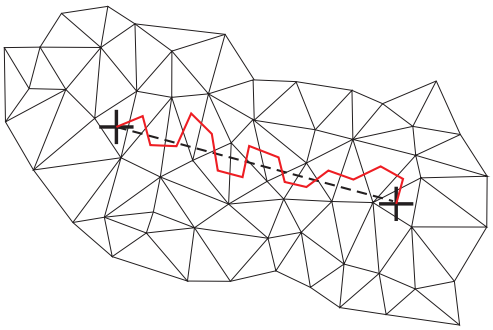

Figure 8: Resulting pathway in segment S_1.

After the nodes $A(N)$ and $B(N)$ are known, the A* search algorithm [25] is launched. As the result, the best weighted path $path(S_i)$ of segment S_i is obtained (Figure 8).

Construction of the resulting path

After all the segments S_i are processed, a set of paths is obtained. As the last step, all paths $path(S_i)$ have to be concatenated in order to form one continuous path connecting A with B. Since the ending point $B(S_i)$ of S_i is equal to the starting point $A(S_{i+1})$ of S_{i+1}, the connection process is straightforward. The pseudocode (Figure 9) presents the overview of the whole algorithm.

Require: set of 3D points P, points A and B
  $S \leftarrow createSegments(A, B)$
  $T \leftarrow$ empty list of tunnels
  for $S_i \in S$ **do**
    $P_i \leftarrow selectRelevantPoints(P, S_i)$
    $T(P_i) \leftarrow$ Delaunay triangulation of P_i
    adjust $A(S_i)$ and $B(S_i)$
    refine $T(P_i)$
    $G(P_i) \leftarrow$ convert $T(P_i)$
    evaluate all edges in $G(P_i)$
    $A(N) \leftarrow$ select starting node from $G(P_i)$
    $B(N) \leftarrow$ select ending node from $G(P_i)$
    $path(S_i) \leftarrow$ search algorithm($G(P_i)$, $A(N)$, $B(N)$)
    $T \leftarrow path(S_i)$
  end for
  $path(A, B) \leftarrow concat$(T)
  return $path(A, B)$

Figure 9: Tunnel connecting two points of interest

IV. Results and Discussion

In the testing and verification phase, this algorithm was launched on protein structures of various sizes and inner arrangement. We will mention only several examples: epoxide hydrolase with PDB ID 1CQZ, acetylcholinesterase complexed with huperzine A (PDB ID 1VOT), haloalkane dehalogenase (PDB ID 2HAD), GroEL-GroES-(ADP)7 chaperonin complex (PDB ID 1AON) and ribosome indexed as 70S_RF2. The binding sites for the input were taken from the CSA database [1] or, when not present, selected by biochemists according to their prior knowledge or by analyzing inner cavities (coming out from the fact that each binding site is enclosed in an inner cavity).

In each case, the algorithm was able to compute a suitable pathway between selected binding sites. Computation of pathways connecting three active sites of 1CQZ (selected from the CSA database) can be seen in Figure 10 (left). Figure 10 (right) shows the capability of our solution to detect intramolecular tunnels in large ribosomal structure 70S_RF2. This particular case presents four manually selected points as an input.

The algorithm was implemented in Java as a single-thread process. On a common 2.5GHz computer with 32-bit operating system the running time for 1CQZ (8218 atoms) was approximately 8 seconds whereas for 70S_RF2 (149412 atoms) was 45 seconds. This measurement was done with experimental settings of parameters ($dist_{max} = 50$Å, $r = 8$Å).

In the worst case, finding a path among two points on opposite sides of the structure and with the specific shape of the structure, the algorithm has to process all atoms in one step. But, for the computation of paths in Figure 10 (left) only approx. 32% and for 70S_RF2 only approx. 18% of all atoms were taken into account. The algorithm starts with the selection of relevant atoms for the current segment which is done in $\mathcal{O}(n)$. Then, the Delaunay triangulation using the QuickHull algorithm is obtained in worst in $\mathcal{O}(n^2)$ ($\mathcal{O}(n \cdot log(n)$ expected [21]). Converting DT to a graph and evaluation of all edges is done in $\mathcal{O}(n)$. The last step is the A* search algorithm which finds the path in worst in $\mathcal{O}(n^2)$ (better complexity expected since it is heavily dependent on the heuristic function used).

