Global Journal of Engineering and Technology Advances

eISSN: 2582-5003

Cross Ref DOI: 10.30574/gjeta

Journal homepage: https://gjeta.com/

(Review Article)

Assessing ligand Traversability in protein tunnels: A motion planning approach*

Rohankumar Patel * and Ankur Patel

Analytical Research and Development Departments as Research Scientist III, Amneal Pharmaceuticals, Bridgewater, NJ, US.

Global Journal of Engineering and Technology Advances, 2023, 15(03), 217-227

Publication history: Received on 20 April 2023; revised on 02 June 2023; accepted on 05 June 2023

Article DOI: https://doi.org/10.30574/gjeta.2023.15.3.0096

Abstract

This paper presents a clever movement arranging framework to evaluate the traversability of ligand molecules within protein tunnels, utilizing the Haloalkane dehalogenase protein (PDB ID 1CQW) as a case study. The approach involves defining key metrics such as accessibility, throughput, and ligand scale to assess local and global traversability of tunnel segments. By implementing a planner that scales ligand radii, we systematically analyze the impact of various scaling factors on the success rates of ligand trajectories. Visualization techniques, including color mapping of tunnel properties and trajectory clustering, facilitate the interpretation of the results. Our experimental findings reveal significant insights into tunnel accessibility, highlighting that regions near bottlenecks exhibit critical limitations while alternative pathways may offer viable routes. The results underscore the importance of dynamic assessments in tunnel traversability, suggesting that while static models provide foundational insights, incorporating motion dynamics can enhance our understanding of ligand behavior within protein structures. This work aims to inform future studies in protein engineering and drug design by providing a comprehensive methodology for analyzing ligand transport through complex protein architectures.

Keywords: Ligand traversability; Protein tunnels; Haloalkane dehalogenase; Dynamic protein analysis; Ligand trajectory visualization; Protein engineering; Biochemical pathway analysis; Tunnel detection algorithms; Ligand-protein interaction

1. Introduction

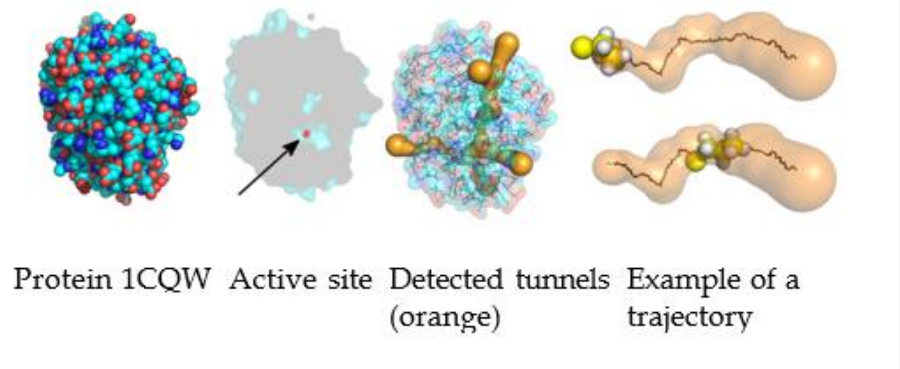


Figure 1 Tunnels in haloalkane dehalogenase with a possible trajectory of 1-chlorpropan ligand

* Corresponding author: Rohankumar Patel

Understanding the interactions between proteins and other small molecules is crucial in many research fields, including drug design and protein engineering. These interactions are highly influenced by the ability to transport the small molecule, called ligand, to the protein active site. The active site can be considered as a deeply buried inner cavity with the ability to interact with the incoming ligand. A transportation path must connect the protein's outer environment with the active site to transport the ligand to the active site. These paths, called tunnels, have to be wide enough, and the physico-chemical properties of the surrounding amino acids have to be compatible with the ligand (Fig. 1).

Traditionally, tunnels are detected using Voronoi diagrams [1, 2] assuming A spherical ligand (probe) they are represented as a sequence of spheres. Biochemists decide if a tunnel can be used to transport a given ligand mainly based on the tunnel length and bottleneck (i.e., the radius of the smallest sphere that forms the tunnel). This is used, for example, in protein engineering, where the task is to change selected properties of a protein, e.g., its stability under different outer conditions [3] or its activity of the protein towards other molecules [4].

Whereas tunnel computation is already a well-established research field, the simulation of ligand transportation through the detected tunnels is rather new. Ligands are typically non-spherical, so it is difficult to estimate their traversability through tunnels computed for a spherical probe. The decision based only on spherical tunnels requires previous expertise in the domain, and yet it may be imprecise. Compute the trajectories of the ligand considering its shape and possibly also its conformation changes to provide biochemists with a better insight into the behavior of non-spherical ligands.

We propose to analyze the traversability of the ligands in the tunnels using motion planning. The principle of Rapidly Exploring Random Tree (RRT) [5] planner is used and further modified. The main idea of the proposed planner is to generate random samples around a virtual sphere moving through the tunnel, which guides the search in the configuration space. This allows us to focus the sampling of the configuration space around the tunnel. The flexibility of the ligand is modeled using a predefined set of typical conformations. To enable ligand movements in narrow tunnels, a scaled down version of the ligand is used, similarly, e.g., to [6]. It also helps to keep potential solutions that do not fit the geometric restraints but can still be feasible because of their physicochemical properties.

The proposed method can be seen as an extension for the traditionally used tunnel detection tools. Our motivation is to help biochemists perform virtual screening, where they test the traversability of a ligand through a given tunnel. To predict the success of ligand traversability, the virtual screening performs hundreds of thousands of tests and checks whether the ligand passed through the tunnel. The proposed approach's advantage is that we can search the ligand path inside a specific tunnel, which substantially decreases the computational time and resources required for virtual screening.

2. Related Work

The assessment of protein structure hoping to uncover the sections has been maintained by different computational programming gadgets that take the math of the protein as data and explore the internal void space (e.g., CAVER 1.0 [7] or MOLE [8]). Early techniques for tunnel distinguishing proof utilized a discretized 3D cross-section, where each cell is viewed as involved or free, depending on the presence of particles of the protein. Entries can then be glanced through using standard outline search methods, like Dijkstra's computation. Also, the cross-section can be used to recognize other significant properties like pockets, pits, or channels [7, 9]. The undeniable hindrance of the framework-based strategies is the high memory interest and reliance on the matrix goal.

