Computational approaches to investigate structural and functional properties of transmembrane proteins

Dissertation
zur Erlangung des Grades
des Doktors der Naturwissenschaften
der Naturwissenschaftlich–Technischen Fakultät III
Chemie, Pharmazie, Bio- und Werkstoffwissenschaften
der Universität des Saarlandes

von

Po-Hsien Lee

Saarbrücken
März 2014

Tag des Kolloquiums:	25. July, 2014
Dekan:	Prof. Dr. Volkhard Helms
Berichterstatter:	Prof. Dr. Albrecht Ott
Vorsitz:	Prof. Dr. Karin Römisch
Akad. Mitarbeiter:	Dr. Sonja Keßler

Acknowledgement

I am very glad and thankful to do my PhD studies in Prof. Volhkard Helms' group and in Germany. Prof. Helms provided me a position, interesting projects, and substantial support for my research. My work is funded by Deutsche Forschungsgemeinschaft (DFG) through Graduiertenkolleg (GK) 1276 and Sonderforschungsbereiche (SFB) 1027. I would like to thank all my colleagues and friends who helped me and cooperated with me during these years. Especially:

Dr. Tihamér Geyer supervised me to finish the project of Brownian dynamics simulation. He taught me a lot about physics and how physicists deal with scientific problems.

Dr. Micheal Hutter put many efforts on polishing my English for each of my manuscripts before submission.

Dr. Wei Gu helped me to settle down when I arrived at Saarbrücken several years ago.

Mrs. Kerstin Gronow-Pudelek, the secretary of our group, did a lot of tedious paper works for me.

Prof. Albrecht Ott and Dr. Mikhail Zhukovsky cooperated with me in the CXCR4 docking project.

Prof. Patrick Huber and Sebastian Mörz performed the experiments of cytochrome c permeation and had helpful discussions with us about the protein translocation simulations.

Duy Nguyen cooperated with me in the project of pore-lining residue prediction.

Of course, I would also like to thank all my lovely Taiwanese friends in Saarbrücken who have helped me for my daily life and accompanied me during these years.

Special thanks go to my father who enlightened and aroused my interests in science during my childhood.

Kurzfassung

Die vorliegende Dissertation behandelt die Entwicklung bioinformatischer Methoden zur Untersuchung struktureller und funktionaler Eigenschaften von Transmembranproteinen.

Zunächst haben wir mit PROPORES eine Software entwickelt, die zur Identifikation von Poren sowie porenständigen Residuen bei bekannten Proteinstrukturen verwendet werden kann und die im Gegensatz zu den meisten grid-basierten Methoden orientierungsunabhängig funktioniert.

In einer Fallstudie verwendeten wir PROPORES, um porenständige Residuen von alpha-helikalen Transmembranproteinen in einem nicht-redundanten Datensatz zu identifizieren. Basierend auf diesen Daten haben wir die Software PRIMSIPLR entwickelt, um sequenzbasierte Vorhersagen für die besagten Residuen der Proteine zu treffen.

Darüberhinaus haben wir mithilfe Brown'scher Dynamiksimulationen die Translokation von Protein-ähnlichen Partikeln durch Nanoporen errechnet. Dabei war festzustellen, dass die geometrische Einschränkung enger Poren zu einer Anhäufung der großen Partikel am Poreneingang führt, was wiederum eine reduzierte Diffusität der Proteinmoleküle im Inneren der Pore zur Folge hat.

Experimentelle Studien lassen vermuten, dass die Bindung von Cholesterol an den CXCR4 Transmembranrezeptor eine wichtige Rolle bei dessen Funktion spielt. Da dieser als Corezeptor auch bei der HIV-Infektion eine wichtige Rolle hat, verwendeten wir Dockingsimulationen, um mögliche Cholesterolbindungsstellen auf der lipidständigen Oberfläche zu identifizieren.

Summary

In this thesis, we present the development and implementation of computational methods to investigate structural and functional properties of transmembrane proteins. To identify pores and pore lining-residues of known protein structures, we developed the toolkit PROPORES. PROPORES is a grid-based method that avoids the orientation dependence of most grid-based approaches. As an application, we used PROPORES to identify pore-lining residues of a non-redundant set of α-helical transmembrane protein structures. Based on this dataset, the tool PRIMSIPLR was developed to predict pore-lining residues of α-helical transmembrane proteins from their sequences. Besides, we performed coarse-grained Brownian dynamics simulations for the translocation of protein-like particles through nanopores. We found that the geometric constriction by the narrow pore leads to an accumulation of the larger particles at the pore entrance, which in turn compensates for the reduced diffusivity of the protein particles inside the pore. Docking simulations were also performed to identify possible cholesterol binding sites on the lipid exposable surface of the protein CXCR4. Experimental studies show that cholesterol binding is important for the function of CXCR4 transmembrane receptor which is a coreceptor for HIV infection. We hope these studies can provide useful microscopic details, structural features, and suggest possible mechanisms to the experimental biologists working on transmembrane proteins.

Zusammenfassung

Heutzutage stellen bioinformatische Vorhersagemethoden in der Strukturbiologie einen etablierten Standard dar. Die vorliegende Dissertation beschreibt die Entwicklung von bioinformatischen Methoden zur Vorhersage struktureller und funktionaler Eigenschaften von Transmembranproteinen sowie deren Anwendung. Transmembranproteine spielen eine wichtige Rolle bei einer Vielzahl von biologischen Prozessen, beispielsweise bei der Signalverarbeitung oder in Stoffwechselvorgängen. Als Mediator zwischen der wässrigen Umgebung in der Zelle und der hydrophoben Lipiddoppelschicht in der Membran verfügen diese Proteine typischerweise über Poren oder Konkavitäten, die spezifisch Liganden aufnehmen oder Substrate durch die Zellmembran hindurch transportieren. Zur Identifikation solcher Poren sowie der porenständigen Residuen auf Basis von Proteinstrukturen haben wir die Software PROPORES entwickelt. Mit ihrer Hilfe kann man für den Transmembrankanal eines Proteins eine Trajektorie durch den Kanal sowie dessen Radius bestimmen. Diese Funktionalität ist vergleichbar mit etablierten Programmen wie PASS, HOLE, und MolAxis. Zusätzlich ist unsere Software in der Lage die Konnektivität zweier benachbarter Poren zu überprüfen. Zu diesem Zweck können die Residuen an den jeweiligen Porenenden sterisch sinnvoll repositioniert werden. PROPORES ist eine Gitter-basierte Methode, und im Gegensatz zu den meisten Gitter-basierten Methoden ist unsere Methode orientierungsunabhängig. Trotzdem bleibt die exponentiell wachsende Laufzeit bei Verwendung feinerer Auflösungen ein zu berücksichtigender Aspekt. Allerdings steckt in der Parallelisierung unserer Methode Potential zu deren Performancesteigerung.

Zu Testzwecken haben wir einen Datensatz mit nicht-redundanten Strukturen α-helikaler Transmembranproteine zusammengestellt und die porenständigen Residuen durch PROPORES bestimmen lassen. Solche Residuen sind verantwortlich für die Funktion von Membrantransportern und -kanälen, weil sie in direktem Kontakt mit den Substraten oder dem umgebenden Wasser stehen.

Anschließend haben wir die Software PRIMSIPLR zur sequenzbasierten Vorhersage von porenständigen Residuen α-helikaler Transmembranproteine entwickelt. Dazu trainierten wir eine Supportvektormaschine auf Basis von Features, die aus dem gleichen Datensatz wie oben abgeleitet wurden. Diese Features beinhalten evolutionäre Informationen wie die Bewertung von Sequenzpositionen und deren evolutive Konservierung. Obwohl hierfür vom physikochemischen Standpunkt aus kein offensichtlicher Zusammenhang bekannt ist bzw. festgestellt werden konnte, lieferten diese Features einen entscheidenden Beitrag zur Verbesserung unserer Vorhersagemethode. So gelang es auf dem gleichen Testdatensatz bessere Ergebnisse zu erzielen als mit der bereits publizierten Methode MEMSAT-SVM. Eine Evaluierung der Resultate ergab, dass sich die meisten fälschlich klassifizierten Residuen innerhalb der Membran befinden. Um künftig auch solche Residuen korrekt unterscheiden zu können, müssen spezifischere und aussagekräftigere Features gefunden und implementiert werden.

Einen dynamischen Ansatz zur Untersuchung der Proteintranslokation durch Nanoporen haben wir mit einer coarse-grained Brown'schen Dynamik-Simulation verfolgt. In diesem Modell wurden die Proteine als kugelförmige Einheiten, verbunden durch Federpotentiale,

nach dem Vorbild des Cytochroms C repräsentiert. Die Poren wurden als zylindrische Tunnel mit verschiedenen Größenverhältnissen modelliert. Die Diffusion haben wir mithilfe eines Konzentrationsgradienten über die Membran hinweg angeregt. Um den Rechenaufwand zu reduzieren haben wir diesen Gradienten mithilfe zweier virtueller Grenzflächen modelliert, mit denen man eine konstante externe Teilchenkonzentration einstellen kann. Die Ergebnisse zeigen, dass sich unsere errechneten Translokationsraten für gefaltete Proteine mit der analytischen Lösung von Brunn *et al.* decken. Darüber hinaus konnten wir feststellen, dass ein engerer Porendurchmesser und die damit verbundene geometrische Einschränkung zu einer Anhäufung großer Partikel am Poreneingang führt, wodurch ein reduziertes Diffusionsvermögen für Proteine innerhalb der Poren kompensiert wird. Das Modell für die ungefalteten Proteine zeigte, dass sich hierbei die längeren Polymerketten entlang der Porenachse orientieren. Dadurch reduzieren sich die Freiheitsgrade im System und damit auch die Translokationsrate der Proteine. Diese Analysen stellen einen ersten Versuch zur Untersuchung der Translokation von Proteinen durch biologische Membranen dar. Zukünftig ließen sich solche Translokationsprozesse durch die Einbindung weiterer Strukturmerkmale realistischer abbilden.

Abgesehen von ihrer Funktion als Substrattransporter können Membranproteine allosterisch reguliert werden. Diese Regulierung findet durch Wechselwirkungen der Proteinoberfläche mit der Membran statt. In experimentellen Studien wurde gezeigt, dass die Bindung von Cholesterol eine bedeutende Rolle für die Funktion des CXCR4 Transmembranrezeptors spielt, der wiederum ein Korezeptor für eine HIV-Infektion ist. Um systematisch und effizient mögliche Bindungsstellen für Cholesterol an der lipid-ständigen Oberfläche von Transmembranproteinen

zu finden, haben wir Docking-Simulationen mit einem Divide-and-Conquer Ansatz verknüpft. Da die von Autodock verwendete Scoringfunktion ursprünglich für wässrige Lösungen ausgelegt ist, haben wir vier Kristallstrukturen von Cholesterol-Proteinkomplexen testweise erneut gedockt. Die Resultate bestätigen, dass Autodock zur Identifizierung von Cholesterol-Bindungsstellen – auch in der Membran - geeignet ist. Anschließend haben wir das gleiche Docking zur Untersuchung der Wechselwirkungen zwischen CXCR4 und Cholesterol angewandt. Die vorhergesagte Bindungsstelle mit der höchsten Affinität befindet sich in der Einkerbung zwischen den Transmembranhelices 1 und 7 nahe der inneren Membran-Wasser-Grenzschicht. Ein nahestehendes Lysin formt eine Wasserstoffbrücke zwischen seiner ε-Aminogruppe und der Hydroxyl-Gruppe des Cholesterols. Weitere Interaktionen entstehen durch einen Tyrosinring mit der aromatischen Seitenkette des Cholesterols sowie durch weitere hydrophobe Kontakte mit unpolaren Seitenketten. Durch Sequenzaligments konnten wir zeigen, dass ähnliche mutmaßliche Cholesterolbindungsstellen ebenfalls im CCR5 HIV-Korezeptor zu finden sind. Wir vermuten, dass eine Mutation der Cholesterolbindungsstellen bei CXCR4 und CCR5 diese Proteine unempfindlich gegen die Cholesterolkonzentration in der Membran machen könnten. Diese Erkenntnisse könnten bei der Wirkstoffentwicklung zur Therapie von AIDS sowie anderer Krankheiten, bei denen CXCR4 und CCR5 involviert sind, eine Rolle spielen.

