EET-Based Motion Planning Applied to Protein-Ligand Interactions

Bachelor's Thesis submitted

by

Friederike Fischer
(346827)

to

Robotics and Biology Laboratory
Institute of Computer Engineering and Microelectronics
Faculty IV - Electrical Engineering and Computer Science
Technische Universität Berlin

Examiners:
Professor Dr. Oliver Brock
Professor Dr. Juri Rappsilber

Advisor:
Arne Sieverling

Berlin, April 06, 2017

Eidesstattliche Erklärung

Die selbstständige und eigenhändige Anfertigung versichere ich an Eides Statt.

Friederike Fischer Berlin, den 06. April 2017

Contents

List of Abbreviations **3**

List of Figures **4**

List of Tables **5**

List of Algorithms **6**

1 Kurze Zusammenfassung **7**

2 Abstract **8**

3 Introduction **9**

4 Related Work **13**
4.1 Protein modeling as kinematic chains 13
4.2 Sampling-based motion planning algorithms 13
4.2.1 PRM 13
4.2.2 RRT 14
4.2.3 EET 15
4.3 Motion planning for proteins 15
4.3.1 PRM for molecular simulations 15
4.3.2 RRT for molecular simulations 16

5 Problem Description **17**
5.1 Motion planning for proteins 17
5.2 Three kinds of degrees of freedom 17
5.3 Virtual end-effector 19

6 Method **21**
6.1 EET adaptation 21
6.1.1 Virtual end-effectors 21
6.1.2 Null space sampling 22
6.2 Ligand and protein model 23

7 Experimental Results **25**
7.1 Test setup 25
7.2 Results 27
7.3 Analysis 28
7.4 Virtual end-effector experiment 30

8 Conclusion and Future Work **32**

References **33**

List of Abbreviations

APH	4-aminophthalhydrazide
DoF	Degree of Freedom
EET	Exploration and Exploitation Tree
HIV	Human immunodeficiency virus
HIVP	HIV protease
ML-RRT	Manhattan-like RRT
PCR	Probabilistic Conformational Roadmap
PDB	Protein Data Bank
PRM	Probabilistic Roadmap
RCSB	Research Collaboratory for Structural Bioinformatics
RL	Robotics Library
RRT	Rapidly-exploring Random Tree
RRTCon	RRT-Connect
SRS	Stochastic Roadmap Simulation
TGT	tRNA-guanine transglycosylase

List of Figures

1 Protein-ligand interaction 1ENU 11
2 Protein-ligand interaction 2PSV 11
3 Protein-ligand interaction 3O96 12
4 Schematic view of the PRM algorithm 14
5 Schematic view of the RRT algorithm 14
6 Proteins and robots 18
7 Workspace tunnels 19
8 Ligands with virtual end-effectors 20
9 Unbinding of the protein-ligand interaction 1ENU 26
10 Unbinding of the protein-ligand interaction 2PSV 27
11 Unbinding of the protein-ligand interaction 3O96 28
12 2PSV problem with RRTCon 29

List of Tables

1 Symbols and variables used in the algorithms 22
2 Numerical results using 70% of the van der Waals radii 29
3 Numerical results using 80% of the van der Waals radii 30
4 Numerical results using different virtual end-effectors 31

List of Algorithms

1 CONNECT algorithm 23
2 EXTEND algorithm 23
3 NULL SPACE SAMPLING algorithm 23

1 Kurze Zusammenfassung

Bei der Wechselwirkung zwischen Proteinen und Liganden dockt der Ligand an die aktive Stelle des Proteins und verändert dadurch die Proteinfunktion. Für die Entwicklung von Medikamenten ist es entscheidend zu wissen, welche Liganden die aktive Stelle eines Proteins erreichen können. Diese Liganden sind potentielle Kandidaten, die das gegebene Protein hemmen oder aktivieren, und dadurch die Krankheit heilen. Wir zeigen, wie man Sampling-basierte Bewegungsplanung anwendet, um das Problem zu lösen, einen Liganden von einem Protein zu trennen. Das Problem ist aufwändig und dessen Simulierung und Lösung erfordert einen hohen Rechenaufwand, weil der Suchraum eine hohe Dimensionalität aufweist. Wir nutzen die Analyse des Problems, um die Rechenzeit zu reduzieren. Der Tunnel, der die Oberfläche des Proteins und die aktive Stelle verbindet, ist der Bereich, von dem Samples verwendet werden sollen, um das Problem so effizient wie möglich zu lösen. Außerdem unterscheiden wir zwischen verschiedenen Arten von Freiheitsgraden des Problems und kommen zu dem Schluss, dass die Bewegung für jede dieser Gruppen individuell berechnet werden sollte. Die drei Gruppen bestehen aus den Freiheitsgraden des Liganden in Bezug auf das Protein, die interne Beweglichkeit des Liganden und die beweglichen Seitenketten des Proteins. Eine erweiterte Version des Exploration/Exploitation Tree Planers setzt diese Überlegungen um. Dieser Ansatz löst verschiedene medizinisch relevante Wechselwirkungen zwischen Proteinen und Liganden effizient.

2 Abstract

In protein-ligand interactions, the ligand docks at the active site of the protein to alter the proteins function. For drug design it is essential to know which ligands can reach the active site of a protein. These ligands are potential candidates that inhibit or activate the given protein, and thereby cure a disease. We will show how to use sampling-based motion planning to solve protein-ligand disassembly problems. The problem is computationally expensive, because of the high dimensionality of the search space. We exploit the knowledge of the problem to reduce the computing time. The tunnel connecting the surface of the protein with the active site is the area to be sampled in order to solve the problem as efficient as possible. We differentiate between different sets of degrees of freedom of the problem and come to the conclusion that the motion for these sets should be computed individually. The three sets consist of the degrees of freedom of the ligand with respect to the protein, the ligands internal mobility, and the mobile side chains of the protein. An extended version of the Exploration/Exploitation Tree planner implements these considerations. This approach solves several medically relevant protein-ligand interactions efficiently.

3 Introduction

In protein-ligand interactions, the ligand docks at the active site of the protein and thereby alters the function of the protein. This can be the activation or inhibition of some process, such as a motion of the protein. To do so, the active site of the protein must be accessible for the ligand, a small molecule. In the problems addressed in this work, the active site can only be reached through a tunnel.

In structural bioinformatics, simulating the dynamics of protein-ligand interactions can help to analyze and understand molecular interactions. In practice, molecular simulations can be helpful for drug design against the human immunodeficiency virus (HIV) [13] or cancer [33]. They could accelerate the designing process by narrowing down the potential molecules to those candidates that, from a mechanical point of view, can reach the active site of the biomolecule. Figures 1, 2, and 3 illustrate the three protein-ligand interactions addressed in this work. tRNA-guanine transglycosylase (TGT)[1] is analyzed to develop a drug against Shigellosis [13]. HIV protease[2] is one leverage point to treat HIV [10]. The analyzes of the human AKT1[3] is promising for the treatment of cancer [33]. Protein-ligand interactions are referred to by the code given by the RCSB Protein Data Bank (PDB)[4] [7].

The problem is to determine whether a valid path for the ligand to the active site exists, and if so, to simulate the unbinding process and obtain the valid path as a result. This biochemical problem presents many degrees of freedom (DoFs). When taking into account the binding energies, the simulation of this procedure is called virtual screening [14], [29], [31]. Virtual screening is promising for drug discovery [17], but is computationally expensive. Thus, a trade-off between biological and chemical accuracy, and computational efficiency will be considered.

By considering the kinematic model of the protein and the ligand as articulated objects, this problem can be treated as a path planning problem [19], [20] (Section 5). The aim of path planning algorithms is to compute collision-free paths that connect the start and the goal configuration of a given problem, possibly satisfying additional constraints. The problem is difficult, because it presents many degrees of freedom (DoFs). We solve this task by using sampling-based motion planning algorithms from the field of robotics.

State of the art motion planning algorithms, such as the Probabilistic Roadmap (PRM) [15] or the Rapidly-exploring Random Tree (RRT) [21], [22], can be adapted to be able to perform simulations of protein-ligand disassemblies (Section 4). We will use the RRT-based Exploration/Exploitation Tree (EET) planner [26] for molecular simulations, so Section 4 also recapitulates the basic idea of the EET.