As of now, the most commonly used method for managing tunnel revelation relies upon standard Voronoi diagrams (VD) or Weighted Voronoi Outlines (WVD). The traditional VD is figured on centers tending to the focal points of all particles, ignoring the radii of atoms. To consider particles with different radii, the heaps of individuals still hang out by the van der Walls radii of particles in WVD. An elective game plan is to calculate a non weighted VD on an extensive point set, where a couple of circles deduced each particle with a little radius [1, 2]. VD-based procedures are less mentioned and faster than system based methodologies.

The tunnels detected by the approaches mentioned above are evaluated using basic characteristics like bottleneck, length, curvature, and a list of surrounding residues, which are later used to estimate the interaction possibilities. The biggest disadvantage of both grid based and VD-based methods is that the shape of the ligand is not taken into account during tunnel detection, and it is therefore not easy to estimate if (and how) a ligand might traverse the tunnels.

Compute a trajectory considering the shape of the ligand to determine if it can pass the tunnel. This can be formulated as a motion planning problem in a high dimensional configuration space. The configuration space C is formed by all possible configurations of the ligand in the tunnel, i.e., considering its rotation, translation, and possibly also other degrees of freedom responsible for the conformation changes. The dimension of the configuration space is given by the degrees of freedom (DOF) of the ligand, i.e., 6D for a rigid ligand and 6D + *n* for a flexible ligand with *n* DOFs. Sampling-based motion planning methods can be used to search this high-dimensional configuration space [10]. The idea of sampling-based motion planning is to randomly sample C and classify the samples as free or non-free using collision detection. The free samples are stored in a roadmap (a graph structure), in which a path can be searched using standard graph-search methods.

Sampling-based motion planning methods are suitable for computing the trajectories of the ligand as they can cope with many DOF robots (objects) of arbitrary shapes. The flexibility of ligands (or even proteins) can be modeled using a multi-link kinematic chain, where torsional angles can change [11]. This is useful, e.g., in the protein folding studies [12–15] or analysis of loop motions [16].

Quickly Investigating Irregular Tree (RRT) [5] is a solitary inquiry examining based movement arranging strategy that gradually constructs a design tree T established at the underlying setup q_{start}. In every emphasis of RRT, an irregular design $q_{rand} \in C$ is created, and its closest hub $q_{near} \in T$ in the tree is found. Another design q_{new} is built on the line associating q_{near} and q_{rand} somewhere far off ε from q_{near}. Assuming that q_{new} is without crash, it is added to the tree. The calculation ends assuming that the tree moves toward the objective design sufficiently close.

Motion planning for flexible ligands in protein tunnels brings two main issues: the ligand flexibility increases the dimension of the configuration space and the necessity to plan in protein tunnels leads to the narrow passage problem. A narrow passage is a region in the configuration space whose removal changes the connectivity of the free space [17]. Narrow passages have a smaller volume than other regions, and it is, therefore, difficult to sample them densely enough using the uniform distribution used in basic sampling-based planners. The presence of narrow passages in the configuration space, especially if they contain part of the solution, requires many iterations in order to put enough samples there and consequently increase the planning time. Typical tunnels have bottlenecks smaller than 1.0 Å, so they can already be considered narrow passages for ligands with more than two atoms. To cope with the narrow passages, they have to be sampled more densely. For the family of RRT planners, it is useful to change the distribution of random samples according to the growth of the tree [18–20]. The kinematic chain representation used to model the ligand flexibility can generate all possible conformations, but it also increases the dimension of the configuration space. However, not all conformations are feasible, and it is therefore necessary to verify the feasibility of a given conformation based on energy, which is time consuming. An alternative solution is to employ a library of known conformations. The conformation changes of protein amino acids and ligands are represented by so called rotamers stored in different rotamer libraries (e.g., Dunbrack library [21]). Possible rotamer conformations correspond to energetically and geometrically favorable positions.

A RRT-based strategy for registering exit pathways for a little adaptable particle was introduced in [22]. To adapt to numerous DOF ligands, the RRT-ML variant [23] was utilized. RRT-ML extends the tree fundamentally utilizing those DOFs that are fundamental for accomplishing movement of the ligand (i.e., turn what's more, interpretation), and it utilizes the other DOFs (i.e., that are answerable for conformity changes) assuming they prevent the development of the tree. The approach [22] may,, notwithstanding,, experience the ill effects of the limited section issue, as it expects to discover some leave pathway for a given ligand, which requires looking through the entire setup space of the ligand/protein complex. Despite the fact that there are normally different pathways from a given dynamic site, not every one of them are safe by a given ligand. Numerous cycles are expected to find an answer, which expands the computational time.

In this paper, we propose to analyze the traversability of each tunnel separately. This can bring several advantages for the biochemists. The tunnels in a protein being studied can be computed using standard tunnel detection tools for spherical probes and assessed according to their length, curvature, bottleneck, and biochemical properties (e.g., partial charge). These characteristics are already used in many research studies. Selected tunnels can further be analyzed with a given ligand to determine the traversability. According to the found trajectories, researchers may decide of the given ligand can reach the active site, which helps to organize subsequent MD simulations or other experiments.

The proposed work employs the RRT planning principle with three extensions: a) the configuration space is not sampled uniformly but uses a guided sampling along the tunnel centerline, b) the radii of the ligand atoms can be decreased, and c) ligand flexibility is modeled using a library of predefined conformations. The first two extensions are designed to cope with the narrow passage problem. By reducing the scale of the atomic radii, the ligand can move more without collisions, which widens the narrow passages and increases the probability of placing samples into them. The scaling-down technique has been used, e.g. for motion planning of deformable objects [24, 25], and they are also used in the most related MoMa-LigPath tool [6]. The number of dimensions of the configuration space does not depend on the ligand's

DOF by using the predefined set of conformations. Library of conformations is also used in different types of calculations, e.g. [26].