V. Conclusion

In this paper, we presented a novel algorithm, which is able to determine non-colliding pathways among a set of spheres randomly scattered in the 3D space. This algorithm can be applied for solving a general path planning problem and it is based on the computational geometry. To demonstrate its application, we adopted this solution to detect intramolecular tunnels in protein macromolecules. The curvature of the detected tunnels and the preciseness of the algorithm is dependent on initial settings. By changing the input parameters biochemists are able to reach the desired pathways. Our solution is equipped with simple and user friendly interface where users can change the input parameters and thus influence the size, shape and curvature of resulting tunnel. The algorithm is not limited to the size of the input data set because only a subset of necessary atoms is required in every phase of computation. Paths computed by our algorithm can also reveal potentially interesting places of protein structures that lie on the path between two or more active sites.

VI. Future work

Our future research in this area will concentrate on an extension of this algorithm to process molecular dynamics, computation of multiple tunnels between more binding sites and more automatized settings of initial parameters. Moreover, our aim is also to decrease the computational time of the algorithm. This can be reached by parallelization of the algorithm. In fact, because all path segments are identified once before the main computation starts, we can process each segment in different processing unit independently.

 ISBN: 978-1-61208-335-3

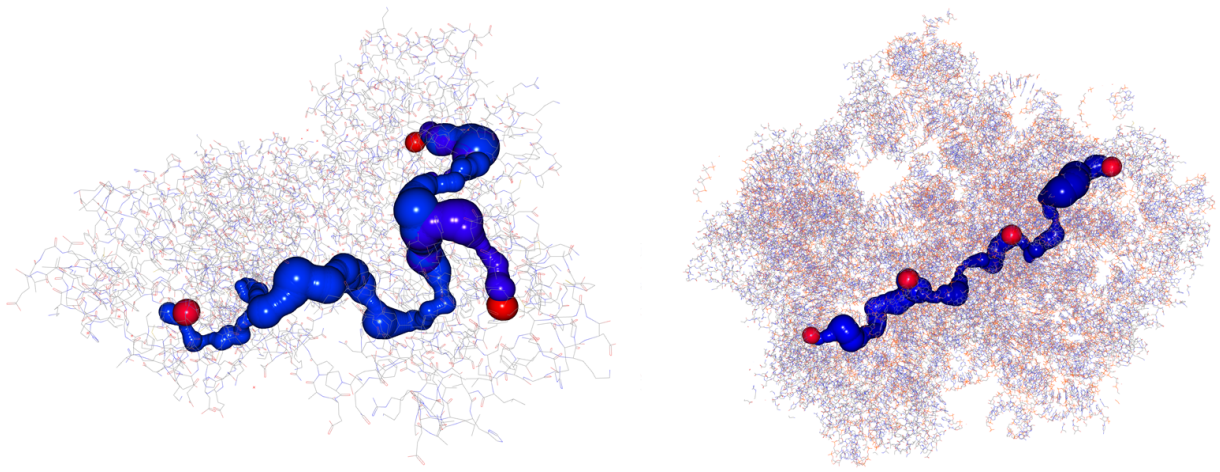

Figure 10: Left - path among 3 known binding sites in the 1CQZ (8218 atoms) protein structure. Right - path among 4 custom sites in 70S_RF2 structure (149412 atoms).

REFERENCES

[1] N. Furnham, G. L. Holliday, T. A. P. de Beer, J. O. B. Jacobsen, W. R. Pearson, and J. M. Thornton, "The catalytic site atlas 2.0: cataloging catalytic sites and residues identified in enzymes," *Nucleic Acids Research*, vol. 42, no. D1, pp. D485–D489, 2014. [Online]. Available: http://nar.oxfordjournals.org/content/42/D1/D485.abstract [retrieved: 03, 2014]

[2] S. Lutz and U. Bornscheuer, *Protein Engineering Handbook*, ser. Protein Engineering Handbook. Wiley, 2012, no. sv. 3.

[3] E. W. Miles, S. Rhee, and D. R. Davies, "The molecular basis of substrate channeling," *Journal of Biological Chemistry*, vol. 274, no. 18, pp. 12 193–12 196, 1999.

[4] C. C. Hyde, S. A. Ahmed, E. A. Padlan, E. W. Miles, and D. R. Davies, "Three-dimensional structure of the tryptophan synthase alpha 2 beta 2 multienzyme complex from Salmonella typhimurium," *J. Biol. Chem.*, vol. 263, no. 33, pp. 17 857–17 871, Nov 1988.

[5] A. Gora, J. Brezovsky, and J. Damborsky, "Gates of enzymes," *Chemical Reviews*, vol. 113, no. 8, pp. 5871–5923, 2013. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/cr300384w

[6] F. M. Raushel, J. B. Thoden, and H. M. Holden, "Enzymes with molecular tunnels," *Accounts of Chemical Research*, vol. 36, no. 7, pp. 539–548, 2003.