We need to consider the different properties of the DoFs to overcome the high number of DoFs in protein-ligand complexes and to solve protein-ligand disassamblies efficiently. Backbone motion is especially expensive, because motions of the backbone make it necessary to recalculate many side chains that are affected by the motion. Thus, this work focuses on simulations of proteins with

[1] TGT and its inhibitor will be referred to by their PDB code 1ENU.
[2] HIV protease combined with its inhibitor will be referred to as 2PSV.
[3] Human AKT1 and its inhibitor will be referred to as 3O96.
[4] `http://www.rcsb.org`

fixed backbones. The differentiation between three different categories of DoFs is important for this work. These properties are the *position and orientation of the ligand with reference to the protein*, the *internal DoFs of the ligand*, and the *mobile side chains of the protein*. First, there is the motion of the ligand with regard to the protein. The tunnel from the surface of the protein to the deep active site is the crucial area to be searched. The tunnel presents a bottleneck, so it is efficient to approximate the location and form of the tunnel and exploit the information to steer the search towards that promising region. On the other side, sampling outside of the surface of the protein is irrelevant for solving the unbinding process, because the hard part is getting the ligand out of the protein, not the motion once it is outside the protein. The difficulty of overcoming the narrow tunnel is shown in the Figures 1, 2, and 3. The motion defining the *ligands position and orientation with respect to the protein* is crucial to extract the small ligand from the large protein. One could imagine pulling the ligand through the tunnel until it is outside of the protein to illustrate this motion. Secondly, it can become necessary for larger ligands to twist the ligand or bend some part of it in order to overcome narrow passages. This motion is realized with the *internal DoFs of the ligand*. At last, *side chains of the protein* can block the exit path and make it impossible for the ligand to get out of the tunnel. In that case, it is necessary to move the relevant side chains and clear the exit path for the ligand. With these three insights (Section 5) we come to the conclusion that we need to treat the three different kinds of DoFs individually.

We present an extended version of the Exploration/Exploitation Tree (EET) planner [26] to this account. Using the environmental information and balancing exploration and exploitation, the EET planner enables us to solve these complex problems efficiently. In Section 6 we introduce a modified implementation of the EET that deals with tree data structures with cycles, which cannot be simulated with the standard EET. In the standard EET the end-effector is the part of the robot where the forces are being attached to, for example a hand at the end of a robotic arm for object manipulation. Ligands have several possible end-effectors, for example an atom at the beginning or at the end of the structure, which we will call virtual end-effectors. We will show that the performance depends on the selected virtual end-effector. We increased the exploration of the *internal DoFs of the ligand* by introducing null space sampling. Thereby, the position and orientation of the ligand with respect to the protein does not change, but the individual parts of the ligand and the mobile side chains move.

Using the modified EET enables us to solve complex disassembly problems of protein-ligand interactions with large ligands with internal DoFs and side chains that restrict the direct exit path. The algorithm is able to solve these problems efficiently, so that protein-ligand unbinding processes can be simulated and planned in reasonable computing time that are unsolvable using the RRT-Connect (RRTCon) [18] implemented in the motion planning software Robotics Library (RL)[5] [27](Section 7).

[5] `http://www.roboticslibrary.org/`

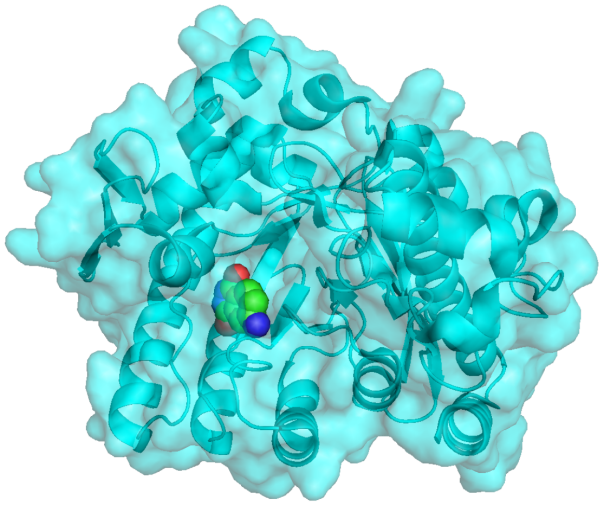


Figure 1: Protein-ligand interaction 1ENU consisting of tRNA-guanine transglycosylase (TGT) and its inhibitor 4-aminophthalhydrazide (APH). 1ENU is important for the development of a medicine against Shigellosis. In the shown start conformation the ligand is placed in the active site of the protein (transparent cyan). The ligand is colored using the PyMOL convention color by atom. Obtained using PyMOL [30].

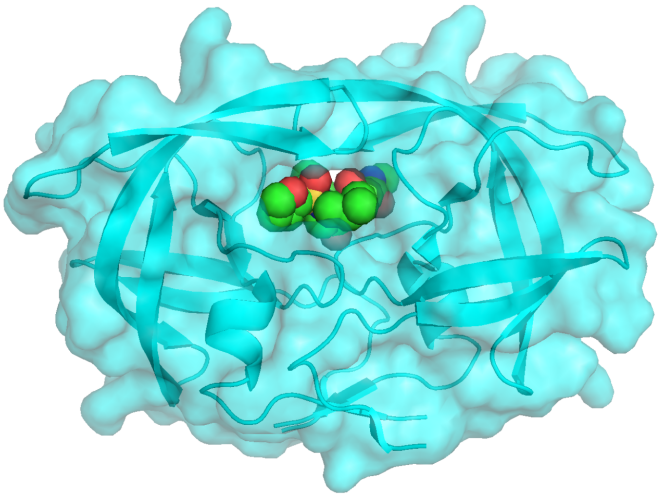


Figure 2: Protein-ligand interaction 2PSV consisting of HIV protease (HIVP) and the inhibitor CARB-KB45. In the shown start conformation the ligand is in the active site of the protein (transparent cyan). The ligand is colored using the PyMOL scheme color by atom. Obtained using PyMOL [30].

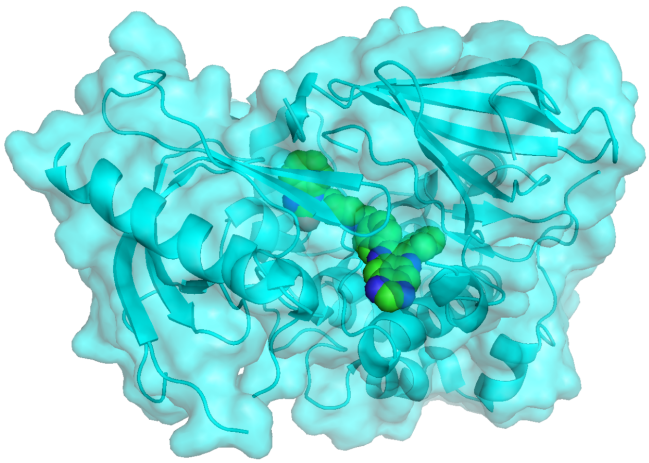


Figure 3: Protein-ligand interaction 3O96 consisting of human AKT1 and its inhibitor VIII. 3O96 is promising for the treatment of cancer. In the shown start conformation the ligand is placed in the deep active site of the protein (transparent cyan). The ligand is colored using the PyMOL convention color by atom. Obtained using PyMOL [30].

4 Related Work

In Molecular Dynamics and Virtual Screening, protein-ligand disassembly processes are simulated with energy functions as the selection criteria. AutoDock [23] is one of the several docking programs that are used today. Considering the energies and all the forces necessary to compute the energy of one conformation is computationally expensive, therefore we focus on less complex, but faster approaches, where simplified models are used.

Proteins and ligands are modeled as kinematic chains, so that sampling-based motion planning algorithms from the field of robotics can be applied to simulate proteins and ligands, and plan solution paths. State of the art motion planning algorithms are presented and based on that, how these algorithms are used for molecular simulations. The Exploring/Exploiting Tree (EET) planner [26] is presented as one of the motion planning algorithms, because we introduce an extended version of the EET in this work suitable for molecular simulations.

4.1 Protein modeling as kinematic chains

Proteins can be modeled as robots [25]. The atoms of the protein are modeled with spheres and all spheres form an articulated object. Molecular degrees of freedom (DoFs) are realized with mechanical DoFs [12] resulting in a kinematic chain. With these protein models, the unbinding of a ligand from the active site of the protein can be formulated as a motion planning problem and solved with sampling-based motion planning algorithms from the field of robotics.

4.2 Sampling-based motion planning algorithms

The Probabilistic Roadmap (PRM) [15] is the state of the art multiple-query motion planning algorithm. The Rapidly-exploring Random Tree (RRT) [21] is the state of the art single-query motion planning algorithm. Several efficient derivatives of the RRT have been developed, such as the RRT-Connect (RRTCon) [18] with an additional goal bias and the Exploring/Exploiting Tree (EET) [26], which balances exploratory and exploitative behavior.