3. Traversability of Tunnels

3.1. Preliminaries

Proteins and ligands are addressed by the hard circle model, where the sweep of every circle (molecule) is given by its van der Waals span. The adaptability of a ligand is demonstrated utilizing the set L of its compliances. The compliances L are utilized from a library (e.g., [21]), or they can be arranged thinking about the possible energy. A protein tunnel is described by a sequence of collision-free spheres $T = ((c_1, r_1), \dots, (c_n, r_n))$, where n denotes the number of spheres, $c_i \in \mathbb{R}^3$ is their 3D position and $r_i > 0$ denotes the maximum collision free radius of a sphere centered at c_i. The tunnels can be found by tunnel detection tools like CAVER 3.0 [2].

To enable the motion of ligands in the narrow tunnels, the atomic radii of the ligand are scaled down by a factor s, $0 < s \leq 1$. A discrete set of scales is used, i.e., $s \in \{s_{min}, s_{min} + s_{\Delta}, \dots, s_{max}\}$, where s_{min} is the minimal allowed scale, $s_{max} = 1$ is the maximal allowed scale and s_{Δ} is the minimal difference between two scales.

A configuration of the ligand $q = (x, y, z, r_x, r_y, r_z, l, s)$ is described by the 3D position (x, y, z) of the reference point of the ligand (e.g., its geometric center), rotation around x, y, and z axes, index of the conformation $l \in L$ and the actual scale s. All possible configurations form the configuration space C. A configuration is collision-free if none of the ligand atoms scaled by s and placed at the position defined by q collides with the protein atoms.

3.2. Computing Initial Configuration

The analysis of tunnel traversability is based on the computation of multiple trajectories of the ligand inside the tunnel, which requires to generate a set Q_{init} of collision-free initial configurations. In this paper, we assume that the ligand has to travel from the beginning to the end of the tunnel. As the tunnels are computed for a spherical probe, the ligand may not fit exactly to the first sphere of the tunnel. The initial configurations have to be searched around the first sphere of the tunnel. To find a new initial configuration, a random sample q is generated around c_i in the distance R_{init} (translation and rotation parts of q are generated randomly, the scale is set to s_{min} and the conformation index is set randomly). If the sample q is collision-free, it can be considered as a new starting configuration, so q is added to Q_{init}. Similarly, a single goal configuration q_{goal} is found around the end of the tunnel. For each starting configuration $q_{start} \in Q_{init}$, K trajectories are created. Each trajectory is computed using a modified RRT, which is introduced in the following section.

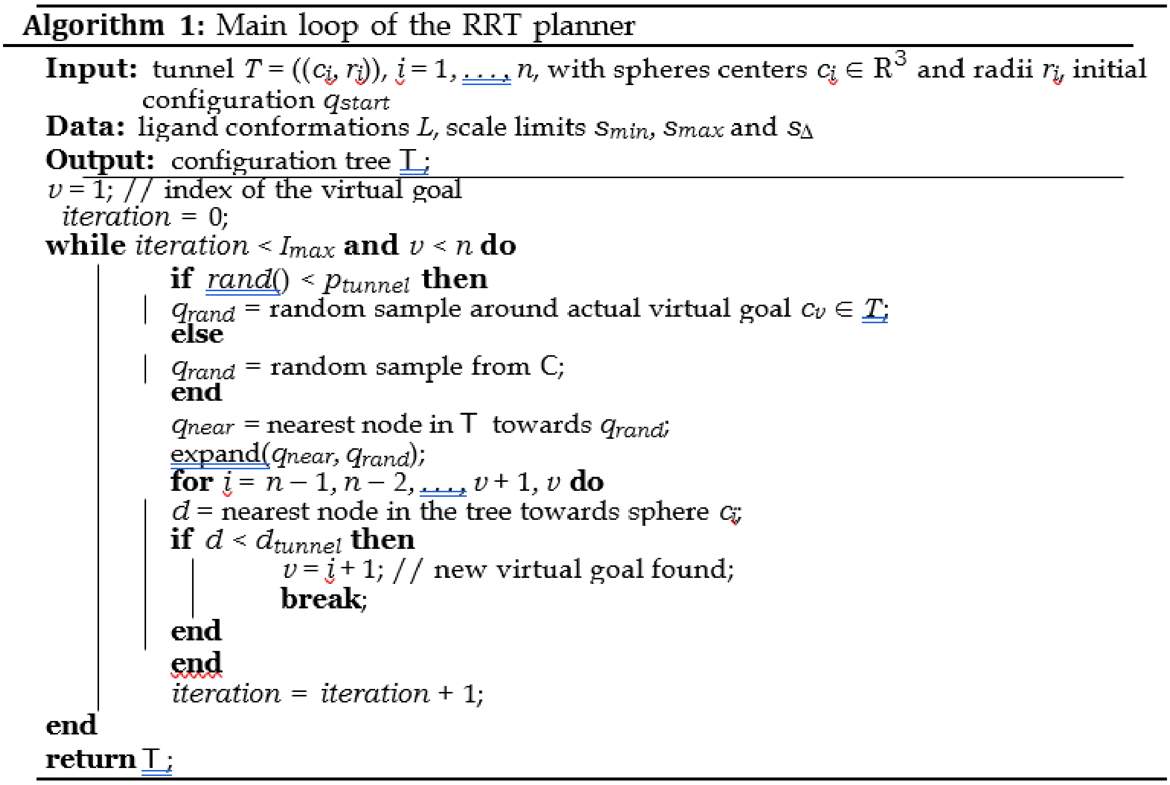
3.3. Computing Single Trajectory of the Ligand

The task of the trajectory computation is to find a trajectory for a ligand in the given tunnel. Here, the original RRT is extended to cope with the specific requirements needed for the traversability of ligands. First, the trajectory has to be found around the given tunnel, but a deflection from the tunnel centerline is allowed. Due to this requirement, the sampling process of RRT has to be adapted in order to follow the tunnel and to prevent the construction of trajectories in the rest of the protein.

The main loop of the proposed method is described in Alg. 1. In each iteration, a random sample q_{rand} is generated, and its nearest node $q_{near} \in$ T in the tree is found. The nearest-neighbor search between q_{rand} and the tree is performed using the weighted 6D Euclidean metric considering both 3D rotation and 3D translation.