Im Rahmen der Dissertation wurden verschiedene bioinformatische Methoden zur Untersuchung von Transmembranproteinen im Hinblick auf unterschiedliche Aspekte entwickelt und implementiert. Ihr Zweck ist es, mikroskopische Details sowie strukturelle Eigenschaften und mögliche Mechanismen für die experimentelle Arbeit mit Transmembranproteinen aufzuklären.

Abstract

Computational approaches are now extensively being used in structural biology. We developed and implemented computational methods to investigate structural and functional properties of transmembrane proteins. Transmembrane proteins play crucial roles in varieties of biological processes such as signal transduction and material exchange. To perform these specific functions in between the aqueous environment and the hydrophobic lipid bilayer, transmembrane proteins usually have concavities or pores to accommodate ligands or to transport substrates. We developed a toolkit PROPORES to identify pores and pore lining-residues of known protein structures. For proteins with long channels inside, our tool can depict the trajectory of the channels and provide the radii along the channel. These two functions are comparable with well known tools such as PASS, HOLE and MolAxis. Our toolkit has a novel function to check the connectivity of two neighboring pores by reposition the gating residues in a sterically allowed way. PROPORES is a grid-based method but it avoids the orientation dependency of most grid-based approaches. However, the exponential growth of running time with finer resolution is still an issue. To parallelize the implementation during pore identification is a possible way to improve the performance.

As an application, we collected a non-redundant set of α-helical transmembrane protein structures and used PROPORES to annotate the pore-lining residues of these structures. Pore-lining residues are crucial for the function of membrane transporters and channels because they directly contact the substrates or the water shell surrounding them. We developed a tool PRIMSIPLR to make predictions of the pore-lining residues for α-helical transmembrane

proteins from their sequences. We trained a support vector machine by the features extracted from the non-redundant set. The features containing evolutionary information such as position specific scores and conservation scores made a substantial contribution for the prediction whereas physicochemical properties had no apparent effect. Our prediction method outperformed to the tool MEMSAT-SVM on a non-redundant testing set. We found that most false classifications are between pore-lining residues and non-pore-lining residues located inside the membrane. Finding more specific and representative features to distinguish these two types of residues remains a tough challenge for future work.

With a dynamic perspective on cross-membrane transportation, we performed coarse-grained Brownian dynamics simulations of protein translocation through nanopores. Proteins were represented as single beads or bead-spring polymers modeled after the protein cytochrome *c*. Pores were modeled as cylindrical holes through the membrane with various aspect ratios. Diffusion was driven by a concentration gradient across the membrane. The concentration gradient was mimicked by two constant density interfaces what saved a lot of computational effort. Our results for the flow rates of folded protein models show good agreement with an analytical solution derived by Brunn *et al.* We also found that the geometric constriction by the narrow pore leads to an accumulation of the larger particles at the pore entrance, which in turn compensates for the reduced diffusivity of the protein particles inside the pore. For the unfolded models, the longer polymers showed stronger orientation preference along the pore axis which restricted their motions and reduced the translocation rate. The system of this study is a starting point to investigate protein translocation through

biological membrane. More structural details can be further added to this system to yield a more realistic presentation of protein translocation processes.

Beside the transport function related to their pore and pore-lining residues, the function of transmembrane proteins may be regulated allosterically due to the interactions on the lipid exposable surface of the protein. There are experimental studies showing that cholesterol binding is important for the function of CXCR4 transmembrane receptor which is a coreceptor for HIV infection. To identify the possible cholesterol binding site on the lipid exposable surface of the protein, we performed docking simulations in a divide-and-conquer scheme to have a systematic search of binding site in an efficient way. In addition, to validate the scoring function of AutoDock which is originally calibrated for protein-ligand interactions in aqueous solution, we performed a redocking exercise on four transmembrane protein structures that were co-crystallized with cholesterol molecules. The redocking results confirmed the suitability of AutoDock for finding cholesterol binding sites on the lipid accessible surface of membrane proteins. We thus employed the same docking analysis for predicting interactions between CXCR4 and cholesterol. The predicted binding site with the highest affinity is located in the groove between transmembrane helices 1 and 7 near the inner membrane-water interface. A lysine nearby to this region establishes a hydrogen bond between its ε-amino group and the hydroxyl group of cholesterol, a tyrosine ring stacks with cholesterol by its aromatic side chain, and several non-polar residues form hydrophobic contacts with cholesterol. Sequence alignment analysis showed that a similar putative cholesterol binding site is also present in another HIV coreceptor, CCR5. We propose that mutation of putative cholesterol binding residues will make CXCR4 and CCR5 insensitive to the membrane cholesterol concentration. We

hope that it will be useful in drug discovery for the therapeutic control of AIDS and other diseases in which CXCR4 and CCR5 are involved.

In this thesis, we developed and implemented multi-scale computational approaches to study transmembrane proteins from different aspects. Our purpose is to provide useful microscopic details, structural features, and possible mechanisms to the experimental biologists working on transmembrane proteins.

Contents

List of figures

List of tables

1 Introduction

1.1 Biological membranes ([1] and the references therein)

The function of a biological membrane is to separate cellular and subcellular compartments from their surroundings. The three key components of the biological membrane are lipids, sugars and proteins (Figure 1.1).

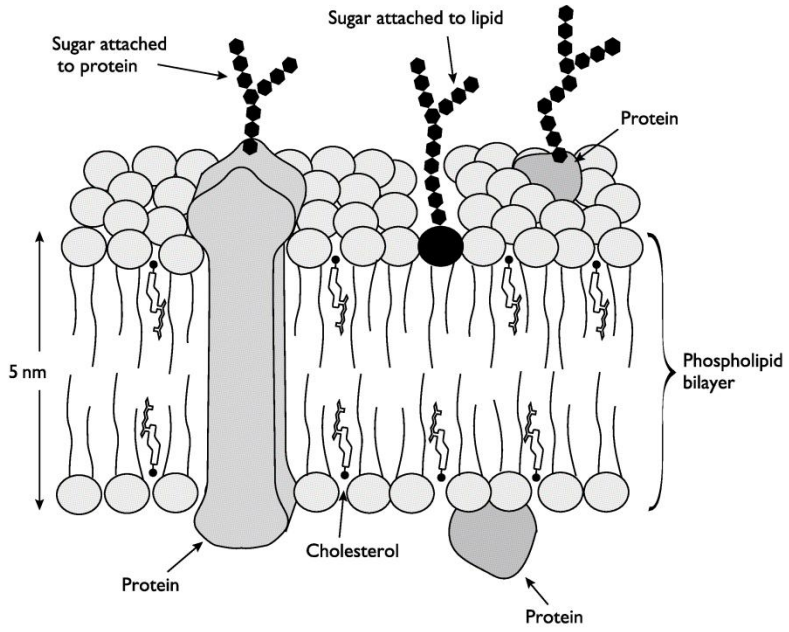

Figure 1.1 Cartoon representation of a biological membrane (this figure was adapted from [1]).

Membrane lipids are amphipathic molecules that are composed of both hydrophobic and hydrophilic groups. The three main types of lipids found in the biological membrane are phospholipids, glycolipids and cholesterol. The chemical structure of a phospholipid (Figure 1.2) contains fatty acids and a phosphate group linked to a glycerol. The fatty acid chain can be

saturated or unsaturated. An unsaturated fatty acid contains between 1 and 4 double bonds in *cis* form. Another type of phospholipid is sphingo-phospholipids which contain sphingosine instead of glycerol to link the fatty acid and phosphate as shown in Figure 1.3. Glycolipids share common structural features with phospholipids. The difference between them is that glycerol (or sphingosine) links to a sugar group (glucose or galactose) in glycolipids instead of phosphate. The structure of cholesterol contains a fused ring structure, a hydroxyl group, and a hydrocarbon chain (Figure. 1.4). Cholesterol is found mainly in the mammalian cell membrane.

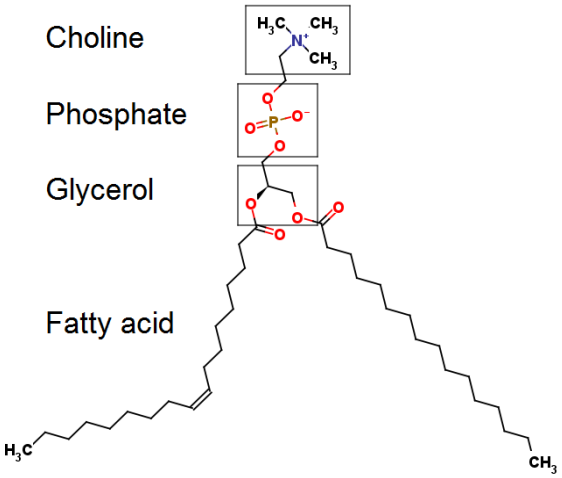

Figure 1.2 Phosopholipids are composed of a glycerol linked to fatty acid chains and a phosphate group.

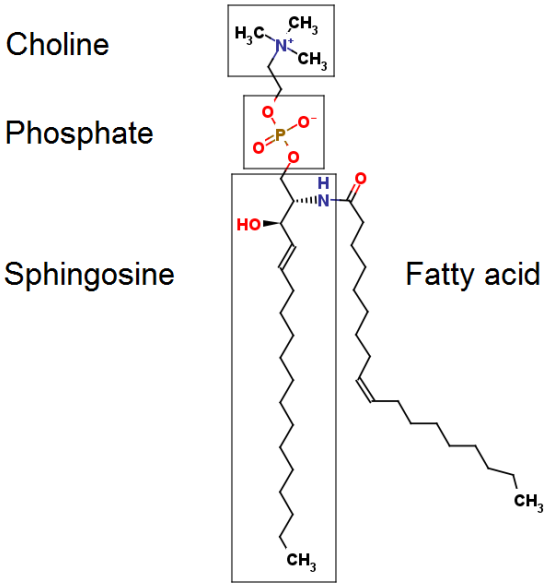

Figure 1.3 The structure of a sphingo-phospholipid contains sphingosine instead of glycerol to link fatty acid and phosphate.

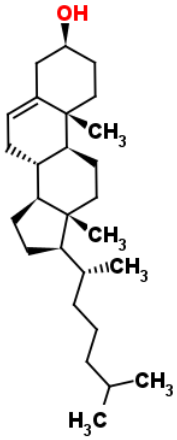

Figure 1.4 Cholesterol contains a fused ring structure, a hydroxyl group, and a shorter hydrophobic tail.