4.2.1 PRM

PRM was the first sampling-based algorithm for motion planning and works well for robots in static workspaces [15]. The PRM constructs a roadmap that holds information about the connectivity of the configuration space and can be used for multiple planning queries. We use the approach of sampling-based planners to disassemble protein-ligand interactions. The advantage of a multiple-query approach for our problem would be that several possible exit paths could be generated after the construction of the roadmap is complete.

The goal of the algorithm is to construct a roadmap with valid configurations and valid transitions between the configurations. The first step of the PRM is choosing a random configuration. If the configuration is collision-free, a new node is added to the roadmap for that sample. Next, the nearest neighbors in the roadmap are found. The new node is connected to the neighbors via a new edge, if a collision-free connection is possible. These steps are repeated until a termination criterion is met. Figure 4 shows the schematic construction of

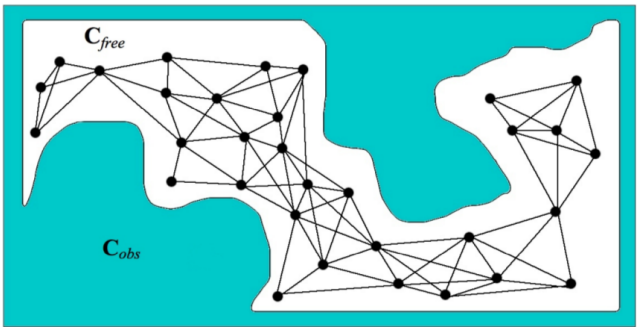


Figure 4: Schematic view of the construction of the PRM. Obtained from [5].

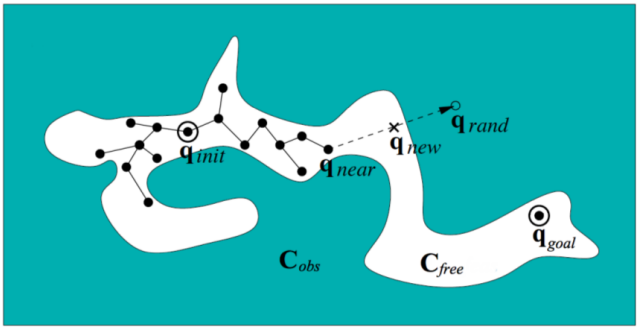


Figure 5: Schematic view of the construction of the RRT. Obtained from [5].

a PRM. After the construction phase, the roadmap can be used to solve any number of queries consisting of a start and a goal configuration. The PRM algorithm returns a valid path connecting the start and goal configuration, if such a path exists.

4.2.2 RRT

The RRT is a sampling-based motion planner and several variants of the RRT, especially the RRTCon and the RRTConCon, form the state of the art motion planners used in robotics, because they are able to handle high numbers of DoFs [21].

The goal of the RRT is to construct a tree that connects the start and the goal configuration, thus solving the path planning problem. The RRT constructs a tree that is rooted in the start configuration in order to explore valid configurations in the configuration space and to find a collision-free path between the start and the goal configuration. Figure 5 illustrates how an RRT is constructed. The first step of the RRT is drawing a random sample $\mathbf{q}_{\text{rand}}$ from configuration space. The existing tree is searched for the nearest neighbor $\mathbf{q}_{\text{near}}$ to $\mathbf{q}_{\text{rand}}$. A new node $\mathbf{q}_{\text{new}}$ is added to the tree between $\mathbf{q}_{\text{near}}$ and $\mathbf{q}_{\text{rand}}$ with a predefined distance d from $\mathbf{q}_{\text{near}}$, if $\mathbf{q}_{\text{new}}$ and the path from $\mathbf{q}_{\text{near}}$ to $\mathbf{q}_{\text{new}}$ are collision-free. An edge is added to the tree between $\mathbf{q}_{\text{near}}$ and $\mathbf{q}_{\text{new}}$. These steps are repeated until $\mathbf{q}_{\text{goal}}$ is connected to the tree and thereby to $\mathbf{q}_{\text{start}}$, or until a termination criterion is met.

4.2.3 EET

Rickert et al. present the Exploring/Exploiting Tree (EET) planner [26] as a computationally efficient method for motion planning. The idea is to balance exploration and exploitation. Exploration searches the configuration space without direct relation to the given task in order to gain a global understanding. Exploitation, on the other hand, searches the space using all available information and with guidance from the given task in order to solve the task as quickly as possible. The planner uses exploitation whenever possible, but when the information is insufficient, it shifts more towards exploration. One source of information is global connectivity, which is gained with a sphere-based wavefront expansion in the workspace. This information about the free parts of the workspace is used to steer the search into promising regions using an attractive force. Another source of information are obstacles, which are represented as repulsive forces, thus steering the search away from the obstacles. In order to balance exploration and exploitation, the planner uses the information whether tree expansion steps were successful or not. Successful tree expansion steps lead to exploiting behavior and failing tree expansion steps lead to exploratory behavior.

The EET planner outperforms all other state of the art planners, such as different variants of PRM and RRT, in several experiments. The standard EET is designed for the application in the field of robotics, so we adapt the algorithm to being able to handle protein-ligand disassembly problems.

4.3 Motion planning for proteins

PRM and RRT planners can be adapted for molecular simulations. Efficient motion planning algorithms are successfully used to simulate proteins.

4.3.1 PRM for molecular simulations

The category of the PRM-based planners for molecular simulations consists of the Probabilistic Conformational Roadmap (PCR) [32] and the Stochastic Roadmap Simulation (SRS) [6]. Both planners substitute the collision checking used in the standard robotics PRM with a probability function that favors low energy conformations [5] to decide whether a random sample is valid. The PCR and the SRS use weighted edges to connect two conformations in the roadmap. The weights represent the probability of one conformation transitioning into the other. The difference between the PCR and the SRS is the transition probability.

When it comes to searching high-dimensional space, PRM-based planners have the shortcoming of homogeneous search. That is a problem, because irrelevant conformation space is searched exhaustively when performing uniform sampling. The construction of the roadmap is therefore computationally expensive and inefficient. Thus, we will use a single-query approach for the benefit of a fast computation of one possible exit path and accept the loss of other possible exit paths. PRM-based planners treat all kinds of DoFs equally, thus being inefficient for computing paths for the ligand into or out of the protein.

4.3.2 RRT for molecular simulations

Cortés et al. introduce the idea of decoupling DoFs [11] by using active and passive configuration parameters for path planning for articulated objects. Solving a complex problem with many articulated DoFs is computationally expensive. By dividing the configuration parameters into active and passive, the active parts are essential for the disassembly, while the passive parts are only moved if they hinder the disassembly progress. The algorithm automatically identifies the passive parts that have to be moved. Thus, the computing time is reduced compared to the standard RRT and complex problems with hundreds of DoFs are manageable. The so called Manhattan-like RRT (ML-RRT) is derived from the RRT and can be used to solve the protein-ligand unbinding problems with mobile side chains efficiently. The side chains are the passive parameters and will only be moved if they hinder the active ligand or another passive side chain from progressing.

The ML-RRT presents a powerful approach to overcome the high number of DoFs in molecular simulations. By differentiating between active and passive configuration parameters, the idea of treating different kinds of DoFs individually is successfully implemented. We want to combine the strength of decoupling the DoFs with the strength of the EET.

5 Problem Description

The problem at hand is to simulate and plan the unbinding process of a ligand from the protein. Therefore, we model the ligand and the protein as robots, so that sampling-based motion planning algorithms can be applied to the problem. To overcome the problem of the high dimensionality of the search space, we partition the DoFs of the problem into different sets and the algorithm calculates the motion for each set individually. The problem of handling several possible virtual end-effectors is introduced.

5.1 Motion planning for proteins

Proteins can be modeled as robots [25]. The connections (bonds) between two atoms correspond to links and the rotations around these axes (ϕ and ψ angles for the backbone) are realized with joints (Fig. 6). These joints can be handled individually, so that each joint can be set either fixed, or mobile with a revolute joint. This enables us to build protein models with a fixed backbone, a set of fixed side chains, and a set of mobile side chains. The atoms are modeled as spheres with a radius corresponding to the van der Waals force of the atom. The van der Waals force is the main repulsive constraint between two non-bonded atoms, thus the geometric model of the protein must obey the van der Waals equilibrium distance [12]. Instead of the configuration space in robotics, we have the conformation space with proteins. With this knowledge we can simulate complex biological protein-ligand unbinding processes using efficient algorithms from the field of robotics.