To guide the growth of the tree through the tunnel, a moving virtual goal is used [18]. The virtual goal v, $1 \leq v \leq n$, is the index of a sphere of the tunnel. The random samples q_{rand} are generated around the sphere $c_v \in T$ with the probability p_{tunnel} and from the whole C otherwise. After the tree reaches the sphere c_v, i.e., the distance of the tree to c_v is less than a predefined threshold d_{tunnel}, the virtual goal is moved to the successor of the last sphere in the tunnel that is reached by the tree (lines 1–1 in Alg. 1). Setting the virtual goal to this successor allows the tree to avoid such parts of the tunnels that are not traversable or reachable by the ligand. This is necessary in dense protein structures, where it is not always possible to follow the tunnel exactly. The algorithm terminates after a predefined number of planning trials I_{max} or if the tree reaches the last sphere in the tunnel, i.e., when $v = n$.

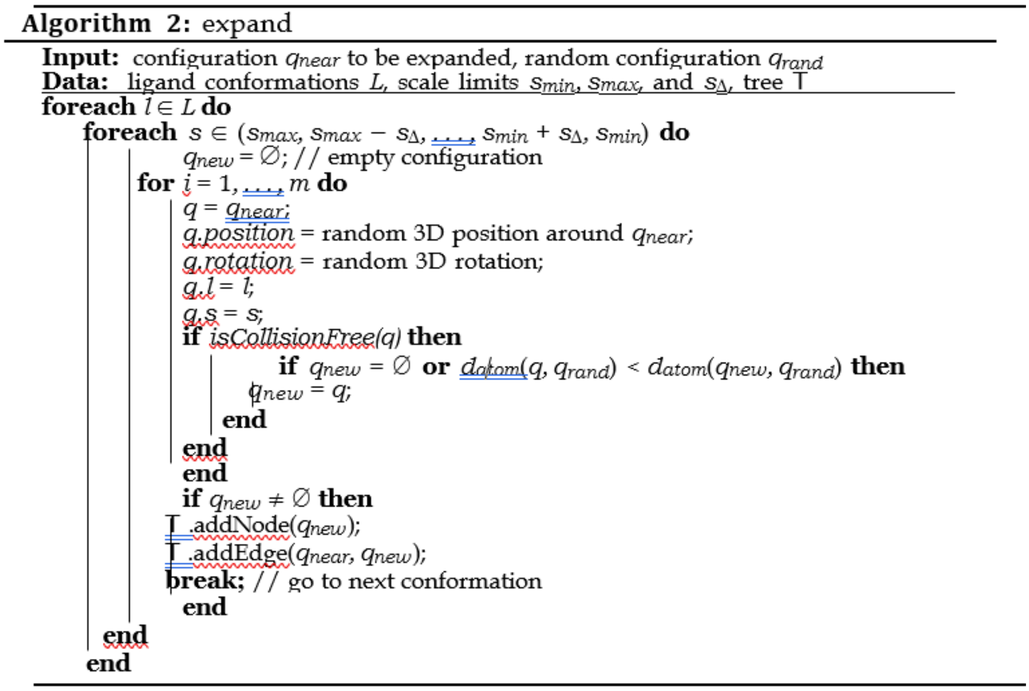
To generate the samples q_{rand} around the virtual goal v, the translation part (x, y, z) of q_{rand} is generated from $N(c_v, \Sigma)$, where Σ is the diagonal matrix with diagonal entries equal to the parameter R_{tunnel}, and the rotational part of q_{rand} is generated using techniques described in [27]. The other two parameters (ligand index l and scale s) of q_{rand} can be left zero, as these are not used in the employed metric for the nearest-neighbor search. The parameter R_{tunnel} influences the distribution of samples around the tunnel centerline. By setting R_{tunnel} to a small value, the planner attempts to find the trajectories inside the tunnel, while higher values of R_{tunnel} cause exploration of paths around the tunnel. We propose to set this parameter to the average tunnel width.



```
Algorithm 1: Main loop of the RRT planner
Input: tunnel $T = ((c_i, r_i))$, $i = 1, \dots, n$, with spheres centers $c_i \in \mathbb{R}^3$ and radii $r_i$, initial
       configuration $q_{start}$
Data: ligand conformations $L$, scale limits $s_{min}$, $s_{max}$ and $s_\Delta$
Output: configuration tree $\mathcal{T}$;
$v = 1$; // index of the virtual goal
 $iteration = 0$;
while $iteration < I_{max}$ and $v < n$ do
        if $rand() < p_{tunnel}$ then
            $q_{rand}$ = random sample around actual virtual goal $c_v \in T$;
        else
            $q_{rand}$ = random sample from C;
        end
        $q_{near}$ = nearest node in $\mathcal{T}$ towards $q_{rand}$;
        expand($q_{near}$, $q_{rand}$);
        for $i = n - 1, n - 2, \dots, v + 1, v$ do
            $d$ = nearest node in the tree towards sphere $c_i$;
            if $d < d_{tunnel}$ then
                    $v = i + 1$; // new virtual goal found;
                    break;
            end
        end
        $iteration = iteration + 1$;
end
return $\mathcal{T}$;
```

The core of the proposed planner is the expansion procedure (Alg. 2), which generates new collision-free nodes around $q_{near} \in \mathcal{T}$. For each ligand conformation $l \in L$, the expansion procedure attempts to find a new collision-free configuration around q_{near} with a maximal scale. First, the maximal scale s_{max} is tested, and m random samples are generated around q_{rand} and tested for collision. The nearest collision-free sample towards q_{rand} is selected and added to the tree. If none of the tested samples is collision-free, the scale is reduced to $s_{max} - s_\Delta$, and the search continues until a collision-free sample is found or until the minimal reduced-scale s_{min} is reached. The random samples are generated similarly as in the case of q_{rand} samples; only their translation part is generated around q_{near}.