The biological membrane is also known as lipid bilayer. The hydrophobic tails of two lipid leaflets face inside whereas the hydrophilic head groups face to the outer aqueous environments (Figure 1.1). Lipid bilayers behave like two dimensional fluids. For example, the lateral diffusion of phospholipids is quite fast and its 2D diffusion coefficient is about 1 $\mu m^2/s$. However, the movement of lipids in the normal direction of the membrane (termed flip-flop) is a rare event because the hydrophilic head group has to go through a thick layer of hydrophobic

tails. However, the more efficient flip-flop motion of particular lipids may be facilitated by a family of transmembrane protein enzymes called flippases.

The fluidity of the membrane is also affected by some factors such as temperature, fatty acid composition, and cholesterol content. It is obvious that the higher temperature results in a higher kinetic energy of the system. The system is more active and the motion (translations, rotations, and vibrations) of the lipid molecules becomes faster. This makes the lipid molecules more disordered especially the long fatty acid chains. At high temperature, the lipid bilayer is in a liquid state and more fluid. In contrast, the lipid bilayer is in the gel state at low temperature when the long hydrophobic tails of lipids are close and orderly packed. The temperature where gel and liquid states interconvert is called the transition temperature. The fatty acid composition also has an effect on the transition temperature of the lipid bilayer. Unsaturated fatty acid chains have bends at the positions of double bonds (Figures 1.2 and 1.3) and that makes the packing of lipid molecules more difficult and lowers the transition temperature. Besides that, shorter hydrocarbon chains of lipids have weaker interactions between each other than long chains and this also lowers the transition temperature. The effect of cholesterol on the postulated lipid rafts is more complicated. The fused-ring structure of cholesterol interacts with the adjacent hydrocarbon chains of phospholipids or glycolipids what makes the lipid raft stiffer and less fluid. However, at high concentration, cholesterols disrupt the order packing of phospholipids and inhibit the possible phase transition.

In addition to the lipids, the other two components of the biological membrane are sugars and proteins. Specific functions such as material transportation or exchange are carried out by proteins. More details about this will be given in the next section. Sugar groups can attach to

either proteins or lipids on the outer lipid raft. It is involved in recognition mechanisms such as immune response due to the large number of combinations and varieties of sugar (carbohydrates) molecular structures on the cell surface. In recent years, more and more research focuses on the carbohydrates on the cell surface and tries implementing therapies for cancer and other diseases based on this.

1.2 Transmembrane proteins ([2] and the references therein)

Lipid bilayers form a permeability barrier from the surroundings and also provide the structural framework of cells or organelles while sugars play an important role in cell-type recognition. The specific functions of cell membranes are carried out by the assistance of elaborate membrane protein machineries. These machineries comprise a variety of transmembrane proteins such as transporters, channels, receptors and enzymes (Figure 1.5). They mediate material exchange or signal transduction across the membrane so that organelles or even entire cells can maintain specific concentrations to perform their intrinsic functions. For example, G-protein coupled receptors (GPCR) trigger signal transduction pathways inside the cell after ligands bind to the extracellular side of the membrane. [3] ATP-binding cassette (ABC) transporters utilize the energy from ATP hydrolysis to facilitate the translocation of a wide range of substrates from ions to oligopeptides and lipids. [4] The translocon complexes Sec61 and SecYEG integrate newly synthesized proteins into the membrane or translocate them across the membrane. [5] The tetrameric K^+ channels regulate electrical potential or maintain the balance of electrolytes across the membrane by rapid and highly selective permeation of potassium ions. [6] The water-specific membrane channel protein, aquaporin, regulates water

homeostasis in different kinds of cells. [7] The trimeric AcrB proteins function as multidrug efflux pump in gram-negative bacteria and are associated with the resistance against various antibiotics. [8] The malfunction of binding or translocation mechanism of these proteins may be related to diseases. On the contrary, a regular transmembrane protein may be used by a virus to infect the host cell. For example, as a member of GPCR, the CXCR4 receptor plays an important role in cell trafficking triggered by its signaling transduction. However, X4-tropic HIV-1 uses CXCR4 as a co-receptor to infect CD4+ T cells. [9] All mentioned transmembrane proteins and their related diseases are hot topics in the fields of novel therapy and drug discovery.

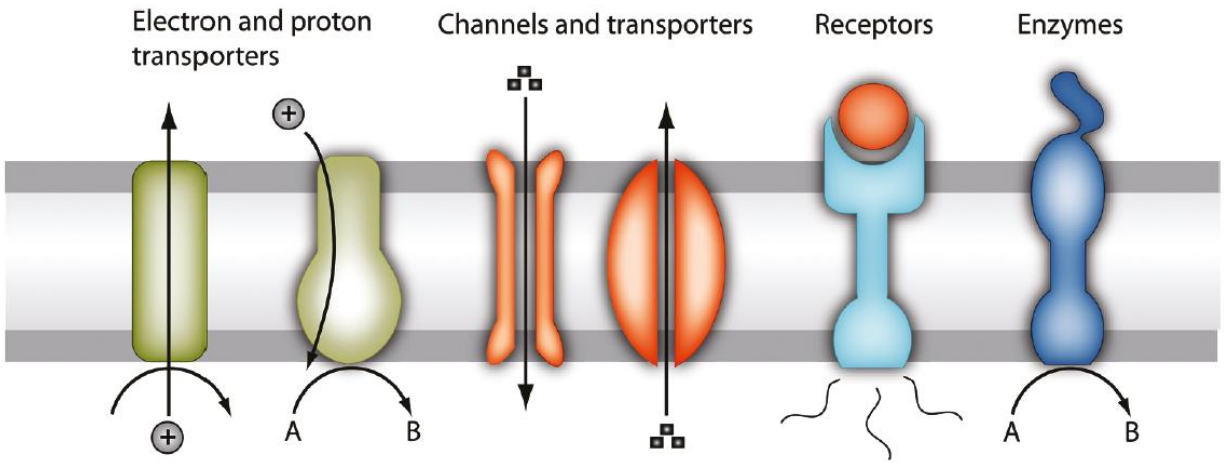

Figure 1.5 Four main types of transmembrane proteins classified by their functions. (This figure was adapted from [2]).

From the structural point of view, two main groups of transmembrane proteins have been identified, namely α-helical and β-barrel transmembrane proteins. An α-helical transmembrane protein contains one to several transmembrane helices. Each helix is at least 20 residues long and the composition is usually dominated by nonpolar residues because of the hydrophobic surroundings. β-barrel transmembrane proteins are formed by regular aligned β-strands in

anti-parallel manner. The residues on a β-strand point inward to the ligand or outward to the lipids alternatively. This structural property will also be present in the sequence pattern so that hydrophobic and hydrophilic residues are in alternating positions if the protein accommodates a polar ligand. In contrast, there is no clear pattern when the ligands are nonpolar.

The integration of transmembrane proteins into the membrane is mediated by the translocon. One of the most studied systems is the SRP-dependent Sec pathway. The SRP (signal recognition particle) protein first recognizes and binds to the signal peptide of a nascent polypeptide chain emerging from the ribosome. SRP then binds to the receptor on the membrane and the ribosome with polypeptide is guided to form a big complex with the translocon. The nascent polypeptide chain is threaded into the pore of translocon and the process is driven by the energy from hydrolysis of ATP. The function of the translocon is either to translocate proteins to the other side of the membrane or to integrate them into the membrane. The mechanism of distinguishing between translocation and integration is still not clear. One assumption is based on a partition process that the hydrophobic helix tends to be integrated into lipids whereas the more hydrophilic helix prefers to be transferred.

1.3 Structure bioinformatics on transmembrane proteins

Since X-ray crystallographic data on transmembrane proteins is relatively rare, dozens of structural prediction methods addressing various features of these proteins have been developed. [10] For example, OCTOPUS [11] and MEMSAT-SVM [12] predict the topology (including the positions of transmembrane helices, re-entrant helices and signal peptides) of α-helical transmembrane proteins. Interestingly, Nugent and Jones also developed an extended

version of MEMSAT-SVM that predicts pore-lining helices and residues. [13] Tools such as TMX [14] and BTMX [15] predict the burial status (residue exposed to lipid bilayer or not) of each residue in α-helical and β-barrel transmembrane proteins, respectively. TMHcon [16] and MEMPACK [17] predict the helix-helix contact map of α-helical transmembrane proteins. All of these methods adopt a common strategy to predict structural features from the protein primary sequence. First, they consider evolutionary information of the protein that is typically represented by the position specific scoring matrix (PSSM) derived from multiple alignments of homologous sequences. Second, all methods use machine learning algorithms such as support vector machines (SVM), artificial neural networks (ANN), and hidden Markov models (HMM) to build a model according to the features extracted from a set of membrane proteins with known 3D structures.

To have a possible 3D model of an unknown transmembrane protein, we would use a sequence alignment algorithm (e.g. BLAST [18]) to find the homologous proteins in the protein structure database PDB [19] and build the model based on the found templates. This approach is known as homology modeling. If no homologous proteins can be found based on sequence similarity, threading or fold recognition methods (e.g. Phyre [20]) provide the possibility to find the distantly related protein templates with common fold. The difference between homology modeling and fold recognition is due to the scoring matrix used. The scoring matrix used in homology modeling is a substitution matrix which describes the observed rate of change between residue pairs. However, the scoring matrix used in fold recognition emphasizes on the preference of a residue type to take on a specific local structure and the matrix may combine the information from evolution, physicochemical properties and even the prediction results of

other structural traits. *ab initio* methods (e.g. Rosetta [21]) are another approach to build a protein 3D model if there is no reliable template found by the previous two approaches. It predicts the protein structure from sequence based on a force field or a scoring function derived from known protein structures. It is noted that the highest accuracy or the best quality of the protein model will be obtained by homology modeling whereas the *ab initio* approach seems to be suitable for small peptides up to 60-80 amino acids in length currently.

Once the protein structure is resolved, tens of existing algorithms or tools are ready to be used to analyze the structural traits of the protein (ex. secondary structures, contact maps, electrostatic properties, pore identification and so on). For transmembrane proteins, methods such as TMDET [22] and PPM [23] predict the spatial arrangement inside the membrane (including position and orientation). For example, TMDET determines the position of a protein in the membrane by maximizing the relative hydrophobic membrane exposed surface area. PPM optimizes three structural parameters to obtain the minimal transfer energy of the protein from water to the lipid bilayer. The parameters are the thickness of the hydrophobic slab, the tilt angle between the protein and the membrane normal, and the position of the protein along the membrane normal.