The key question is whether it is possible for a specific ligand to reach the deep active site of a protein. This problem can be formulated as a path planning problem and the knowledge of planners in robotics can be exploited in order to solve this problem. $\mathcal{C}$ is the conformation space and one conformation $\mathbf{q}$ defines all translations and angles of the mobile parts with regard to the world frame. One conformation is composed of the *translation* $\mathbf{p}_{\text{ligand}}$ and the *rotation* $\mathbf{r}_{\text{ligand}}$ of the ligand, the joint angles $\mathbf{q}_{\text{ligand}}$ of the *internal DoFs of the ligand*, and the joint angles $\mathbf{q}_{\text{chains}}$ of all atoms that form the *mobile side chains of the protein*. Thus, a conformation vector is defined as $\mathbf{q} = \{\mathbf{p}_{\text{ligand}}, \mathbf{r}_{\text{ligand}}, \mathbf{q}_{\text{ligand}}, \mathbf{q}_{\text{chains}}\}$. Thereby, $\mathbf{p}_{\text{ligand}}$ and $\mathbf{r}_{\text{ligand}}$ correspond to insight (1), $\mathbf{q}_{\text{ligand}}$ to insight (2), and $\mathbf{q}_{\text{chains}}$ to insight (3) defined in Section 5.2.

The start and the goal position are given as $\mathbf{q}_{\text{start}}$ and $\mathbf{q}_{\text{goal}}$, where $\mathbf{q}_{\text{start}}$ is a conformation with the ligand being in the deep active site of the protein and $\mathbf{q}_{\text{goal}}$ is a conformation with the ligand being outside the surface of the protein (Fig. 1, 2, and 3). The goal is to find a collision-free path in $\mathcal{C}$ that connects $\mathbf{q}_{\text{start}}$ and $\mathbf{q}_{\text{goal}}$.

5.2 Three kinds of degrees of freedom

This is a complex problem, because of the high dimensionality of the conformation space $\mathcal{C}$, thus the search space. The problems addressed in this work have up to 30 DoFs. In order to handle this amount of DoFs, we have to differentiate between the different kinds of DoFs and treat these groups individually and according to their properties. There are three different kinds of DoFs.

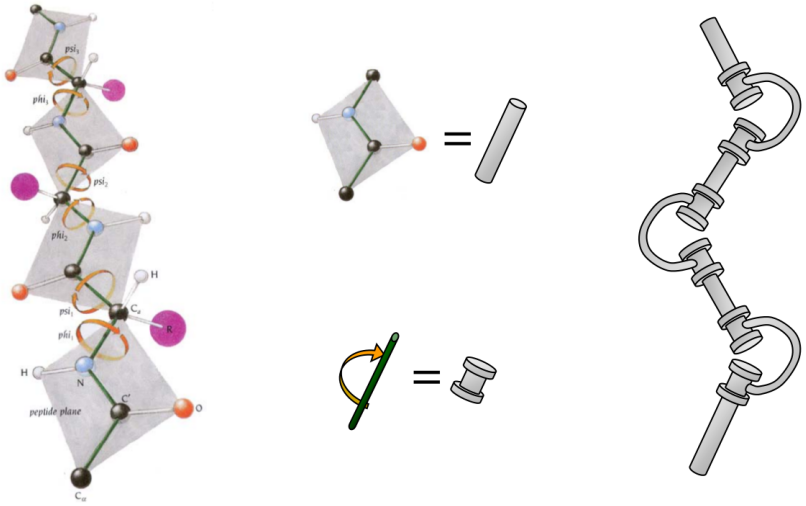
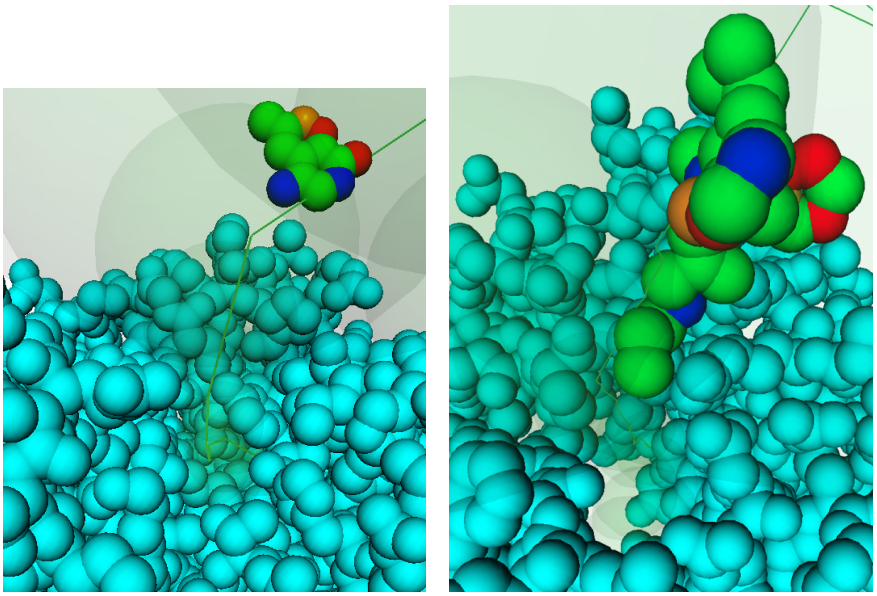


Figure 6: Connection between proteins and robots showing that proteins can be modeled as robots. The connections (bonds) between two atoms correspond to robotic links and the rotations around these axes (ϕ and ψ angles for the backbone) correspond to joints. Obtained from [8] and [9].

1. The *ligands position and orientation with regard to the protein*: These DoFs are crucial for finding a path and must be varied extensively. The information of the workspace tunnel provided by the sphere-based wavefront expansion of the EET is used in order to solve the problem efficiently. The spheres connect the start and the goal conformation and thereby generate an obstacle-free tunnel for the ligand (Fig. 7). This category will be referred to as insight (1).

2. The *internal DoFs of the ligand*: These DoFs become crucial, when internal motion of the ligand is necessary to exit the protein. The exploratory behavior of the EET applied to the internal DoFs of the ligand is increased by null space sampling (Section 6.1.2) to overcome this problem. The planner searches actively for ligand conformations, which enable the ligand to exit the protein tunnel. This challenge will be referred to as insight (2).

3. The *mobile side chains of the protein*: These DoFs are important, if side chains block the exit path. Most of the side chains are irrelevant for solving the unbinding problem, because they are located not in the tunnel. Thus, making all side chains mobile would be computationally inefficient and not add any value to the planning process. For those side chains relevant for solving the problem, the planner searches for protein conformations, where the side chains are pushed away and enable the ligand to exit the protein. The user can decide which side chains are relevant for the task and specify them to be mobile. This will be referred to as insight (3).



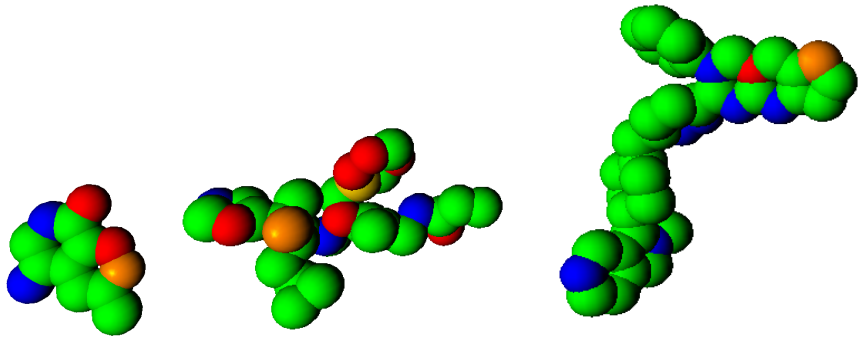
(a) (b)

Figure 7: Workspace tunnels (green spheres) provided by the sphere-based wavefront expansion of the EET. The workspace tunnel represents free workspace and connects the start and the goal conformation. The proteins are colored in cyan and the ligands by atom type using the PyMOL sheme. (a) Protein-ligand interaction 1ENU (b) Protein-ligand interaction 2PSV

5.3 Virtual end-effector

In the standard EET, the end-effector is the part of the robot where the forces are being attached to. Ligands have several possible end-effectors, because each atom is a possible end-effector. We call these virtual end-effectors. Figure 8 shows three different ligands with their chosen virtual end-effectors marked in orange. The PyMOL coloring scheme[6] [1] was used for the rest of the atoms: oxygen in red, nitrogen in blue, and carbon in green.