The sampling-based methods are sensitive to the employed metrics, especially if the objects are not symmetrical, which is the case of the non-spherical ligands. To consider the actual shape of the ligand (which is different in each conformation) and to support finding such configurations that maximally approach q_{rand}, the distance between newly generated configurations and q_{rand} is measured as the smallest 3D distance between an atom of the ligand placed at q and the 3D position of q_{rand} ($d_{atom}(q, q_{rand})$ on line 2 in Alg. 2). By computing the distance between the nearest atoms, the shape of the ligand is actually considered. This metric supports the retraction of the ligand towards q_{rand}.



```
Algorithm 2: expand
Input: configuration $q_{near}$ to be expanded, random configuration $q_{rand}$
Data: ligand conformations $L$, scale limits $s_{min}$, $s_{max}$, and $s_{\Delta}$, tree T
foreach $l \in L$ do
    foreach $s \in (s_{max}, s_{max} - s_{\Delta}, \dots, s_{min} + s_{\Delta}, s_{min})$ do
        $q_{new} = \emptyset$; // empty configuration
        for $i = 1, \dots, m$ do
            $q = q_{near}$;
            $q.position$ = random 3D position around $q_{near}$;
            $q.rotation$ = random 3D rotation;
            $q.l = l$;
            $q.s = s$;
            if $isCollisionFree(q)$ then
                if $q_{new} = \emptyset$ or $d_{atom}(q, q_{rand}) < d_{atom}(q_{new}, q_{rand})$ then
                    $q_{new} = q$;
                end
            end
        end
        if $q_{new} \neq \emptyset$ then
            T .addNode($q_{new}$);
            T .addEdge($q_{near}$, $q_{new}$);
            break; // go to next conformation
        end
    end
end
```

The result of each planning trial is the tree T of collision-free configurations in which a path between q_{start} (root of the tree) and q' is found, where q' is the nearest node in the tree towards q_{goal} (using 3D Euclidean metric). The path $P = (q_i)$, $q_i \in$ C is represented as a sequence of collision-free configurations. The path is found in the tree even if the tree does not approach q_{goal} close enough. Considering these non-feasible solutions is also necessary to evaluate difficult areas of the tunnels, e.g., bottlenecks. The utilization of all computed paths for the evaluation of tunnel difficulty is described in the next section.

3.4. Traversability Characteristics

For each initial configuration $q_{start} \in Q_{init}$, K trajectories are computed, which results in the set of $K|Q_{init}|$ trajectories. All these trajectories are used to compute the following properties of the tunnel T. A trajectory P reaches the tunnel sphere $c_i \in T$ if the 3D Euclidean distance of the nearest configuration $q \in P$ towards c_i is less than r_i (radius of the sphere c_i). Let $N(i)$ denote the number of trajectories that reached i–th sphere of the tunnel, $i = 1, \dots, n$. Three basic characteristics of the tunnel are computed from the trajectories: accessibility, throughput, and the scaling factor.

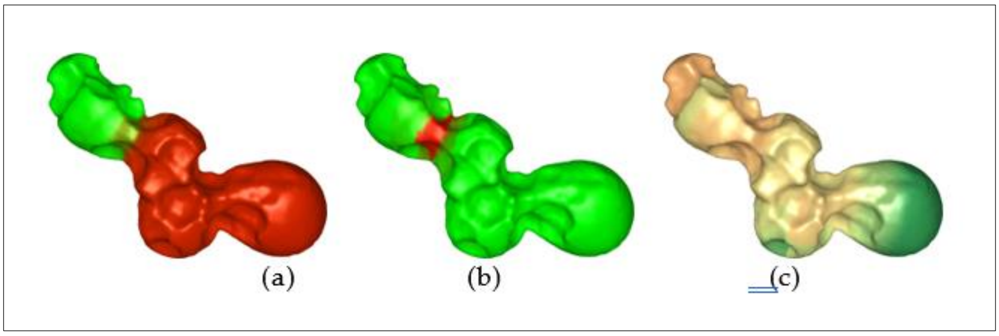
The accessibility $A(i) = N(i)/N$ of the sphere i is the probability of reaching the sphere i, where $N = K|Q_{init}|$ is the total number of trajectories. The accessibility shows how probable it is to pass the tunnel up to the sphere i. Obviously, the most important is $A(n)$ of the last sphere of the tunnel, which can be considered as the overall difficulty of the tunnel. The ligand passage may, however be strongly affected by the first bottleneck, so the parts of the tunnel located behind the bottleneck has low accessibility.

The throughput $T(i)$ is the ratio of trajectories that passed sphere i (i.e., visited sphere $i + 1$) and reached the sphere i, i.e., $T(i) = N(i + 1)/N(i)$. The throughput is not computed for the last sphere ($i = N$). The throughput shows the local accessibility of tunnel parts, and it can be used to detect places where most of the trajectories end. The proposed planner is allowed to scale down the radii of ligand spheres up to the permitted scale s_{min}. It can be expected that narrower parts of the tunnel are more often passed with a more scaled-down ligand than the wider parts. Ligand scale $L(i)$ at the tunnel sphere i is the average scale of ligand that reaches the sphere i.

3.5. Visualization of Results

The above-defined characteristics can be shown as a graph or, better yet, presented visually by mapping them using colors to the tunnel spheres. Alternatively, the surface representation can be used to visualize the tunnel. In this case, a 3D point on the tunnel surface is colored according to the property value in its nearest tunnel part i, i.e., such part i whose center c_i is the closest to the point among all tunnel parts.

The examples of the color mapping are shown in Fig. 2. The accessibility (Fig. 2a) shows that more than half of the tunnel is not accessible (red part of the tunnel). The throughput shows (Fig. 2b) that the only difficulty is the part around the bottleneck (red color in Fig. 2b), and the second half of the tunnel is also traversable.



Besides the color mapping, it is also necessary to show the computed trajectories. Simple visualization of all trajectories could, however, be too slow for an interactive work. Therefore, the trajectories are first clustered, and then only the clusters are visualized. Due to the different lengths of the trajectories, they are first converted to a normalized form. Let c_{start} represent the average starting position of all trajectories and let d_{max} be the 3D Euclidean distance of the most distant configuration from c_{start} among all trajectories. A set of M spheres centered at c_{start} are created with radii , $ri' = i\frac{dmax}{\mathrm{M}}$ where $i = 0,\ldots,M-1$. The trajectory $P = (q_1,\ldots,q_n)$ of length n

is represented by the normalized vector $v = (x_1,\ldots,x_M)$ of length M, where x_i is the 3D position of the nearest configuration $q \in P$ to the surface of the i-th sphere with the radius r_i'. The distance between two normalized trajectories v_i and v_j is defined as

$\mathrm{d}(\mathrm{v}_\mathrm{i},\mathrm{v}_\mathrm{j})= \frac{1}{M}\sum_{1\le k\le N} |\ \mathrm{x^i}_\mathrm{k}\ \mathrm{x^j}_\mathrm{k}|$. This distance is used in the UPGMA clustering technique [28]. The trajectories can be visualized using a representative of each cluster. The number of trajectories in each cluster is represented by the width of the polyline.