[7] H. Edelsbrunner and E. P. Mucke, "Three-dimensional alpha shapes," *ACM Trans. Graph.*, vol. 13, no. 1, pp. 43–72, 1994.

[8] M. F. Sanner, A. J. Olson, and J. C. Spehner, "Reduced surface: an efficient way to compute molecular surfaces," *Biopolymers*, vol. 38, no. 3, pp. 305–320, Mar 1996.

[9] M. Krone, M. Falk, S. Rehm, J. Pleiss, and T. Ertl, "Interactive exploration of protein cavities," in *Proceedings of the 13th Eurographics / IEEE - VGTC Conference on Visualization*, ser. EuroVis'11. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2011, pp. 673–682. [Online]. Available: http://dx.doi.org/10.1111/j.1467-8659.2011.01916.x [retrieved: 03, 2014]

[10] K. Ghosh, "A solution of polygon containment, spatial planning, and other related problems using minkowski operations," *Comput. Vision Graph. Image Process.*, vol. 49, no. 1, pp. 1–35, 1990.

[11] J. Janet, R. Luo, and M. Kay, "The essential visibility graph: an approach to global motion planning for autonomous mobile robots," *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, vol. 2, pp. 1958 –1963 vol.2, may. 1995.

[12] F. Lingelbach, "Path planning using probabilistic cell decomposition," *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, vol. 1, pp. 467 – 472 Vol.1, apr. 2004.

[13] K. Fujimura and H. Samet, "A hierarchical strategy for path planning among moving obstacles [mobile robot]," *Robotics and Automation, IEEE Transactions on*, vol. 5, no. 1, pp. 61 –69, feb. 1989.

[14] P. Fiorini and Z. Shillert, "Motion planning in dynamic environments using velocity obstacles," *International Journal of Robotics Research*, vol. 17, pp. 760–772, 1998.

[15] Y. Kwon, J. Cho, S. Kwon, and J. Joh, "Collision avoidance of moving obstacles for underwater robots," *Journal of Systemics, Cybernetics and Informatics*, vol. 4, no. 5, pp. 86–91, 2006.

[16] P. Beneš, P. Medek, O. Strnad, and J. Sochor, "Computation of dynamic channels in proteins," in *Proceedings of Biotechno*, U. o. S. Pei-Yuan Qian, KAUST and H. K. Technology, Eds. Venice/Mestre, Italy: Neuveden, 2011, pp. 78–83.

[17] P. Beneš, O. Strnad, and J. Sochor, "New path planning method for computation of constrained dynamic channels in proteins," *WSCG Full papers proceedings*, pp. 81–88, 2011.

[18] M. Petřek et al., "Caver: A new tool to explore routes from protein clefts, pockets and cavities," *BMC Bioinformatics*, vol. 7, p. 316, 2006.

[19] G. Voronoi, "Nouvelles applications des parametres continus a la theorie des formes quadratiques. duesieme memoire: Recherches sur les parallelodères primitifs," *J. Reine Angew. Math*, vol. 134, p. 198287, 1908.

[20] P. Medek, P. Beneš, and J. Sochor, "Computation of tunnels in protein molecules using delaunay triangulation," *Journal of WSCG*, vol. 15(1-3), pp. 107–114, 2007.

[21] C. Barber, C. Bradford, D. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Trans. Math. Softw.*, vol. 22, no. 4, pp. 469–483, 1996.

[22] M. Petřek, P. Košinová, J. Koča, and M. Otyepka, "Mole: A Voronoi diagram-based explorer of molecular channels, pores, and tunnels," *Structure*, vol. 15, pp. 1357–1363, 2007.

[23] E. Chovancová et al., "Caver 3.0: A tool for the analysis of transport pathways in dynamic protein structures," *PLoS Comput Biol*, vol. 8, no. 10, p. e1002708, 2012. [Online]. Available: http://dx.doi.org/10.1371/journal.pcbi.1002708 [retrieved: 03, 2014]

[24] N. Lindow, D. Baum, and H. Hege, "Voronoi-based extraction and visualization of molecular paths," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2025–2034, Dec. 2011. [Online]. Available: http://dx.doi.org/10.1109/TVCG.2011.259 [retrieved: 03, 2014]

[25] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems, Science, and Cybernetics*, vol. SSC-4, no. 2, pp. 100–107, 1968.

 ISBN: 978-1-61208-335-3