[6] `https://pymolwiki.org/index.php/Advanced_Coloring`



(a) (b) (c)

Figure 8: Ligands with chosen virtual end-effectors marked in orange. Ligands are colored by atom type using the PyMOL scheme. (a) 1ENU (b) 2PSV (c) 3O96

6 Method

To solve protein-ligand interactions efficiently, we introduce an extended version of the sampling-based motion planner EET. The EET already implements features that are promising for the decoupled motion planning for the different sets of DoFs. The information contained in the sphere-based wavefront expansion about free workspace is exploited to plan the *ligands position and orientation with reference to the protein* as efficiently as possible. The exploratory behavior of the EET is efficient to find favorable conformations of the *internal DoFs of the ligand* and the *mobile side chains of the protein* that enable the planner to solve the problem.

A kinematic model of the ligand and the protein is built for the molecular simulation using the extended EET planner.

6.1 EET adaptation

In the exploration phase, the EET planner creates a tree consisting of spheres that overlap each other and indicate free workspace. The center of each sphere is placed on the surface of the previous sphere and its radius is the distance to the nearest obstacle. The initial sphere is created around the start. The exploration phase terminates when the spheres connect the start and the goal of the workspace. This sphere-based wavefront provides an obstacle-free tunnel that connects start and goal.

In the exploration and exploitation phase, the EET planner uses the knowledge about the connectivity of the workspace contained in the sphere tree. In addition to the workspace tree, the EET constructs a conformation space tree similar to the RRT. New conformations are not sampled uniformly random, but alongside the centers of the workspace spheres. Thus, the ligand is pulled through the middle of the workspace tunnel using the chosen virtual end-effector as a control point. For the translation of the control point between workspace and conformation space, the pseudo-inverse of the Jacobian matrix is necessary. If collisions hinder the disassembly, the planner gradually shifts from exploiting the workspace connectivity to exploratory behavior.

The ligand has multiple possible end-effectors, but the control point must be well-defined and the pseudo-inverse of the Jacobian matrix can only be calculated for one end-effector. Also, the EET planner minimizes the motion needed for the transition from one conformation to the next, so that only the ligands position and orientation are considered, not the internal DoFs. This behavior is inefficient for protein-ligand interactions, because the internal motion of the ligand is often necessary to overcome narrow passages. So, two modifications are necessary to extend the EET planner for molecular simulations. These correspond to the insights (1) and (2) concerning the different categories of DoFs.

6.1.1 Virtual end-effectors

Ligands have several possible end-effectors, because each atom is a possible end-effector. We call these virtual end-effectors. For one end-effector, the Jacobian $\mathbf{J}$ consists of a matrix $\mathrm{x}_1 \in \mathbf{R}^6$. $\mathbf{J}$ is extended by another matrix $\mathrm{x}_\mathrm{i} \in \mathbf{R}^6$ for each additional virtual end-effector i. For calculations, such as the inverse of $\mathbf{J}$,

Table 1: Symbols and variables used in the algorithms

Element	Description
$\mathcal{C}$	Conformation space
$\mathbf{I}$	identity matrix
$\mathbf{J}$	Jacobian matrix
$\mathbf{J}^{\dagger}$	transposed Jacobian matrix
$\mathbf{p}$	position of the ligand with reference to the protein
$\mathbf{q}$	joint conformation with position $\mathbf{p}$ and orientation $\mathbf{r}$ of the ligand with reference to the protein
$\mathbf{r}$	orientation of the ligand with reference to the protein
$\mathbf{T}$	workspace frame with position and orientation

only one virtual end-effector of the set of possible end-effectors is active, so the block concerning the active end-effector is extracted from $\mathbf{J}$ and used as $\mathbf{J}$.

The performance depends on the chosen virtual end-effector. The results in Section 7.4 show that pulling a ligand out of the tunnel is more efficient than pushing it. Attaching forces to the virtual end-effector in order to pull it out of the tunnel implements insight (1).

6.1.2 Null space sampling

Using null space sampling [16], [24] increases the exploration of the *internal DoFs of the ligand*. Thereby, insight (2) is implemented.

The EET planner balances exploitation and exploration to build a conformation space tree [28]. For the expansion of the conformation space tree a CONNECT step (Algorithm 1) is applied. A CONNECT step consists of executing several EXTEND steps (Algorithm 2) until the sample is reached or further EXTEND steps become impossible due to collisions. The notation used in the algorithms is explained in Table 1. Line 5 of Algorithm 2 minimizes the motion necessary to fulfill the translation from workspace to conformation space, but we want the opposite behavior and increase the internal motion. Thus, we add null space sampling (Algorithm 2, line 7). The NULL SPACE SAMPLING algorithm described in Algorithm 3 computes the null space displacement in conformation space by first taking a random sample (line 1) and then setting the translation and the rotation part of the vector to zero (line 3). Thus producing internal motion of the ligand, while the pose of the ligand with regard to the protein does not change.

Algorithm 1 CONNECT algorithm

Require: $\mathbf{q}_{\text{near}}, \mathbf{T}_{\text{sample}}$
Ensure: $\mathbf{q}_{\text{new}}$
$\mathbf{q}_{\text{new}} \leftarrow \emptyset$
repeat
  $\mathbf{q}_{\text{new}'} \leftarrow \text{EXTEND}(\mathbf{q}_{\text{near}}, \mathbf{T}_{\text{sample}})$
  if $\mathbf{q}_{\text{new}'} \neq \emptyset$ **then**
    $\mathbf{q}_{\text{near}} \leftarrow \mathbf{q}_{\text{new}}$
    $\mathbf{q}_{\text{new}} \leftarrow \mathbf{q}_{\text{new}'}$
  end if
until $\mathbf{q}_{\text{new}'} = \emptyset$
return $\mathbf{q}_{\text{new}}$

Algorithm 2 EXTEND algorithm

Require: $\mathbf{q}_{\text{near}}, \mathbf{T}_{\text{sample}}, \Delta\mathbf{q}_{\text{null}}$
Ensure: $\mathbf{q}_{\text{new}}$
if SINGULAR($\mathbf{q}_{\text{near}}$) **then**
  $\mathbf{q}_{\text{new}} \leftarrow \text{RAND}()$
else
  $\Delta\mathbf{x} \leftarrow \mathbf{T}_{\text{near}} - \mathbf{T}_{\text{sample}}$
  $\Delta\mathbf{q}_{\text{task}} \leftarrow \mathbf{J}^{\dagger}(\mathbf{q}_{\text{near}})\Delta\mathbf{x}$
  $\Delta\mathbf{q}_{\text{null}} \leftarrow (\mathbf{I} - \mathbf{J}^{\dagger}(\mathbf{q}_{\text{near}})\mathbf{J}(\mathbf{q}_{\text{near}}))\Delta\mathbf{q}_{\text{null}}$
  $\Delta\mathbf{q} \leftarrow \Delta\mathbf{q}_{\text{task}} + \Delta\mathbf{q}_{\text{null}}$
  $\mathbf{q}_{\text{new}} \leftarrow \mathbf{q}_{\text{near}} + \Delta\mathbf{q}$
end if
if $\mathbf{q}_{\text{new}} \in \mathcal{C}_{\text{free}}$ **then**
  return $\mathbf{q}_{new}$
else
  return $\emptyset$
end if

Algorithm 3 NULL SPACE SAMPLING algorithm

Require: $\mathbf{q}_{\text{near}}$
Ensure: $\Delta\mathbf{q}_{\text{null}}$
$\mathbf{q}_{\text{null}} \leftarrow \text{RAND}()$
$\Delta\mathbf{q}_{\text{null}} \leftarrow \mathbf{q}_{\text{null}} - \mathbf{q}_{\text{near}}$
$\Delta\mathbf{q}_{\text{null}} \leftarrow (\mathbf{p}_{\text{zero}}, \mathbf{r}_{\text{zero}}, \Delta\mathbf{q}_{\text{null}})$

6.2 Ligand and protein model

The internal data structure used to model the kinematic structure of the ligand and the protein is explained in the following.

The absolute coordinates of the atoms are extracted from the model deposited in the RSCB Protein Data Bank (PDB) [7][7]. Additionally, the connect information is used for the ligand model and the order of the residues for the protein model. The kinematic structures are modeled as graphs, where nodes represent atoms and edges represent the connections between two atoms. For

[7] `http://www.rcsb.org`

the ligand model the connections are obtained from the PDB connect information. For the protein model each residue is added to the graph and thereby the backbone atoms $N - C_\alpha - C$ are connected to the previous backbone atoms and the side chain atoms are added and connected depending on the residue type.