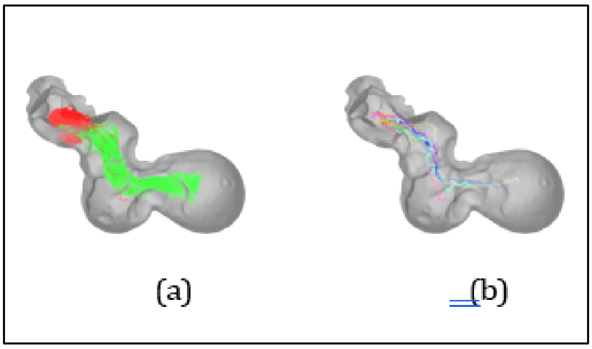


Figure 2 Visualization of the trajectories. The tunnel begins in the top left corner. (a) All trajectories (~ 5000) colored according to whether they reached the end of the tunnel (green) or not (red). (b) Visualization using clusters of trajectories

4. Experimental Verification

The tunnels in the Haloalkane dehalogenase protein (PDB ID 1CQW) have been analyzed using the proposed approach. Three tunnels were detected using CAVER 3.0 [2] for the spherical probe of radius 0.9 Å(Fig. 3a). The traversability was evaluated for 1-chlorpropan (denoted as L_1) with 11 atoms and and 1-chlorbutan (denoted as L_2) with 14 atoms. L_1 was represented by 12 conformations, and L_2 by a set of 114 conformations. Examples of three conformations of L_1 and L_2 are depicted in Fig. 3b. Further information about the experiments can be found at http://mrs.felk.cvut.cz/ isrr2017.

Both tested ligands have more than ten atoms and therefore, they cannot fit into the tunnel with 0.9 Å , so the radii of ligands were scaled down. Four different scaling down factors were used: $s_{min} = \{0.3, 0.4, 0.5, 0.6\}$. No trajectories were found for $s_{min} > 0.6$. For each scale, 50 collision-free initial configurations were found in the $R_{init} = 2$Å radius around the first sphere of the tunnel. For each ligand, each minimal scale s_{min}, and each initial configuration, 100 trajectories were computed using the proposed planner. The parameters of the planner were: I_{max} = 10,000, m = 50, p_{tunnel} = 0.9, d_{tunnel} = 1.5Å, R_{tunnel} = 2 Å. For each initial configuration, the success rate is computed as the ratio of trajectories that reached the end of the tunnel (to the distance R_{tunnel} = 2 Å or less) over 100 trials.

he average runtimes, success rates, and sizes of the built trees for the first tunnel are shown in Table 1. The highest average success rate was achieved for the most scaled-down ligands (s_{min} = 0.3), and it decreases with the increasing s_{min}. The difference between the minimal and maximal success rates indicates how the ligand's initial configurations influence its ability to pass the tunnels. For example, the most difficult initial configuration of L_1 and s_{min} = 0.3 lead to success rate 78 % (the easiest initial configuration lead to success rate 100 %), but for the scale s_{min} = 0.4, the worst success rate is only 42 % and the best 96 %. The traversability of larger ligands (s_{min} = 0.4 vs. s_{min} = 0.3) is therefore more sensitive to the initial configuration. This shows the importance of testing the traversability from multiple initial configurations. The runtime is significantly higher for the L_2 ligand, which is caused by the larger number of conformations that need to be examined in each expansion. The average runtimes, which are in the order of minutes, are negligible in the comparison of runtimes of MD simulations.

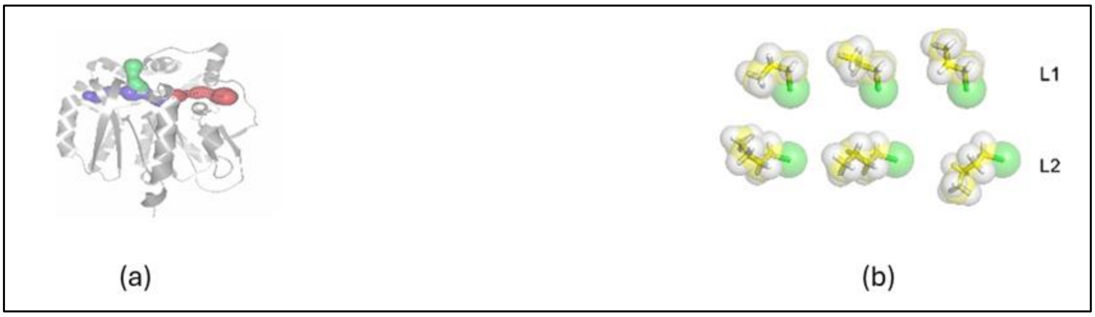


Figure 3 (a) The protein 1CQW is visualized using the cartoon representation (gray) with three tunnels (red, green, and blue) were detected by CAVER 3.0. The first tunnel is depicted in red. (b) Examples of three conformations of L_1 (top) and L_2 (bottom)

Table 1 Runtime and success ratio for ligand L_1 (left) and L_2 (right)

Scale	Runtime [s]	Success rate	Tree size
s_{min}	avg. (std.)	min/max/avg	
0.3	59.25 (40.21)	78/100/91.4	41k
0.4	98.58 (68.55)	42/96/78.1	47k
0.5	162.43 (85.14)	2/60/18.3	47k
0.6	160.71 (64.71)	0/0/0.0	38k

Scale	Runtime [s]	Success rate	Tree size
smin	avg. (std.)	min/max/avg	
0.3	253.42 (100.73)	100/100/100	45k
0.4	641.13 (460.82)	88/100/95.4	49k
0.5	1852.27 (736.44)	0/70/24.3	44k
0.6	1995.02 (478.62)	0/0/0.0	35k

Trajectories for L_1 in all detected tunnels were classified as successful if they reached the last sphere of the tunnel to distance $d_{tunnel} = 2\ \mathring{A}$, and they were considered unsuccessful otherwise. The throughput computed from the trajectories shows that the tunnels are difficult not only around bottlenecks but also in other places.