1.4 Research topics in this thesis

To correctly and efficiently identify pore lining residues in the structures of transmembrane proteins, we tested several published software and web servers which will be introduced in Section 2.1. However, some issues (including the orientation dependency of the structure, manual determination of the search start point, failure on specific cases) were found in these

tools. Thus, we decided to develop a new algorithm that is based on the ideas of two published tools. Chapter 3 presents this novel algorithm to identify pores and pore-lining residues of a protein 3D structure that resolves the mentioned issues. This tool is then implemented for the identification of pore-lining residues in our non-redundant set of α-helical transmembrane proteins. The first prediction method for pore-lining residues was published by Nugent and Jones in 2012. [13] Their training data set seems small and the performance of prediction needs improvement. As described in Chapter 4, we compiled a non-redundant, comprehensive dataset of pore-containing α-helical transmembrane proteins. A machine learning algorithm was trained and optimized by features of this dataset and then used to make predictions of pore-lining residues for novel protein sequences of α-helical transmembrane proteins. Besides the properties of pores, we are also interested in the functional mechanism of pores in transmembrane proteins. Protein translocation across the membrane is the essential mechanism of live cells. However, many simulation studies focus on the translocation of the much longer and stiffer double-strand DNA, whereas few simulation studies dealt with the behaviors of relatively small and flexible proteins when they move through a pore. So as described in Chapter 5, we performed coarse-grained Brownian dynamic simulations for protein translocation through nanopores. A variety of pore dimensions and protein particles were considered to investigate the phenomenon occurring near the pore entrance. In addition, in cooperation with Prof. Ott's group, we conducted a specific case study of a transmembrane protein, CXCR4. CXCR4 had been addressed in many previous studies because of its substantial role in HIV infection. Several experimental studies pointed out that cholesterol has an effect on the ratio of HIV infection. Thus, we designed a divide-and-conquer docking scheme to

thoroughly search the most possible cholesterol binding site on the membrane exposed surfaces of CXCR4. This case study is presented in Chapter 6.

1.5 Aims of this thesis

We developed and implemented multi-scale computational approaches (from atomistic models to coarse-grained models, from the primary sequence to the tertiary structure) to study transmembrane proteins. We would like to show how the approaches we adopted including simulations, machine learning, and docking can be well implemented in these studies. Our purpose is to provide useful microscopic details, structural features, and possible mechanisms for experimental biologists working on transmembrane proteins. The theories, motivations, main ideas and implementations are presented in detail in the following chapters. The last chapter summarizes the main conclusions.

2 Theory

2.1 Brief review of pore identification algorithms

Here, we briefly review a few published pore identification algorithms and explain the main ideas behind them. Basically, these algorithms can be classified into four categories by the way in which they partition the protein structures.

The first class encompasses the most popular strategy of grid-based methods and includes the programs POCKET [24], LIGSITE [25, 26], dxTuber [27], HOLLOW [28], 3V [29], CAVER [30], and CHUNNEL [31]. In these methods, protein structures are projected onto a 3D grid. POCKET scans void grid voxels along the x-, y- and z-axes for protein–solvent–protein (PSP) events and assigns void grid voxels which belong to two PSP events as pocket grid voxels (Figure 2.1). dxTuber converts a set of protein conformations from a molecular dynamics simulation into residence probabilities of protein and solvent in grid voxels and then identifies the pockets in a similar manner as POCKET. LIGSITE searches four more diagonal directions to reduce the orientation dependency of the results. HOLLOW and 3V both use two rolling probes to define the internal volume of the protein whereby the larger probe is used for the boundaries of pores and the smaller probe determines the resolution (Figure 2.2). The program CAVER was specifically developed for finding channels inside protein structures. In CAVER, each void grid voxel is assigned a cost value. Dijkstra's algorithm [32] is then applied to find the lowest cost path among void grid voxels and identifies this path as a channel (Figure 2.3). CHUNNEL adopts a similar approach as CAVER but automatically specifies the starting point inside the pore region from a topological analysis of the protein surface.

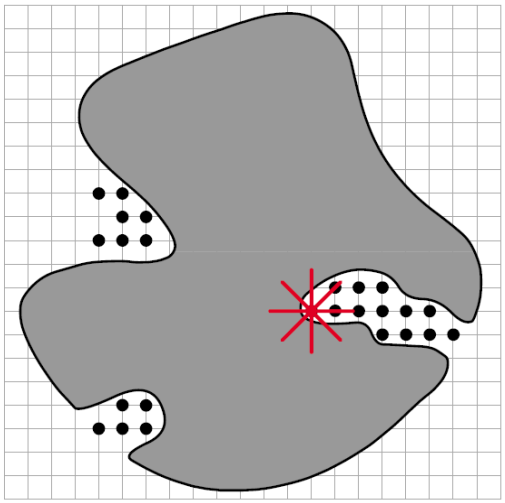

Figure 2.1 Pore identification strategies used in POCKET, LIGSITE, and dxTuber. The protein (grey area) is projected onto a grid. Red lines indicate the scanning directions for protein-solvent-protein events. Black spots locate the volume identified as pore region.(This figure was adapted from [33]).

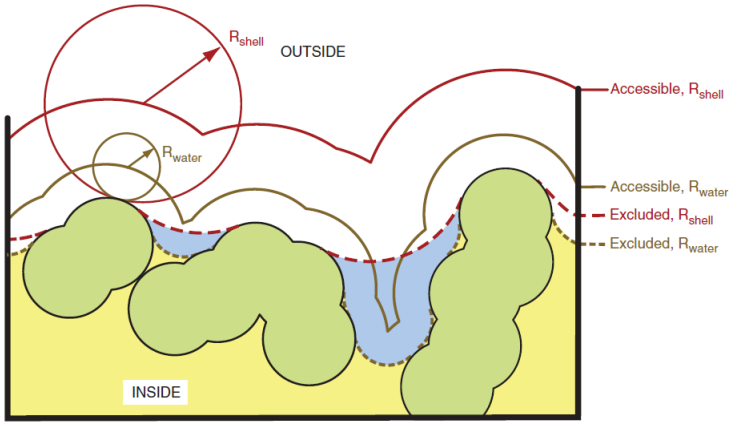

Figure 2.2 Pockets are identified by rolling probes in HOLLOW and 3V. The van der Waals volume of the protein is shown as green area. Red and brown contours show the accessible surfaces generated by two sizes of probes while the dashed lines depict the molecular surfaces according to the two probes. The blue volume enclosed by two molecular surfaces is determined as pore volume. (This figure was adapted from [29]).

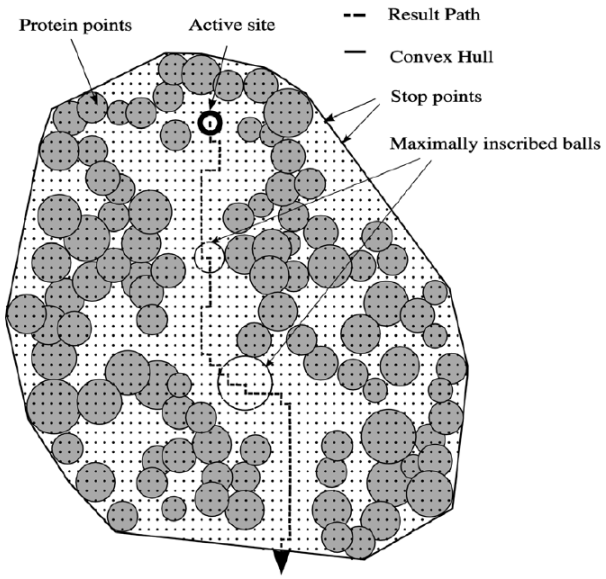

Figure 2.3 Channel search algorithm adopted by CAVER and CHUNNEL. Grey spheres are protein atoms with van der Waals radii. The convex hull encompasses all protein atoms and defines the stopping condition of the search algorithm. Each void grid voxel is assigned a cost value according to the size of the maximally inscribed ball (open circles). The lowest cost path (dashed line) is found by Dijkstra's algorithm as the channel in the protein. (This figure was adapted from [30]).

The second class of methods utilizes Voronoi Diagrams (or its dual graph Delaunay triangulation) such as MOLE [34], MolAxis [35], and CAST [36]. MOLE is an improved version of CAVER that is more efficient and precise than CAVER and uses the same algorithm for finding lowest cost paths as channels in the protein (Figure 2.4). The MolAxis algorithm is quite similar to MOLE except that it approximates the different-sized atoms by clusters of unit spheres. CAST partitions a protein into tetrahedra to form a Delaunay triangulation. The vertices of tetrahedra are protein atoms. Void spaces in the tetrahedra are then grouped by the discrete flow algorithm to form pockets, cavities, or channels.

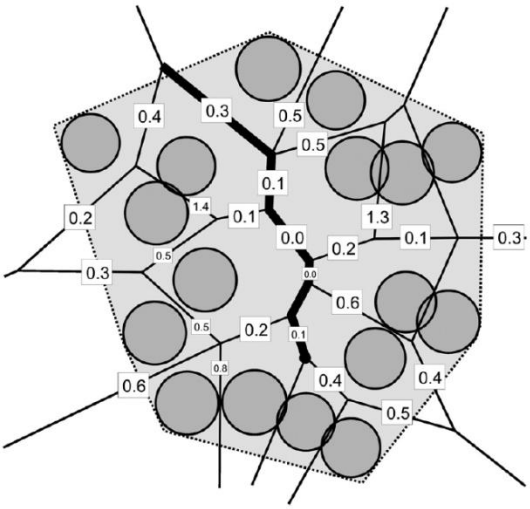

Figure 2.4 Channel search algorithm adopted by MOLE and MolAxis. Protein atoms and the convex hull have the same definitions as those in Figure 2.3. A Voronoi diagram (lines) is applied to find the central lines (or facets in 3D) between atoms. The optimal route (thick line) is also computed by Dijkstra's algorithm as the channel in the protein. (This figure was adapted from [34]).

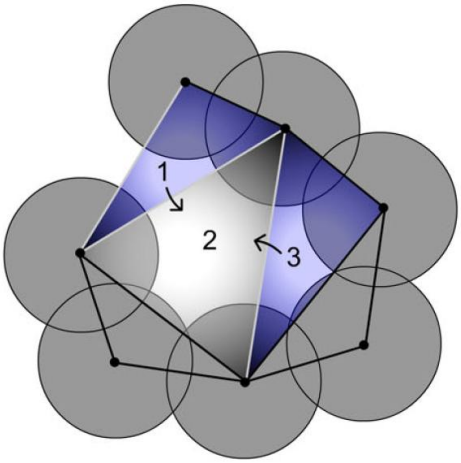

Figure 2.5 Delaunay triangulation method used by CAST to determine pore volume. Delaunay triangles are made according to protein atoms. The discrete flow algorithm is then used to group triangles to a continuous pore volume. The flows (arrows) start from blunt triangles and end in an acute triangle.(This figure was adapted from [33]).

The third class contains sphere-filling methods such as PASS [37] and SURFNET [38]. The basic idea of PASS is based on the geometric relationship of a sphere being placed on top of three other given spheres. The protein is carpeted with spheres layer by layer. The carpeting spheres are then filtered according to their burial degree by protein atoms. A cluster of carpeting spheres represents a pocket (Figure 2.6). PASS is designed specifically for detecting pockets.

SURFNET fits suitable spheres between all atom pairs of a protein. The overlapping spheres depict the pockets, cavities, or channels (Figure 2.7).