Each joint can be handled individually, so it can be set either fixed or mobile. This enables us to build ligand models with internal DoFs and protein models with a fixed backbone, a set of fixed side chains, and a set of mobile side chains. Atoms in cycles are set to be fixed, so they cannot move independently of the rest of the cycle atoms. Cycles can only be moved altogether. The minimum spanning tree is calculated from the graph structure in order to get a valid kinematic model without cycles. One edge in every cycle is lost in this step, but the model behaves correctly, because the cycles are fixed and cannot break apart. Atoms with only one neighbor in the tree are called leaves. Leaves are set to be fixed in the model, because they do not change the conformation, but cost computing time by adding a DoF.

The free-flying ligand has six DoFs: three for the x-, y- and z-direction, which are realized with three prismatic joints, and three around the x-, y- and z-axis, which are realized with three revolute joints. These six DoFs define the *ligands position and orientation with reference to the protein*. Each mobile, meaning non-fixed, atom of the ligand adds a DoF to the ligand model. These DoFs define the *internal DoFs of the ligand* and their number depends on the size and complexity of the ligand. The backbone of the protein is fixed, because considering backbone motion is computationally expensive and not necessary for the problems at hand. Each *mobile side chain of the protein* adds the amount of mobile atoms that are part of the side chain to the protein model.

N neighbors in the tree structure of the ligand and the protein will be ignored for each atom during the collision checks, so that the ligand and the protein do not collide with themselves in the natural start conformation. Bonded atoms cannot be in collision.

The tree data structure provides a flexible representation of the ligand and the protein. Thus, enabling the user to specify an arbitrary set of *mobile side chains*, which is necessary to implement insight (3).

7 Experimental Results

The main experiment focuses on solving three protein-ligand interactions with different challenges using the RRT-Connect (RRTCon) and three variants of the extended EET. The challenges correspond to the insights (1) to (3) concerning the different kinds of DoFs. The first experiment focuses on an example protein-ligand interaction where the six DoFs of the *ligands position and orientation with regard to the protein* are sufficient to solve the problem. The second test shows how the *internal DoFs of the ligand* become necessary in addition to the first six DoFs to solve a more complex unbinding. In the last experiment, the *mobile side chains of the protein* have to be simulated in addition to the previous DoFs in order to solve the problem.

Additionally, an experiment showing that the performance of the extended EET depends on the selected virtual end-effector is presented.

7.1 Test setup

The RRTCon planner and the EET planner are implemented in the motion planning software Robotics Library (RL)[8] [27]. The planners are applied to protein-ligand interactions of different complexities. The size of the ligand and the size and location of the active site and the corresponding tunnel affect the difficulty of the problems. Protein-ligand interactions will be referred to by their PDB code. In this work three different protein-ligand interaction problems are addressed:

1. 1ENU: For the development of drugs against Shigellosis, inhibitors of tRNA-guanine transglycosylase (TGT) are important [13]. One promising inhibitor is 4-aminophthalhydrazide (APH). The complex consisting of TGT and APH is deposited with the PDB code 1ENU in the PDB. The ligand of 1ENU (Fig. 8a) is small and compact. Together with a comparatively easily reachable active site (Fig. 1) this makes it a simple problem. The disassembly process of 1ENU is shown in Figure 9 and in this video [2].

2. 2PSV: HIV protease (HIVP) transforms newly synthesized virus proteins into their functional and infectious form. Thus, it is important to design mutation-resistant HIVP inhibitors and one promising candidate is CARB-KB45 [10]. The complex consisting of the HIVP and CARB-KB45 is listed in the PDB with the PDB code 2PSV. The ligand of 2PSV (Fig. 8b) is complex and has 16 internal DoFs. It has to change its orientation to exit the active site and its motion is restricted by side chains (Fig. 2). The size and internal agility of the ligand together with the narrow active site make this a hard problem. The disassembly of 2PSV is shown in Figure 10 and in this video [3].

3. 3O96: Human AKT1 pathway inhibitors are promising for the treatment of cancer, because a dysregulation of the AKT1 pathway correlates with several human cancers [33]. The allosteric Inhibitor VIII bound to AKT1 is deposited in the PDB with the PDB code 3O96. The ligand of 3O96 (Fig. 8c) is large, but has only five internal DoFs. The cycles forming the

[8] `http://www.roboticslibrary.org/`

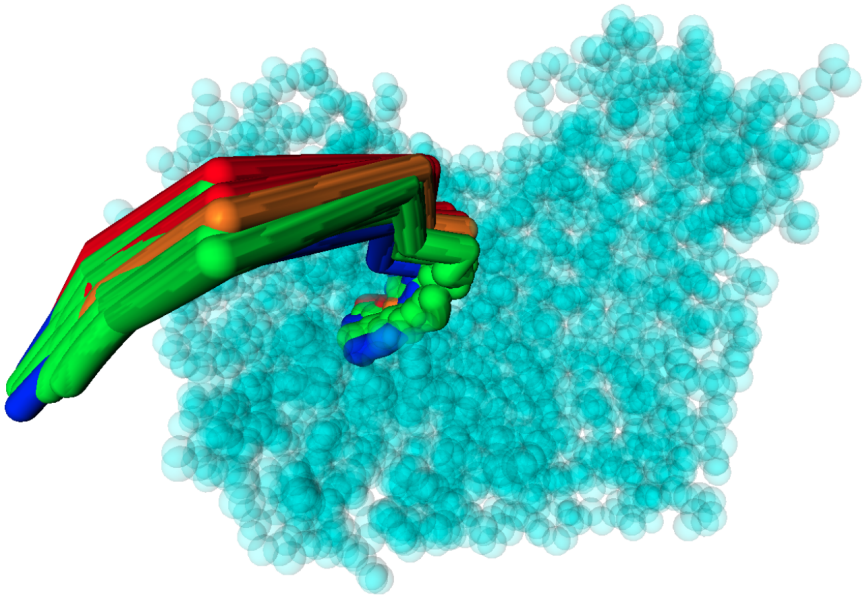


Figure 9: Unbinding process of the protein-ligand interaction 1ENU. Complete disassembly process with the ligands start conformation being in the active site of the protein and the goal conformation outside the surface of the protein. The protein is colored in transparent cyan and the ligand is colored by atom type using the PyMOL convention. The solution path was calculated with the extended EET planner for protein-ligand interactions using the *ligands position and orientation with regard to the protein*.

ligand restrict its internal agility. Together with the very long and very narrow tunnel (Fig. 3) this is a hard problem. The disassembly of 3O96 is shown in Figure 11 and in this video [4].

The RRTCon planner and three variants of the EET planner are applied to each problem. The variants of the EET are characterized by the kinds of DoFs that are active. Successively another kind of DoFs is added to the simulation so that the tests correspond to the insights (1)-(3).

1. RRTCon: The RRTCon is a variant of the RRT (Section 4.2.2). Is uses an additional goal bias by trying to connect each new node with the goal conformation.

2. EET standard: The EET standard samples different conformations of the *ligands position and orientation with regard to the protein*. Six DoFs are considered in the motion planning. The EET standard implements insight (1).

3. EET internal DoFs: The EET internal DoFs additionally considers the internal mobility of the ligand. The number of DoFs added to the simulation depends on the *internal DoFs of the ligand*. This variant corresponds to insight (2).

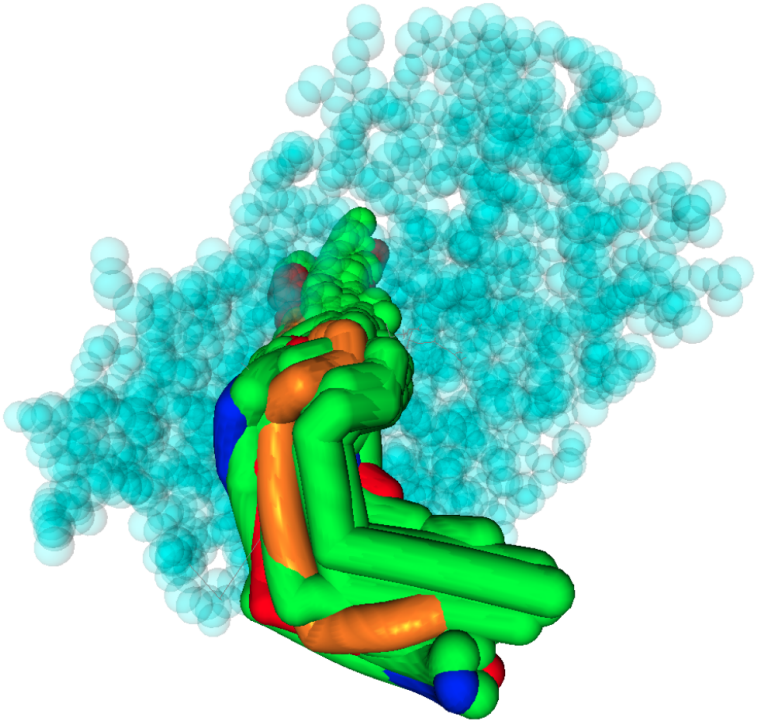


Figure 10: Unbinding process of the protein-ligand interaction 2PSV. Complete disassembly process with the ligands start conformation being in the active site of the protein and the goal conformation outside the surface of the protein. The protein is colored in transparent cyan and the ligand is colored by atom type using the PyMOL convention. The solution path was calculated with the extended EET planner with *internal DoFs of the ligand*.