The comparison of the classic bottleneck (i.e., measured by the radius of the spherical probe) and throughput is depicted in Fig. 4a. The trajectories for $s_{min} = 0.4$ are depicted in Fig. 4b and for $s_{min} = 0.5$ in Fig. 4c. The successful trajectories for $s_{min} = 0.4$ reveal that the end of tunnel No. 3 can be approached by two different pathways (one in the tunnel and another one outside the tunnel). The detail is depicted in Fig. 5. Despite the low bottleneck of this tunnel, the ligand may reach its end using the alternative pathway.

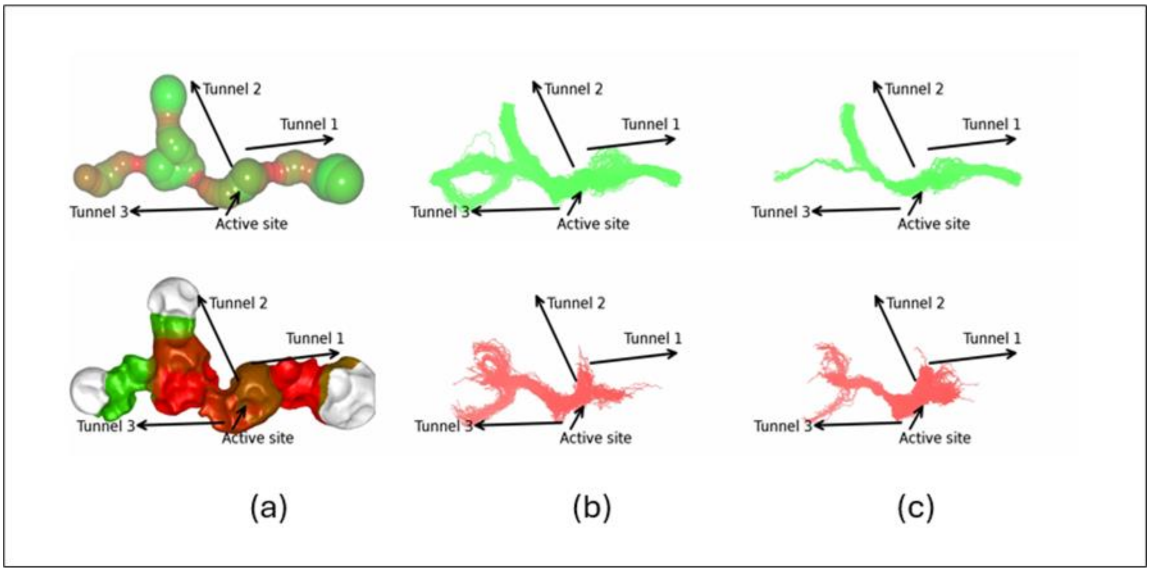


Figure 4 (a) Classic bottleneck for spherical probe (top) and visualization of throughput (bottom) for the ligand L_1. (b) Successful (green, top) trajectories that reached end points of the tunnels and unsuccessful ones (red, bottom) for $s_{min} = 0.4$. (c) Successful and unsuccessful trajectories for $s_{min} = 0.5$

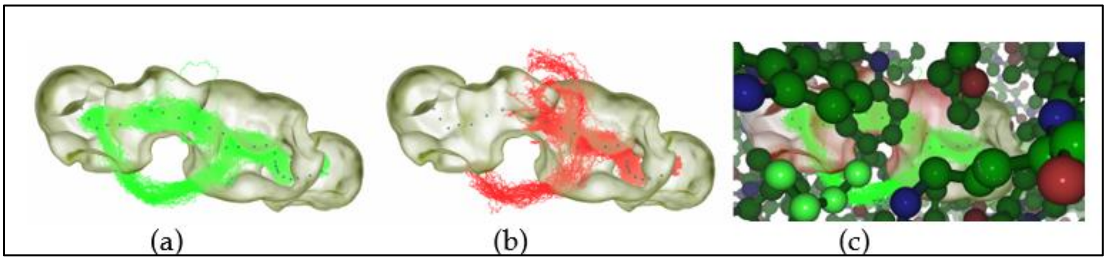


Figure 5 Detailed view of alternative pathway around the third tunnel in 1CQW. The successful trajectories are in green (a) and the unsuccessful in red (b). (c) Shows visualization of protein atoms around the tunnel

5. Discussion

The computed trajectories cannot be considered ‘real’; they should rather be considered a hint for the biochemists. One of the reasons is that the trajectories are computed inside a static protein and static tunnels. Real proteins are dynamic structures, and consequently, the tunnels are also dynamic: they move, merge, or even disappear due to the motions of protein atoms. On the contrary, usually static tunnels are used in protein engineering. Despite such a simplification, researchers often used information about the static tunnels to estimate the possibility of chemical reactions at the active sites.

The utilization of scaled-down ligands causes the second limitation. The protein atoms fill the internal void space; therefore, the tunnels identified inside proteins without ligands tend to be very narrow. Scaling down the ligand atoms is necessary to enable at least some motion of the ligands inside such narrow tunnels. It is used also in other related tool, e.g. [6].

Due to these reasons, the computed trajectories can be considered either as too optimistic (e.g., because they are computed on a wide tunnel that can be, in fact, closed due to molecular dynamics), or too pessimistic (e.g., the trajectory is not found because of a narrow passage around the tunnel, which can be opened due to the molecular dynamics). Despite these limitations, testing the tunnel traversability using motion planning technique can provide chemists with more information then the simple bottleneck radius, which is used nowadays. For example, possible detours from the tunnel can be identified, which indicates that even a tunnel with a small bottleneck can be traversed (Fig. 5).

The proposed planner is very simple, and it can be improved in many ways: by using a less aggressive scaling-down technique (in order to obtain better trajectories) or by using only a subset of conformations and a better expansion (in order to speed it up). However, having a better motion planner (in terms of speed or even success rate) does not automatically mean that the resulting trajectories will help the biochemists more. The presented work shows possible advantages of the analysis based on ligand trajectories. By comparing the planned trajectories with trajectories observed in MD simulations, thresholds for the success rate or minimal tunnel throughput can be set, which will help biochemists make decisions.