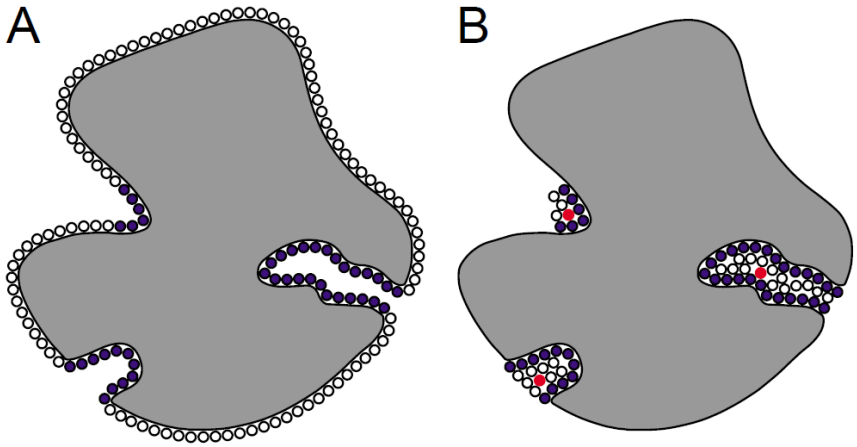

Figure 2.6 Pocket detection strategy used by PASS. (A) The protein surface is carpeted with spheres layer by layer. (B) The carpeting spheres are then filtered according to their burial degree by protein atoms. Each cluster of carpeting spheres represents a pocket. (This figure was adapted from [33]).

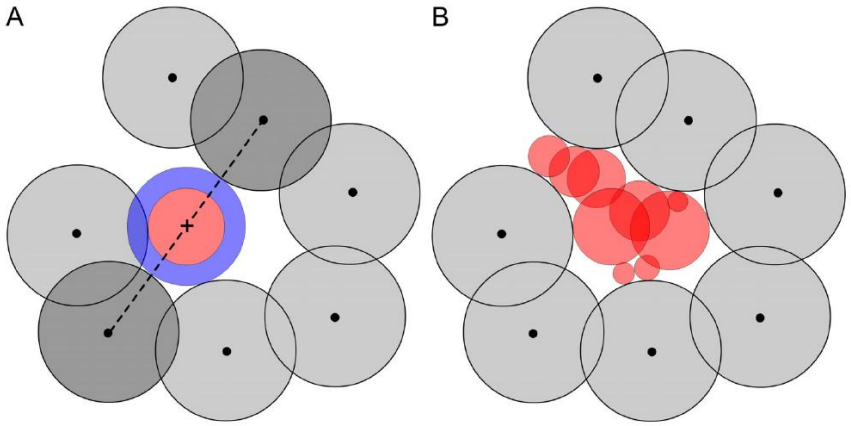

Figure 2.7 Sphere-filling method used by SURFNET. (A) A tangent sphere (blue) is fitted in between a pair of atoms and its size is then adjusted to avoid clashes with other atoms (red). (B) The overlapping spheres can be grouped and depict the pockets, cavities, or channels. (This figure was adapted from citation [33]).

Methods of the last class are slice and optimization methods such as HOLE [39] and PoreWalker [40]. HOLE splits a protein into slices along the start vector defined by the user. Monte Carlo optimization is used to find the largest sphere that can be squeezed into the gap between protein atoms of each slice (Figure 2.8). HOLE generates a channel tunneling through the protein. PoreWalker also cuts the protein into slices, but the program is specific for transmembrane proteins and takes advantage of the special arrangements of secondary structures when defining the initial pore axis. The pore axis is optimized in an iterative way to find the channel lined by most residues (Figure 2.9).

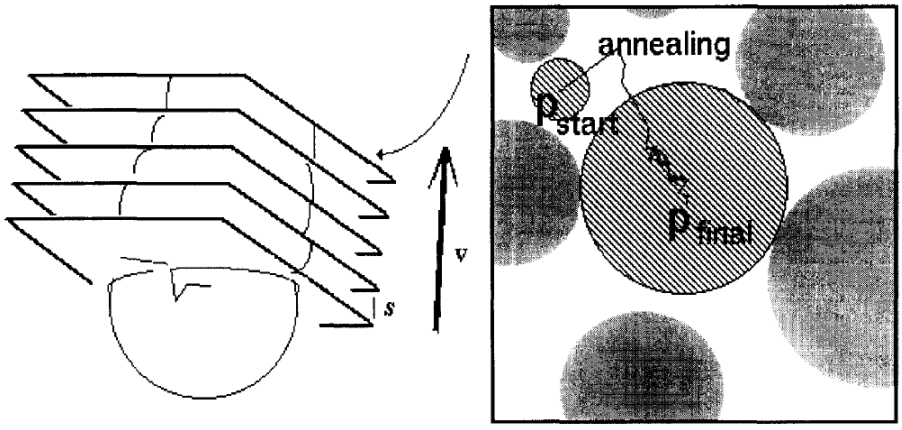

Figure 2.8 Tunneling method adopted by HOLE. The protein is scanned plane by plane for void space inside. In each plane, Monte Carlo optimization is used to find the largest sphere that can be squeezed into the gap between protein atoms and that determines the radius of the channel at this position. (This figure was adapted from [39]).

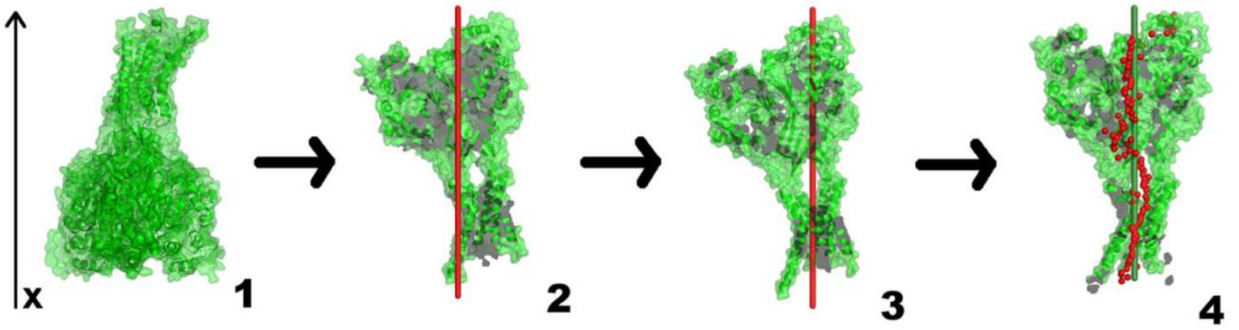

Figure 2.9 Channel depiction algorithm of PoreWalker. The channel orientation of a transmembrane protein is first determined as the averaged orientation of its secondary structures and then iteratively adjusted by pore-lining residues through the channel. (This figure was adapted from [40]).

2.2 Support vector machine (SVM) ([41] and the references therein)

The support vector machine (SVM) method is a supervised machine learning algorithm for the binary classification problem. Supervised learning refers to finding or estimating a functional relationship between a large number of examples with input/output pairs. The binary classification deals with a learning problem whose target outputs constitute only two classes (i.e. yes or no, positive or negative).

The basic model of a support vector machine is the maximal margin classifier which works only for the linearly separable data. The linear classification of this model is performed by a linear function $f(x)$ as Eq. 2.1 where $\mathbf{x}$ is an input vector with n dimensions, $\mathbf{w}$ and b are parameters that determine the function. An input is assigned to the positive class if $f(x) > 0$ or to the negative class if $f(x) < 0$.

$$f(x) = \langle \mathbf{w} \cdot \mathbf{x} \rangle + b = \sum_{i=1}^{n} w_i x_i + b \qquad (2.1)$$

The separating function ($f(x) = 0$) is the so-called hyperplane. A hyperplane is a subspace of dimension n-1 which separate data points into two classes by dividing the n dimensional space into two half spaces. The strategy to choose the hyperplane of maximal margin classifier is to maximize the minimal margin between data points (both positives and negatives) and the hyperplane. The margin can be computed by the geometric distance of two equations (Eqs. 2.2a and 2.2b) that are derived from the hyperplane where $\mathbf{x}^+$ is a positive point which gives a minimal margin between positive points and the hyperplane and $\mathbf{x}^-$ gives that for negative points. Eq. 2.3 shows that the geometric margin γ is equal to $1/\| \mathbf{w} \|^2$.

$$f(x) = \langle \mathbf{w} \cdot \mathbf{x}^+ \rangle + b = 1 \tag{2.2a}$$

$$f(x) = \langle \mathbf{w} \cdot \mathbf{x}^- \rangle + b = -1 \tag{2.2b}$$

$$\gamma = \frac{1}{2}\left(\langle \frac{\mathbf{w}}{\|\mathbf{w}\|^2} \cdot \mathbf{x}^+ \rangle - \langle \frac{\mathbf{w}}{\|\mathbf{w}\|^2} \cdot \mathbf{x}^- \rangle \right) = \frac{1}{\|\mathbf{w}\|^2} \tag{2.3}$$

Hence, the classification problem can be formulated as an optimization problem (Eq. 2.4). For a given set S of $\mathbf{x}/\mathrm{y}$ pairs, the problem is to maximize the separating margin γ (which is equivalent to minimizing $\| \mathbf{w} \|^2$) within the constraints of that both positive and negative points will be positioned on their proper sides and no point is located in the margin space.

$$S = \left((\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_l, y_l) \right)$$

$$\text{Minimize } \|\mathbf{w}\|^2 \tag{2.4}$$

$$\text{Subject to } y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) \geq 1, i = 1, \cdots, l$$

This constrained optimization problem can be transformed into a Lagrangian (Eq. 2.5) which introduces the Lagrange multipliers $\boldsymbol{\alpha}$ to combine the original optimization problem and its constraints. The stationary points of the Lagrangian can be used to detect solutions. Actually, the Lagrange multiplier method allows only equality constraints. It is the Karush-Kuhn-Tucker necessary conditions that generalize the Lagrange multiplier methods to allow inequality.

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{l} \alpha_i [y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle) - 1], \alpha_i \geq 0 \tag{2.5}$$

Eq. 2.5 is called primal Lagrangian and its solutions can be used to obtain the dual form of the Lagrangian. The solution of the primal Lagrangian is found by differentiating with respect to $\mathbf{w}$ and b as shown in Eqs. 2.6 and 2.7.

$$\begin{aligned} \frac{L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial \mathbf{w}} &= \mathbf{w} - \sum_{i=1}^{l} y_i \alpha_i \mathbf{x}_i = 0, \\ \mathbf{w} &= \sum_{i=1}^{l} y_i \alpha_i \mathbf{x}_i \end{aligned} \tag{2.6}$$

$$\begin{aligned} \frac{L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial b} &= \sum_{i=1}^{l} y_i \alpha_i = 0, \\ 0 &= \sum_{i=1}^{l} y_i \alpha_i \end{aligned} \tag{2.7}$$

To substitute solutions into the primal Lagrangian, we obtain Eq. 2.8 and the optimization problem is reformulated to the dual Lagrangian as shown in Eq. 2.9.