4. EET side chains: The EET side chains considers all three kinds of DoFs, so additionally to the *ligands position and orientation*, and the *internal DoFs of the ligand*, this variant also simulates the *mobile side chains of the protein*. The number of DoFs added to the planning process depends on the side chains that are selected to be mobile by the user. Insight (3) is realized with the EET side chains.

The empirical performance analysis was performed on a lenovo ThinkPad with a 2.20 GHz processor (Intel Core i5-5200U) and 12 GB RAM.

7.2 Results

The numerical results for the simulation of the protein-ligand unbinding presented in Table 2 and Table 3 were obtained by using 70% and 80% of the van der Waals radii, respectively. The tables display the averaged computing time with the standard deviation, the number of nodes, the number of collision queries, and the number of active DoFs which were used by the four planners to solve the three problems. For the EET versions, the exploration time with

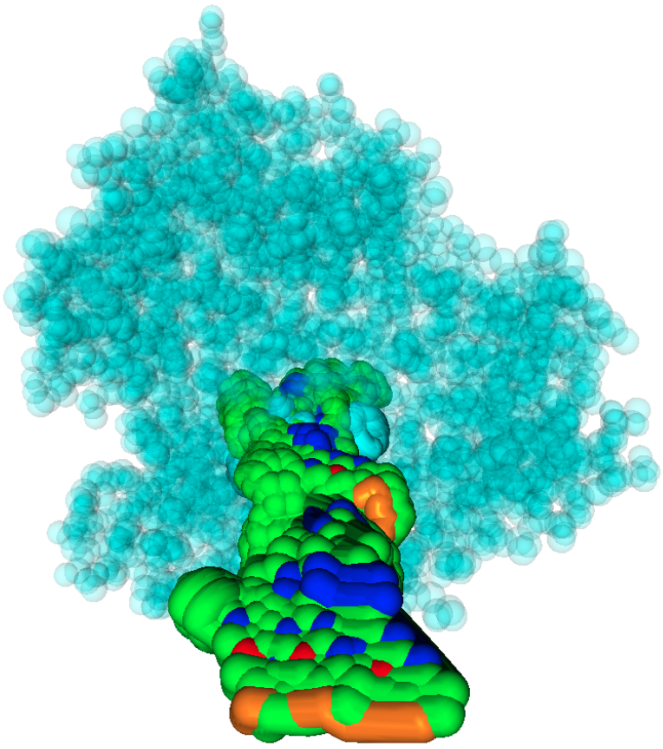


Figure 11: Unbinding process of the protein-ligand interaction 3O96. Complete disassembly process with the ligands start conformation being in the deep active site of the protein and the goal conformation outside the surface of the protein. The protein is colored in transparent cyan and the ligand is colored by atom type using the PyMOL convention. The solution path was calculated with the extended EET planner with *internal DoFs of the ligand* and *active side chains of the protein*.

the standard deviation is listed additionally. The numerical results are averaged over ten runs. The computing time was limited to 15 hours. If a planner was not able to solve the problem within this time, the experiment was aborted and considered as a failure.

The EET side chains is able to solve the protein-ligand interaction 3O96 with 80% of the van der Waals radii with a success rate of 90%, when the termination criterion is set to a maximum computing time of 20 min. Only the successful runs are considered in the statistic.

7.3 Analysis

The variants of the EET outperform the RRTCon. The RRTCon is unable to solve the two hard problems in reasonable computing time (Fig. 12), whereas the EET is able to solve all problems given that the appropriate kinds of DoFs are active. The decision on which kinds of DoFs and which side chains are to be set active is a trade-off. The more DoFs are active, the more problems can be

Table 2: Numerical results using 70% of the van der Waals radii

		1ENU	2PSV	3O96
RRTCon	T[s]	14 ± 9	–	–
	N_{nodes}	121	–	–
	$N_{queries}$	131336	–	–
	N_{DoFs}	6	22	6
EET	T[s]	8 ± 1	–	188 ± 100
standard	$T_{explore}$[s]	5 ± 1	–	0.21 ± 0.04
	N_{nodes}	32	–	1088
	$N_{queries}$	6579	–	89061
	N_{DoFs}	6	6	6
EET	T[s]	8 ± 1	35 ± 23	358 ± 317
internal DoFs	$T_{explore}$[s]	5 ± 1	0.2 ± 0.04	0.20 ± 0.02
	N_{nodes}	32	296	1584
	$N_{queries}$	6579	13473	133919
	N_{DoFs}	6	22	11
EET	T[s]	21 ± 11	44 ± 24	168 ± 132
side chains	$T_{explore}$[s]	1 ± 0	1 ± 0.15	0.24 ± 0.05
	N_{nodes}	38	378	925
	$N_{queries}$	8448	17768	66059
	N_{DoFs}	7	29	16

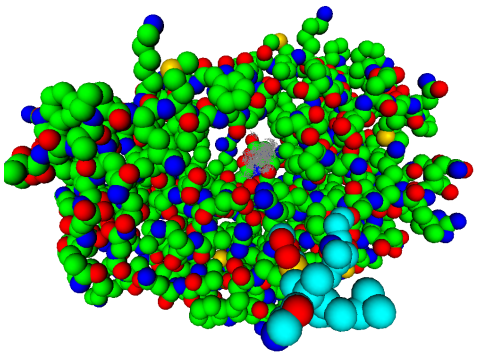


Figure 12: 2PSV problem not solved with RRTCon. Ligand in goal conformation outside of the protein. Projection of the search tree (gray) centering around the active site of the protein, which is the ligands start conformation.

solved that otherwise would be unsolvable. On the other hand, the computing time is increased exponentially with additional DoFs that have to be considered in the simulation. For easy problems the computing time is increased if too many DoFs are active. It follows that only the necessary kinds of DoFs should be selected to be active for each problem and a minimum set of mobile side chains should be used.

Table 3: Numerical results using 80% of the van der Waals radii

		1ENU	2PSV	3O96
RRTCon	T[s]	11 ± 6	–	–
	N_{nodes}	122	–	–
	$N_{queries}$	88341	–	–
	N_{DoFs}	6	22	6
EET	T[s]	9 ± 4	–	–
standard	$T_{explore}$[s]	0.49 ± 0.06	–	–
	N_{nodes}	47	–	–
	$N_{queries}$	7659	–	–
	N_{DoFs}	6	6	6
EET	T[s]	9 ± 4	62 ± 48	–
internal DoFs	$T_{explore}$[s]	0.49 ± 0.06	0.1 ± 0.01	–
	N_{nodes}	47	771	–
	$N_{queries}$	7659	36798	–
	N_{DoFs}	6	22	11
EET	T[s]	13 ± 3	144 ± 87	475 ± 252
side chains	$T_{explore}$[s]	0.42 ± 0.06	0.10 ± 0.01	0.27 ± 0.05
	N_{nodes}	50	1116	2032
	$N_{queries}$	8130	65726	172620
	N_{DoFs}	7	29	16

7.4 Virtual end-effector experiment

The performance of the extended EET depends on the chosen virtual end-effector. For this experiment the protein-ligand interaction 2PSV is simulated using two different virtual end-effectors. The virtual end-effector with the ID 1594 is close to the goal conformation, thus pulling the ligand from the active site and out of the protein. On the other hand, the virtual end-effector with the ID 1573 is as far away from the goal conformation as possible, so that the ligand is pushed through the tunnel towards the goal conformation. The EET variants internal DoFs and side chains are used to solve the two problems. For the simulations the percentage of the van der Waals radii is set to 80%.