6. Conclusion

The protein tunnels transport pathways for ligands to the active sites. In this paper, we propose to analyze the traversability in the tunnels using motion planning that can consider both the shape of the protein and ligand. We proposed modifications to the RRT method to compute trajectories for a ligand represented by a library of common conformations. The corresponding configuration space is sampled using guided sampling, where a virtual sphere moves along the tunnel and attracts growth of the tree towards it. To enable trajectory computation in narrow tunnels, the atomic radii of the ligand are scaled down. Computed trajectories allow us to evaluate the traversability of the tunnels using accessibility and throughput and by visualizing typical pathways found by the planner. These properties can help us understand the importance of tunnels better.

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] Yaffe, E., Fishelovitch, D., Wolfson, H. J., Halperin, D. & Nussinov, R. MolAxis: efficient and accurate identification of channels in macromolecules. Proteins: Struct. Funct. Bioinf. 73, 72–86 (2008). URL http://dx.doi.org/10.1002/prot.22052.

[2] Chovancov´a, E. et al. CAVER 3.0: A tool for the analysis of transport pathways in dynamic protein structures. Pathways in Dynamic Protein Structures, PLoS Comput. Biol. 8(10) (2012).

[3] Koudel´akov´a, T. et al. Engineering enzyme stability and resistance to an organic cosolvent by modification of residues in the access tunnel. Angew. Chem. Int. Ed. 52, 1959–1963 (2013). URL http://dx.doi.org/10.1002/anie.201206708.

[4] Pavlov´a, M. et al. Redesigning dehalogenase access tunnels as a strategy for degrading an anthropogenic substrate. Nat. Chem. Biol. 5, 727–733 (2009).

[5] LaValle, S. M. Rapidly-exploring random trees: A new tool for path planning (1998). Technical report 98–11.

[6] Cort´es, J. et al. A path planning approach for computing large-amplitude motions of flexible molecules. Bioinformatics 21, i116–i125 (2005).

[7] Petˇrek, M. et al. CAVER: a new tool to explore routes from protein clefts, pockets and cavities. BMC Bioinformatics 7 (2006). URL http://dx.doi.org/10.1186/ 1471-2105-7-316.

[8] Petˇrek, M., Koˇsinov´a, P., Koˇca, J. & Otyepka, M. MOLE: A Voronoi diagram-based explorer of molecular channels, pores, and tunnels. Structure 15, 1357–1363 (2007). URL http://www.sciencedirect.com/science/article/pii/S0969212607003759.

[9] Sehnal, D. et al. Mole 2.0: advanced approach for analysis of biomacromolecular channels. J. Cheminf. 5, 39 (2013).

[10] LaValle, S. M. Planning Algorithms (Cambridge University Press, Cambridge, U.K., 2006).

[11] Song, G. & Amato, N. M. Using motion planning to study protein folding pathways, 287–296 (2001).

[12] Al-Bluwi, I., Sim´eon, T. & Cort´es, J. Motion planning algorithms for molecular simulations: A survey. Comput. Sci. Rev. 6, 125–143 (2012).

[13] Gipson, B., Hsu, D., Kavraki, L. E. & Latombe, J.-C. Computational models of protein kinematics and dynamics: Beyond simulation. Annu. Rev. Anal. Chem. (Palo Alto, Calif.) 5, 273 (2012).

[14] Amato, N. M. & Song, G. Using motion planning to study protein folding pathways. J. Comput. Biol. 9, 149–168 (2002).

[15] Raveh, B., Enosh, A., Schueler-Furman, O. & Halperin, D. Rapid sampling of molecular motions with prior information constraints. PLoS Comput. Biol. 5, e1000295 (2009).

[16] Cort´es, J., Sim´eon, T., Remaud-Sim´eon, M. & Tran, V. Geometric algorithms for the conformational analysis of long protein loops. J. Comput. Chem. 25, 956–967 (2004).

[17] Kurniawati, H. & Hsu, D. Workspace importance sampling for Probabilistic Roadmap Planning, Vol. 2, 1618–1623 (2004).

[18] Von´asek, V., Faigl, J., Krajn´ık, T. & Pˇreuˇcil, L. in RRT-path — a guided rapidly exploring random tree 307–316 (Springer, 2009).

[19] Von´asek, V. A guided approach to sampling-based motion planning. Ph.D. the- sis, Czech Technical University in Prague (2016). URL https://dspace.cvut.cz/ handle/10467/62503.

[20] Denny, J., Sandstr¨om, R., Bregger, A. & Amato, N. M. Dynamic region-biased rapidly-exploring random trees (2016).

[21] Dunbrack, R. L. Rotamer libraries in the 21st century. Curr. Opin. Struct. Biol.12, 431–440 (2002).

[22] Cort´es, J., Le, D. T., Iehl, R. & Sim´eon, T. Simulating ligand-induced conforma- tional changes in proteins using a mechanical disassembly method. Phys. Chem. Chem. Phys. 12, 8268–8276 (2010).

[23] Cort´es, J., Jaillet, L. & Sim´eon, T. Molecular disassembly with RRT-like algorithms, 3301–3306 (2007).

[24] Gayle, R., Segars, P., Lin, M. C. & Manocha, D. Path planning for deformable robots in complex environments., Vol. 2005, 225–232 (2005).

[25] Alterovitz, R. & Goldberg, K. Motion Planning in Medicine: Optimization and Simulation Algorithms for Image-Guided Procedures Vol. 50 (Springer Science & Business Media, 2008).

[26] Kellogg, E. H., Leaver-Fay, A. & Baker, D. Role of conformational sampling in computing mutation-induced changes in protein structure and stability. Proteins 79, 830–838 (2011).

[27] Kuffner, J. J. Effective sampling and distance metrics for 3D rigid body path planning, 3993–3998 (2004).

[28] Sokal, R. R. & Michener, C. D. A statistical method for evaluating systematic relationships. University of Kansas Science Bulletin 38, 1409–1438 (1958).