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \tag{2.8}$$

$$S = \big((\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_l, y_l)\big)$$

$$\text{Maximize } W(\boldsymbol{\alpha}) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \tag{2.9}$$

$$\text{Subject to } \sum_{i=1}^{l} y_i \alpha_i = 0,\ \alpha_i \geq 0,\ i = 1, \cdots, l$$

The optimal solutions $\boldsymbol{\alpha}^*$, $\mathbf{w}^*$ and b^* must satisfy the Karush-Kuhn-Tucker conditions (Eq. 2.10). This necessary condition also implies that α_i^* are nonzero only when the $\mathbf{x}_i/y_i$ pairs satisfy either Eq. 2.2a or Eq. 2.2b. All other α_i^* are zero. The weight vector is represented as $\mathbf{w}^* = \sum y_i \alpha_i^* \mathbf{x}_i$. Thus, the points $\mathbf{x}_i$ involved in the weight vector are so-called support vectors.

$$\alpha_i^* [y_i (\langle \mathbf{w}^* \cdot \mathbf{x}_i \rangle + b^*) - 1] = 0, i = 1, \cdots, l \tag{2.10}$$

To find the $\boldsymbol{\alpha}^*$, the Sequential Minimal Optimization (SMO) algorithm was developed. The idea is to optimize only a subset of $\boldsymbol{\alpha}$ at each iteration. The condition $\sum y_i \alpha_i = 0$ in Eq. 2.9 implies that the minimal number of $\boldsymbol{\alpha}$ that can be optimized is 2. Thus, at each iteration of SMO, two elements (α_i and α_j) are chosen to be optimized while the others are fixed. To speed up the convergence, SMO heuristically chooses the two elements of $\boldsymbol{\alpha}$. The procedure contains two separate criteria to choose the first and the second points, respectively. The first point is chosen by looking through the training set for the one violating the Karush-Kuhn-Tucker conditions. The second point is chosen to cause a large change on the pair of corresponding α. After finding $\boldsymbol{\alpha}^*$, the value of b can be obtained by substituting support vectors and $\boldsymbol{\alpha}^*$ into Eq. 2.2.

To deal with real data that is rarely linearly separable, two approaches were developed to expand the capability of SVMs. One is the "soft margin optimization" to tolerate the noise or outliers. The other one is "kernel-induced feature space" which is used to map the data into a higher dimensional space. The linear learning SVM mentioned above has a higher probability to separate the data if the data are properly mapped in the new feature space. The following paragraphs will introduce the ideas behind these. In practice, these two approaches are often combined and implemented in SVM.

To avoid the case that the separation of a data set is ruined by a few points (noise), "soft margin optimization" was proposed. Here, so-called slack variables ξ were introduced to tolerate the violation of constraints or even tolerate a few misclassifications. Here, we briefly describe one of the soft margin algorithms named 2-norm soft margin. The target of optimization and the constraints in Eq. 2.4 are extended to Eq. 2.11 while C is the penalty and this parameter has to be tuned during the SVM training.

$$\begin{gathered} S = \left((\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_l, y_l)\right) \\ \text{Minimize } \|\mathbf{w}\|^2 + C\sum_{i=1}^{l}\xi_i^2 \\ \text{Subject to } \; y_i(\langle \mathbf{w}\cdot\mathbf{x}_i\rangle + b) \geq 1 - \xi_i, i = 1, \cdots, l \\ \xi_i \geq 0, i = 1, \cdots, l \end{gathered} \tag{2.11}$$

The new primal Lagrangian including slack variables is given in Eq. 2.12.

$$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{2}\sum_{i=1}^{l}\xi_i^2 - \sum_{i=1}^{l}\alpha_i[y_i(\langle \mathbf{w}\cdot\mathbf{x}_i\rangle) - 1 + \xi_i], \alpha_i \geq 0 \tag{2.12}$$

The solutions of the primal Lagrangian are

$$\frac{L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha})}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{l} y_i \alpha_i \mathbf{x}_i = 0, \tag{2.13}$$

$$\frac{L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha})}{\partial \xi} = C\boldsymbol{\xi} - \boldsymbol{\alpha} = \mathbf{0}, \tag{2.14}$$

$$\frac{L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha})}{\partial b} = \sum_{i=1}^{l} y_i \alpha_i = 0, \tag{2.15}$$

Resubstituting the relations of Eqs. 2.13, 2.14 and 2.15 into Eq. 2.12 gives the dual form of the Lagrangian (Eq. 2.16).

$$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle - \frac{1}{2C} \langle \boldsymbol{\alpha} \cdot \boldsymbol{\alpha} \rangle \tag{2.16}$$

Thus, the optimization problem can be reformulated as Eq. 2.17 and the corresponding Karush-Kuhn-Tucker conditions are given in Eq. 2.18.

$$S = \big((\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_l, y_l)\big)$$

$$\text{Maximize } W(\boldsymbol{\alpha}) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle - \frac{1}{2C} \langle \boldsymbol{\alpha} \cdot \boldsymbol{\alpha} \rangle \tag{2.17}$$

$$\text{Subject to } \sum_{i=1}^{l} y_i \alpha_i = 0,\ \alpha_i \geq 0,\ i = 1, \cdots, l$$

$$\alpha_i^* [y_i (\langle \mathbf{w}^* \cdot \mathbf{x}_i \rangle + b^*) - 1 + \xi_i] = 0, i = 1, \cdots, l \tag{2.18}$$

The other main approach proposed to improve the performance of SVM is "kernel-induced feature space". It is used to map the original data into a higher dimensional feature space and make the data linearly separable in this new feature space. The mapping of a data set is denoted as Eq. 2.19.

$$\mathbf{x} = (x_1, \dots x_n) \mapsto \boldsymbol{\phi}(\mathbf{x}) = (\phi_1(\mathbf{x}), \dots, \phi_N(\mathbf{x})) \tag{2.19}$$

The mapping can be performed implicitly because of the dual representation of Lagrangian. The kernel function is defined as the inner product of two input vectors in the mapped feature space (Eq. 2.20). It is used to replace the original inner product as shown in Eqs. 2.9 and 2.17. With the relation of Eq. 2.6 (or Eq. 2.13), the hyperplane in the new feature space is changed into Eq. 2.21.

$$K(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle \tag{2.20}$$

$$f(\mathbf{x}) = \sum_{i=1}^{n} w_i \phi_i(\mathbf{x}) + b = \sum_{i=1}^{n} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b \tag{2.21}$$

The kernel used in our study (Chapter 4) is the radial basis function kernel (RBF kernel) which is defined in Eq. 2.22. g is also an adjustable parameter during the training procedure. Besides, the feature space of this kernel has an infinite number of dimensions because the exponential function can be decomposed into a polynomial with an infinite number of terms and an infinitely large degree (e.g. Taylor expansion).

$$K(\mathbf{x}, \mathbf{z}) = \exp\left(-g\|\mathbf{x} - \mathbf{z}\|^2\right) \tag{2.22}$$

2.3 Brownian motion and Langevin equation ([42] and the references therein)

The first description of Brownian motion was given by the botanist Robert Brown in 1827. He found under a microscope that the pollen grains in suspended aqueous solution were in rapid irregular motion. After Brown's finding, several observations and experiments were done and published to explain Brownian motion. Their findings can be summarized in the following way: 1. The motion never ceases; 2. Temperature has an effect of the particle motion; 3. The motion is irregular and its trajectory has no tangent; 4. The size of a particle affects the velocity of motion. 5. Two particles move independently.

In 1905, Albert Einstein explained and formulated Brownain motion by means of probability. This was the first quantitative formulation of Brownian motion. He considered the particle motion in terms of diffusion. Let us consider one dimensional particle motion as an example. $P(x,t)$ and $p(\Delta, \tau)$ are probability density functions. The probability of a particle in a unit volume between x and $x+dx$ at time t is denoted as $P(x,t)$. $p(\Delta, \tau)$ stands for the probability of the net flow (particles from neighboring element x' moving into x or the reverse way) within a small time period τ where Δ is $x'-x$. So after time τ, $P(x,t+\tau)$ can be written down as Eq. 2.23.

$$P(x, t+\tau) = \int_{-\infty}^{\infty} P(x+\Delta, t) p(\Delta, \tau) d\Delta \tag{2.23}$$

When expanding $P(x,t+\tau)$ in power of t, expanding $P(x+\Delta,t)$ in power of Δ, and neglecting higher order terms of two expansions, Eq. 2.23 can be simplified to Eq. 2.24.

$$\tau \frac{\partial P(x,t)}{\partial t} = \frac{1}{2} \frac{\partial^2 P(x,t)}{\partial x^2} \overline{\Delta^2} \tag{2.24}$$

Since the translational diffusion coefficient in one dimension is defined as Eq. 2.25, Eq.2.24 can be written as Eq. 2.26.

$$D = \frac{1}{2\tau} \overline{\Delta^2} \tag{2.25}$$

$$\frac{\partial P(x,t)}{\partial t} = D \frac{\partial^2 P(x,t)}{\partial x^2} \tag{2.26}$$

The solution of Eq. 2.26 is given by Eq. 2.27.

$$P(x,t) = \frac{1}{\sqrt{4\pi D t}} \exp\left(\frac{-x^2}{4Dt}\right) \tag{2.27}$$

By calculating the second moment of Eq. 2.27, we obtained the mean square displacement for a particle motion in one dimension (x direction here as Eq. 2.28).

$$\overline{x^2} = 2Dt \tag{2.28}$$

In 1908, Paul Langevin formulated Brownian motion in a different way. He considered that a single particle in the solution feels two forces exerted by solvent molecules, a frictional force and a fluctuating force. The frictional force represents the dynamical friction experienced by the particle. The fluctuating force is due to random kicks by solvent molecules. Based on Newton's second law of motion, Langevin wrote down his equation of motion as Eq. 2.29 where

ζ is the coefficient of friction and $f_r(t)$ stands for the fluctuating force. Eq. 2.29 is now called Langevin Equation.

$$m\frac{d^2x(t)}{dt^2} = -\zeta\frac{dx(t)}{dt} + f_r(t) \tag{2.29}$$

Although $f_r(t)$ means random and irregular force, it is still governed by two assumptions (Eqs. 2.30 and 2.31 where δ is the Dirac delta function). Eq. 2.30 means that the ensemble average of $f_r(t)$ should be zero and Eq. 2.31 shows that $f_r(t)$ is independent between different time points.

$$\overline{f_r(t)} = 0 \tag{2.30}$$

$$\overline{f_r(t)f_r(t')} = 2\zeta kT\delta(t-t') \tag{2.31}$$

To derive Eq. 2.29 for the mean square displacement, Langevin firstly multiplied Eq. 2.29 by $x(t)$. The following equations (Eqs. 2.32 to 2.37) show the detail of derivation.

$$m\frac{d^2x(t)}{dt^2}x(t) = -\zeta\frac{dx(t)}{dt}x(t) + f_r(t)x(t) \tag{2.32}$$

Eq. 2.32 can be rewritten as Eq.2.33 by substituting the terms of derivative by their equivalent forms.

$$\frac{m}{2}\frac{d}{dt}\left(\frac{dx^2(t)}{dt}\right) - m\left(\frac{dx(t)}{dt}\right)^2 = -\frac{\zeta}{2}\frac{dx^2(t)}{dt} + f_r(t)x(t) \tag{2.33}$$

For the macroscopic view, Eq. 2.33 is changed into the averaged form (Eq. 2.34).

$$\frac{m}{2}\frac{d}{dt}\left(\frac{d\overline{x^2(t)}}{dt}\right)-m\left(\frac{d\overline{x(t)}}{dt}\right)^2=-\frac{\zeta}{2}\frac{d\overline{x^2(t)}}{dt}+\overline{f_r(t)x(t)} \tag{2.34}$$

The right most term in Eq. 2.34 vanishes because of the assumption of Eq. 2.30. In addition, according to the kinetic theory in one dimension (Eq. 2.35), Eq. 2.34 becomes Eq. 2.36.

$$\frac{1}{2}m\left(\frac{d\overline{x(t)}}{dt}\right)^2=\frac{1}{2}kT \tag{2.35}$$

$$\frac{m}{2}\frac{d}{dt}\left(\frac{d\overline{x^2(t)}}{dt}\right)+\frac{\zeta}{2}\frac{d\overline{x^2(t)}}{dt}=kT \tag{2.36}$$

The solution of Eq.2.36 is given by

$$\overline{x^2(t)}=\frac{2kT}{m\zeta}t \tag{2.37}$$

Eq. 2.37 is consistent with Eq. 2.28 and the Nernst-Einstein Equation (Eq. 2.38) can be obtained from these two equations.

$$D=\frac{kT}{m\zeta} \tag{2.38}$$

The two expressions of Brownian motion (probability by Einstein and equation of motion by Langevin) derived an equivalent mean square displacement. However, the Langevin equation refers to the motion of a single particle that is based on Newton's law of motion and obeys the law of conservation of momentum. This makes the Langevin equation easier to be implemented

in dynamics simulations of particle motions in the form of so-called Brownian Dynamics. We can directly introduce external forces or interactions between particles into the Langevin equation as described in Section 5.2. In Brownain Dynamics, solvent molecules are considered as a mean field with properties of friction, fluctuation and other interactions (ex. dielectric and hydrodynamics). That is why we have to use a larger time step in Brownain Dynamics simulations. In most cases, it is more efficient than all atom Molecular Dynamics simulations which consider the atomic details of the whole system.