The results in Table 4 show that pulling the ligand out of the tunnel is more efficient than pushing it. The table displays the averaged computing time with the standard deviation, the exploration time with the standard deviation, the number of nodes, the number of collision queries, and the number of active DoFs which were used by the two variants of the EET planner to solve the problems. The numerical results are averaged over ten runs.

The problem is easy to solve if an appropriate virtual end-effector is selected, but becomes unsolvable in reasonable computing time if a pushing virtual end-effector is chosen. This shows that, depending on the problem, all available information on how to best select the virtual end-effector, in this case a pulling one, should be exploited to solve the problem as efficient as possible.

Table 4: Numerical results using different virtual end-effectors

		pulling (ID 1594)	pushing (ID 1573)
EET	T[s]	62 ± 48	–
internal DoFs	$T_{explore}$[s]	0.1 ± 0.01	–
	N_{nodes}	771	–
	$N_{queries}$	36798	–
	N_{DoFs}	22	22
EET	T[s]	144 ± 87	–
side chains	$T_{explore}$[s]	0.10 ± 0.01	–
	N_{nodes}	1116	–
	$N_{queries}$	65726	–
	N_{DoFs}	29	29

8 Conclusion and Future Work

The extended versions of the EET planner introduced in this work efficiently solve protein-ligand disassembly problems. The problem of the high dimensionality of the search space is overcome by a partition of the DoFs and a decoupled motion calculation for each set of DoFs. The results show that the extended versions of the EET outperform the RRTCon in solving medically relevant protein-ligand interactions. Problems with complex ligands become manageable, because the internal mobility of the ligands is handled efficiently by the EET internal DoFs version. Relevant side chains can be selected and simulated using the EET side chains version, enabling us to solve protein-ligand interactions where side chains hinder the ligand from exiting the protein. The more side chains have to be moved and considered in the planning, the complexer becomes the problem and the computing time increases significantly.

A future extension of the algorithm will automatically identify the relevant side chains and add them to the set of mobile side chains. Also, an automated selection of promising virtual end-effectors is to be implemented. The solutions of these additional problems can be integrated into the current EET version and improve the usability.

The motion of the backbone should also be taken into account in the future. In practice, backbone motion is relevant for protein-ligand interactions. Simulating backbone motion is computationally even more expensive, because small backbone motions cause recalculations of all affected side chains to be necessary.

To enhance the drug designing process, a platform for automated testing of a database of ligands against the relevant protein, that is to be inhibited or activated, should be developed. This would produce a selection of ligands that, from a mechanical point of view, are able to reach the active site of the protein. Based on this selection, other simulations considering the energies of the conformations would follow. This procedure can be economically reasonable, because of the reduced computational cost when using the extended EET to solve protein-ligand interactions.

References

[1] Pymol wiki advanced coloring. `https://pymolwiki.org/index.php/Advanced_Coloring`. Accessed: 2017-03-10.

[2] Video showing the disassembly process of the protein-ligand interaction 1enu. `https://youtu.be/KFzD1a5EHWY`. Accessed: 2017-03-24.

[3] Video showing the disassembly process of the protein-ligand interaction 2psv. `https://youtu.be/kjdJENt8Rg4`. Accessed: 2017-03-24.

[4] Video showing the disassembly process of the protein-ligand interaction 3o96. `https://youtu.be/xeTU6ZD5N0Q`. Accessed: 2017-03-24.

[5] Ibrahim Al-Bluwi, Thierry Siméon, and Juan Cortés. Survey: Motion planning algorithms for molecular simulations: A survey. *Comput. Sci. Rev.*, 6(4):125–143, 2012.

[6] M. S. Apaydin, C. E. Guestrin, C. Varma, D. L. Brutlag, and J.-C. Latombe. Stochastic roadmap simulation for the study of ligand- protein interactions. *Bioinformatics*, 18 (Suppl 2):18–26, 2002.

[7] Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. The protein data bank. *Nucleic acids research*, 28(1):235–242, 2000.

[8] Carl Ivar Branden et al. *Introduction to protein structure*. Garland Science, 1999.

[9] Oliver Brock. Computational biology slides. 2015.

[10] Sripriya Chellappan, GS Kiran Kumar Reddy, Akbar Ali, Madhavi NL Nalam, Saima Ghafoor Anjum, Hong Cao, Visvaldas Kairys, Miguel X Fernandes, Michael D Altman, Bruce Tidor, et al. Design of mutation-resistant hiv protease inhibitors with the substrate envelope hypothesis. *Chemical biology & drug design*, 69(5):298–313, 2007.

[11] J. Cortés, L. Jaillet, and T. Siméon. Disassembly path planning for complex articulated objects. *IEEE Transactions on Robotics*, 24(2):475–481, 2008.

[12] Juan Cortés, Thierry Siméon, V Ruiz De Angulo, David Guieysse, Magali Remaud-Siméon, and Vinh Tran. A path planning approach for computing large-amplitude motions of flexible molecules. *Bioinformatics*, 21(1):116–125, 2005.

[13] Ulrich Grädler, Hans-Dieter Gerber, DeeAnne M Goodenough-Lashua, George A Garcia, Ralf Ficner, Klaus Reuter, Milton T Stubbs, and Gerhard Klebe. A new target for shigellosis: rational design and crystallographic studies of inhibitors of trna-guanine transglycosylase. *Journal of molecular biology*, 306(3):455–467, 2001.

[14] Dragos Horvath. A virtual screening approach applied to the search for trypanothione reductase inhibitors. *Journal of Medicinal Chemistry*, 40(15):2412–2423, 1997.

[15] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996.

[16] Oussama Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, 1987.

[17] Douglas B Kitchen, Hélène Decornez, John R Furr, and Jürgen Bajorath. Docking and scoring in virtual screening for drug discovery: methods and applications. *Nature reviews Drug discovery*, 3(11):935–949, 2004.

[18] James J Kuffner and Steven M LaValle. Rrt-connect: An efficient approach to single-query path planning. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 2, pages 995–1001, 2000.

[19] J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, New York, NY, 1991.

[20] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, New York, NY, 2006.

[21] Steven M. Lavalle. Rapidly-exploring random trees: A new tool for path planning. Technical report, Computer Science Department, Iowa State University, 1998.

[22] Steven M. LaValle and James J. Kuffner. Randomized kinodynamic planning. *The International Journal of Robotics Research*, 20(5):378–400, 2001.

[23] Garrett M Morris, David S Goodsell, Robert S Halliday, Ruth Huey, William E Hart, Richard K Belew, Arthur J Olson, et al. Automated docking using a lamarckian genetic algorithm and an empirical binding free energy function. *Journal of computational chemistry*, 19(14):1639–1662, 1998.

[24] Yoshihiko Nakamura, Hideo Hanafusa, and Tsuneo Yoshikawa. Task-priority based redundancy control of robot manipulators. *The International Journal of Robotics Research*, 6(2):3–15, 1987.

[25] David Parsons and John F Canny. Geometric problems in molecular biology and robotics. In *ISMB*, pages 322–330, 1994.

[26] M. Rickert, O. Brock, and A. Knoll. Balancing exploration and exploitation in motion planning. *IEEE International Conference on Robotics and Automation*, pages 2812–2817, 2008.

[27] Markus Rickert. *Efficient Motion Planning for Intuitive Task Execution in Modular Manipulation Systems*. Dissertation, Technische Universität München, Munich, Germany, 2011.

[28] Markus Rickert, Arne Sieverling, and Oliver Brock. Balancing exploration and exploitation in sampling-based motion planning. *IEEE Transactions on Robotics*, 30(6):1305–1317, 2014.

[29] Peter Ripphausen, Britta Nisius, Lisa Peltason, and Jürgen Bajorath. Quo vadis, virtual screening? a comprehensive survey of prospective applications. *Journal of Medicinal Chemistry*, 53(24):8461–8467, 2010.

[30] Schrödinger, LLC. The PyMOL molecular graphics system, version 1.8. November 2015.

[31] Brian K. Shoichet. Virtual screening of chemical libraries. *Nature*, 432(7019):862–865, 2004.

[32] A. P. Singh, J.-C. Latombe, and D. L. Brutlag. A motion planning approach to flexible ligand binding. *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pages 252–261, 1999.

[33] Wen-I Wu, Walter C Voegtli, Hillary L Sturgis, Faith P Dizon, Guy PA Vigers, and Barbara J Brandhuber. Crystal structure of human akt1 with an allosteric inhibitor reveals a new mode of kinase inhibition. *PLoS One*, 5(9):e12913, 2010.