2.4 Molecular docking and AutoDock [43-45]

Due to the great improvements of computational hardware and algorithms, computer aided drug design has become a crucial approach during the discovery or design of new drugs. Molecular docking is one of the best developed and most implemented methods in the field of computer aided drug design. It is used to compute the proper conformation and orientation between a ligand and its receptor. It is also used to estimate the binding affinity or to evaluate the fitness of this binding. In this section, we focus on small molecule-protein docking and take the popular docking software “AutoDock” as an example to introduce the basic components and the ideas of molecular docking. A case study is presented in Chapter 6.

The two main components of a docking software are the scoring function and the search algorithm. Mimicking the atomistic force field used in molecular dynamics simulations, the scoring function of AutoDock contains 5 terms (Eq. 2.39). The developers adopted the Lennard-Jones potential as van der Waals interaction between atom pairs; a 12-10 potential was used

for hydrogen bonding where $E(t)$ is a function of the angle t between the hydrogen bond forming atoms; and a Coulombic term for electrostatic interactions. The fourth term is the desolvation energy where S stands for the solvent accessible surface area of the ligand and V stands for the volume surrounded by protein atoms (or in the opposite way). The last term is the estimation of entropy loss upon ligand binding that is proportional to the number of rotatable bonds of the ligand. Ws are the weights of each term. The developers performed an empirical fit of Eq. 2.39 against the experimental binding affinities from protein-ligand complexes with known structures to obtain the optimal weights (30 complexes for AutoDock3 and 188 complexes for AutoDock4).

$$\Delta G = W_{vdw}\sum_{i,j}\left(\frac{A_{ij}}{r_{ij}^{12}}-\frac{B_{ij}}{r_{ij}^{6}}\right) + W_{hbond}\sum_{i,j}E(t)\left(\frac{C_{ij}}{r_{ij}^{12}}-\frac{D_{ij}}{r_{ij}^{10}}\right) + W_{elec}\sum_{i,j}\frac{q_i q_j}{\varepsilon(r_{ij})r_{ij}} + W_{sol}\sum_{i,j}\left(S_i V_j + S_j V_i\right)e^{(-r_{ij}^2/2\sigma^2)} + W_{conf}N_{tors} \tag{2.39}$$

Besides force field based and empirically fitted scoring functions, there are also groups of scoring functions developed by different techniques or concepts such as knowledge-based scoring functions, machine learning-based scoring functions, and consensus scoring functions. The idea of knowledge-based scoring functions is to observe the frequency of occurrence of contacting atom pairs in protein-ligand complexes and then convert this frequency into a Boltzmann weighted potential by so-called Boltzmann-inversion. Machine learning-based scoring functions take the advantage of the nonlinear mapping capability of machine learning algorithms to properly map the interactions to docking scores. However, there is unfortunately

no single perfect scoring function or at least it has not been found sofar. This led to the development of consensus scoring methods. This approach integrates several existing scoring functions by a voting scheme and thus provides a good compromise of different scoring functions. [46-48]

The search algorithm used in AutoDock is the Lamarkian genetic algorithm. A genetic algorithm (GA) is an optimization algorithm that mimics the process of population evolution under selection. The feature vector represents a chromosome of an individual. Within a generation, each chromosome may suffer from mutations or crossover with another chromosome to generate variants. The purpose of these two steps is to enhance the diversity of the population. Then, each chromosome is evaluated by a fitness function (in the docking problem, the scoring function plays the role of the fitness function) as selection under the external pressure. During this step, the individuals with better fitness will proliferate but ones with worse fitness will be obsolete. After several iterations of mutation, crossover and selection, the population will typically converge to a state with higher fitness than the original one. The optimal solution seems to be inside the population in the converged state. The Lamarkian genetic algorithm is an extension of conventional genetic algorithms to improve the performance of optimization by combining the genetic algorithm and the local optimization algorithm (Solis-Wet algorithm [49] is adopted in AutoDock). The local optimization algorithm is adaptive and adjusts the step size according to the current condition. Besides GA, several other optimization algorithms have been implemented in docking calculations such as simulated annealing, Monte Carlo simulation, particle swarm optimization [50], ant colony optimization [51] and so on.

Furthermore, AutoDock speeds up the docking calculation by precomputing the interaction energies with respect to the protein at each grid point within the user defined grid box by different probes of atom types. When evaluating the energies of different ligand conformations, the binding free energies can be quickly obtained by summing up the precomputed values of component atoms.

Instead of the flexible-ligand-rigid-receptor docking scheme mentioned above, the scientists in the field were recently more concerned about docking to flexible receptors. AutoDock4 allows a partial flexibility of the protein. Side chains of pocket lining residues selected by users can be treated as flexible ligands during docking simulations in AutoDock4. Other software tools such as DOCK6 [52] and GOLD [53] use a rotamer library to generate different conformations of the binding pocket to account for receptor flexibility. One further approach is to generate conformational ensemble of the receptor by molecular dynamics simulations or by software such as CONCOORD and tCONCOORD [54] before performing docking. This is the so-called relaxed complex scheme. [55] The advantage of the method is to dock the ligand to the rare event of pocket opening and the receptor is fully flexible. However, the relaxed complex scheme is much more computationally demand than the other two methods.

3 Identifying continuous pores in protein structures with PROPORES by computational repositioning of gating residues

This work had been published as "Lee PH, Helms V: **Identifying continuous pores in protein structures with PROPORES by computational repositioning of gating residues**. *Proteins* 2012, **80**(2):421-432". My own contribution of this work was to design the algorithms of PoreID, PoreTrace and GateOpen, to evaluate and compare the performance of our tools to other existing tools, to program PROPORES tool package and to write the manuscript for publication.

3.1 Background

For functional reasons, proteins often contain concavities such as pockets, cavities, or channels. Some biological reactions (ex. functions of GPCR receptors, ABC transporters, translocons, K^+ channel, aquaporins and AcrB proteins mentioned in the Section 1.2) rely on the geometric complementarity between proteins and small molecules, and also on the physicochemical properties of the residues that line pockets, cavities, or channels. A considerable number of computational methods have been developed, which identify the concavities of proteins and describe their properties. The review of these works is presented in the Section 2.1.

3.2 Motivation

Among all the methods mentioned in Section 2.1, grid-based method such as POCKET, LIGSITE, and dxTube sometimes give results with an undesirable orientation dependency. Some

of the methods are not automatic and users have to define the starting point (CAVER, MOLE, and MolAxis) or starting vector (HOLE) of the search. Some methods are specific for detecting pockets (PASS) or channels (CAVER, CHUNNEL, MOLE, MolAxis, HOLE, and PoreWalker). Thus, we developed a new toolkit PROPORES (PROtein PORE identification toolS) which includes the three programs PoreID for pore identification, PoreTrace for pore axes determination and Gate-Open for opening the gate between neighboring pores. PROPORES is a grid-based method that avoids orientation dependency of the results. It targets all kinds of pores (pockets, cavities, and channels) and is automatic so that only the PDB file of the target protein has to be specified by the user. The design of the algorithm is presented in the next section.

3.3 Materials and Methods

3.3.1 PoreID: pore identification

The idea of PoreID is based on the POCKET and LIGSITE algorithms. Both methods employ a regular Cartesian grid and define all those void grid voxels as pore grid voxels where the grid voxel belongs to two perpendicular PSP events (Figure 2.1). The POCKET algorithm scans void grid voxels along the three Cartesian axes. As an extension to POCKET, the LIGSITE algorithm also searches along the four cubic diagonal directions to improve the pore identification along the diagonal direction (45°). Most grid-based methods such as POCKET and LIGSITE show an orientation dependency of their results so that, in some extreme cases, different pockets or pores may be found for different rotational orientations of the protein. To avoid this problem, we adopted an approach inspired by the pore identification method SURFNET.

Here, all atom pairs of a protein (or protein complex) are used as framework for defining PSP events. Let us imagine that each atom pair defines the endpoints of the central axis of a cylinder. The radius of each cylinder is set equal to the van der Waals (vdW) radius of the smaller atom. Although the space inside the cylinder may satisfy the conditions for a PSP event, the enclosed volumes do not have to correspond to pores. Two cylinders (Figure 3.1), however, that cross at a roughly perpendicular angle certainly determine an enclosed volume which is defined as pore region by the POCKET algorithm. Due to the way of construction, this volume is constant and independent of the protein orientation.

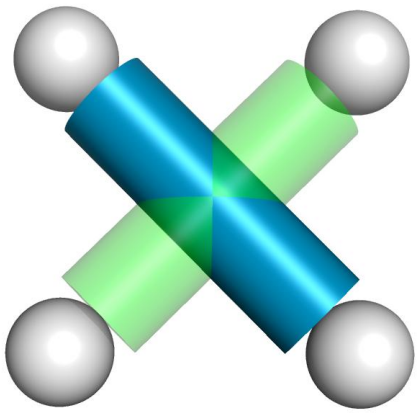

Figure 3.1 Cylindrical scheme for pore identification. Gray spheres represent protein atoms. Although the volume enclosed in either the blue or the green cylinders satisfies the definition of a PSP event, only the volume crossed by both cylinders is identified as pore region here.

The implementation of our pore identification method includes the following steps:

a) Fit all protein atoms into the meshes of a 3D grid of suitable dimensions where each protein atom is represented by a set of grid voxels within a sphere of radius equal to its atomic vdW radius. Void grid voxels are those voxels that do not have any neighboring protein atoms within their vdW radius distances.

b) Scan PSP events along the x-, y- and z-axes for all void grid voxels and label the grid voxels by the number of PSP events encountered. Void grid voxels with two or more PSP events are classified as pore grid voxels.

c) Determine all grid voxels which are crossed by one or more of the vectors connecting the atom pairs (these grid voxels are termed ‘‘tracing grid voxels’’). Discard those vectors whose tracing grid voxels overlap with grid voxels occupied by other atoms. Among the remaining vectors, identify the void grid voxels in each cylinder connecting the respective atom pairs. Label each void grid voxel by the indices of the vectors that enclose the void grid voxel in their cylinders.

d) Scan all void grid voxels which are not identified as pore grid voxels in Step (b). Check the perpendicularity of vector pairs of each void grid voxel. In this step, vector pairs that enclose an angle between 85° and 95° are considered as being perpendicular. Void grid voxels with perpendicular vectors are also classified as pore grid voxels.

e) Trim shallow pore grid voxels on the protein surface. Cluster neighboring pore grid voxels to form several connected, disjoint pores. Analyze pore volumes, pore-lining residues and possible pore axes. The algorithm for determining these pore axes is described in the next section.

The first two Steps (a) and (b) are similar to the POCKET algorithm. Most of the pore grid voxels buried inside the protein, and some on the surface are detected by these two steps. Step (c) serves to discard redundant vectors of atom pairs. For this, we introduced the concept of crossing cylinders. As shown in Figure 3.2, Vector v2 is redundant for the checking of PSP events because v2 overlaps with v1. According to our experience, less than one percent of all vectors

remain after Step (c). In Step (d), we only scan void grid voxels which are not identified in Step (b), and the vector set is non-redundant. Because of the first three steps, the computational effort for Step (d) is substantially reduced.

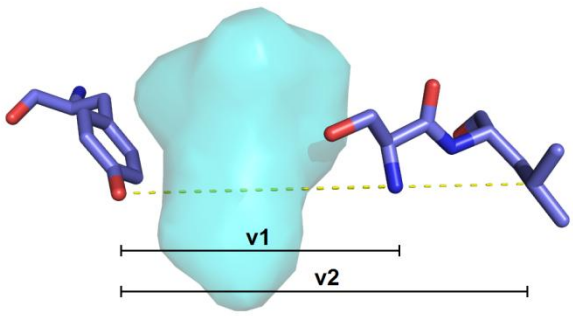

Figure 3.2 Redundant vector in pore identification. The transparent surface shows a pore in the protein. The dashed line represents vectors between atom pairs. Vector v2 is redundant for pore identification because it overlaps with v1 whose atom pair directly lines the pore.

3.3.2 PoreTrace: pore axes determination

An identified pore consists of a set of pore grid voxels. These pore grid voxels can be classified into three categories according to their local environments, namely “grid voxels exposed to the surface”, “grid voxels in contact with protein atoms”, and “grid voxels surrounded by other pore grid voxels only”. Obviously, the endpoints of any meaningful pore axis should be pore grid voxels of the first two cases. Besides, surface patches which are composed of “grid voxels exposed to the surface” must be the entrances of the pores pointing toward or into the protein if the pores are open. So we cluster all exposed grid voxels to form surface patches which resemble lids on holes or pots. Further, there exist three cases of pores depending on the number of exposed surface patches involved. These are “pores containing two or more exposed surface patches”, “pores containing only one exposed surface patch”, and

"pores without exposed surface patch". The implementations of pore axes determination are different for these three cases due to the determination of the endpoints of the pore axis. Our strategy for determining the pore axes is similar to the one used by the program CAVER that applies Dijkstra's algorithm to search the lowest cost paths as pore axes. Each grid voxel is assigned a value of R_{sphere} by the KD-tree algorithm [56], where R_{sphere} is the radius of the largest sphere that can be fitted at the location of the pore grid voxel without overlapping with a protein atom. The details of the implementations are described as follows:

1) For each pore with two or more exposed surface patches, we define the grid voxels with largest R_{sphere} on each exposed surface patch as the endpoints of the pore axes. Note that N exposed surface patches (endpoints) determine C(N,2) pore axes (2-combinations from the set of N elements). Lowest cost paths are searched by Dijkstra's algorithm between any pair of these endpoints. The cost of each grid voxel is $1/R_{sphere}$.
2) For each pore containing only a single exposed surface patch, Dijkstra's algorithm has to be performed twice. First, we define the endpoint on the exposed surface as in Case (1). The cost of each grid voxel is set as $D_{COM}/(R_{sphere} \times {L_{path}}^{3/2})$ for the first run of Dijkstra's algorithm. D_{COM} is the distance from the grid voxel to the center of mass of the protein. L_{path} is the total length of the path. The purpose of this run is to find the other endpoint. Then, the second run of Dijkstra's algorithm is performed to find the lowest cost path between these two endpoints. In this run, the cost is set as $1/R_{sphere}$.
3) For each pore without exposed surface patch, we have to choose one endpoint of the path before applying Dijkstra's algorithm. For this, we choose the grid voxel with maximum distance to the geometric center of the pore. Then Dijkstra's algorithm is used

to search the other endpoint and also the lowest cost path. The cost here is set as $1/(R_{sphere} \times L_{path}^{3/2})$.

When the cost of each grid voxel is $1/R_{sphere}$, the lowest cost path for the protein structure corresponds to the pore with the largest radii along the path. In Cases (2) and (3), we used the additional factor $L_{path}^{3/2}$ to favor longer pore axes. Otherwise the algorithm will preferentially find nearby grid voxels with large R_{sphere} and the found pore axis will be short and only cover a small part of the pore. In Case (2), D_{COM} is included to favor endpoints deeply buried inside the protein. This strategy is motivated by the observation from existing 3D structures of proteins that longer and deeper pores of a protein are typically the ones that are functionally relevant. As the path found in the first run of Dijkstra's algorithm is biased due to the weighting factor D_{COM}, the second run with unbiased cost is necessary.

3.3.3 GateOpen: gate opening of neighboring pores

It is of great interest to identify cases where two neighboring pores may be connected by small conformational changes of the protein side chains. For this, we define pore-lining residues of a pore as those residues that have at least one atom assigned to a grid voxel in direct contact with a grid voxel belonging to the pore. Any two pores with shared pore-lining residues are then defined as ''neighboring pores''. The shared pore-lining residues are candidates to act as a gate between the neighboring pores. To check the connectivity, we try to open the gate by rotating the side chains of the gating residues into sterically allowed positions. The dead-end elimination algorithm [57, 58], which is commonly used for protein side chain prediction, was

adopted to deal with the combinatorial task of identifying the allowed side chain rotations of the gating residues.

We constructed rotamer libraries for the side chains of 17 amino acids (the small or stiff residues glycine, alanine, and proline were excluded). Dihedral angles of each rotatable bond were scanned in steps of 30° from 0° to 360°. For each amino acid, the rotamer library contains full sets of combinations of dihedral angles of all rotatable bonds. A dihedral combination was excluded if it resulted in serious clashes with other atoms in the considered protein structures so that two atomic vdW radii overlapped by more than a given threshold (set here to 1.0 Å by default).

First, the gating residues are mutated to glycine. From a structural point of view, this is equivalent to deleting the side chains of the gating residues. Thus the space previously occupied by the side chains is freed and one can check whether the two neighboring pores are now connected in this mutated protein. If this is the case, then the side chain rotamers of the original gating residues are placed back, and the degree of opening of the gate is evaluated. The potential energy for the evaluation is

$$E_{gate} = \sum_i E_{sphere}(i_r) - \sum_i E_{CA}(i_r) - \sum_i \sum_j E_{interaction}(i_r, j_s) \qquad (3.1)$$

For rotamer i_r, $E_{sphere}(i_r)$ is taken as the sum of the radii R_{sphere} of all side chain atoms. Large $E_{sphere}(i_r)$ values mean that the rotamer i_r is located near the pore axis and the gating residues are trying to block the pore. $E_{CA}(i_r)$ is the sum of all distances between the side chain atoms and the gate center. The gate center is the center of the Cα atoms of all gating residues. Small $E_{CA}(i_r)$ values indicate that the side chain atoms are located near the gate center and tend to

close the gate. $E_{interaction}(i_r,j_s)$ is the interaction term for rotamers i_r and j_s. It is taken as the sum of the distances between all atom pairs of rotamers i_r and j_s. Small $E_{interaction}(i_r,j_s)$ values mean that these two rotamers are located close to each other and narrow the pore radius of the gate. The dead-end elimination algorithm then identifies a combination of rotamers with minimum potential energy so that a maximal space surrounded by the gating residues is opened. Finally, PoreID is performed on the new structure to check the connectivity of the two neighboring pores.

The source code, installation instructions, example files, and a short documentation are available from http://gepard.bioinformatik.uni-saarland.de/software/poreid/poreid-page. The PROPORES software is made available under the GPL license.

3.4 Results and Discussion

3.4.1 Input, output, and options

The program PoreID requires as input the coordinate file of the target protein in standard PDB format. Users may either simply adopt the default settings for general cases or specify three optional parameters, for example, for sampling at a higher resolution or for detecting narrower pores. These parameters are the side length of the cubic grid voxels, the probe radius, and the trimming depth for pores on the protein surface. The side length of the cubic grid voxels determines the resolution of the pore identification and the default value is set to 1.0 Å. In our experience, 0.5 Å is sufficient for generating fine and smooth surfaces of the identified pores. An identified pore encloses a volume in which the probe can be freely rolled without overlapping with any protein atoms. The default value of the probe radius is taken as 1.2 Å. This

value is slightly smaller than 1.4 Å, which is commonly used as a probe for water when computing the popular solvent accessible surface area. [59, 60] We adopted a smaller probe radius here by considering the possible error due to the specified grid resolution. Shallow pores have to be trimmed; otherwise small disjoint pores on the protein surface may be connected by shallow pore grid voxels to form a single, large pore which covers a wide area of the protein surface. The default value of the trimming depth is taken as 1.4 Å, which means that the pores on the protein surface would be trimmed to 1.4 Å in depth.

The output of PoreID consists of three files for each identified pore. These are the pore grid voxels in PDB format for graphical displays, a list of pore-lining residues and an information file that is used as the input for pore axis determination by PoreTrace. PoreTrace has no optional parameters to be specified. The output of the PoreTrace program is a file containing pore axes represented by grid voxels in PDB format with an additional column for the pore radius. GateOpen requires a PDB file of the target protein and the list of the gating residues as input. An optional parameter is the tolerance of two overlapping atoms, and the default value is set as 1.0 Å. The output is a PDB file of the target protein where the side chain conformations of the gating residues have been rotated in order to maximize the connectivity of half pores. All examples in the ‘‘Case Studies’’ section below were generated by setting the PoreID parameters resolution to 0.5 Å, the probe radius to 1.2 Å, and the trimming depth to 1.4 Å. The single exception is that we adopted the PASS parameters when comparing the results from PoreID and PASS. GateOpen was performed using 1.0 Å for the overlapping tolerance.

3.4.2 Performance of the methods

For simplicity, all programs were implemented in Perl programming language and are thus not optimized for speed. The running times of the three programs (PoreID, PoreTrace, and GateOpen) were measured on an Opteron 2.4 GHz CPU. For PoreID, the protein structure of the leucine transporter (LeuT; PDB ID: 2A65, 4044 heavy atoms) was used as benchmark. It is a common issue for grid-based pore identification that a higher resolution will result in smoother pores but this requires longer running times and more computer memory. The discrete property of grid-based design is revealed by the running time of PoreID, and this is also the reason of the sharp increase between resolutions of 0.5 Å and 0.6 Å (Figure. 3.3A]. For example, to scan the grid voxels within 1.7 Å (vdW radius of the carbon atom) in 1D, four operations are required for the resolution of 0.5 Å but three operations for 0.6 Å. The probe radius and the trimming depth have no considerable effect on the running time. For an identified pore, the estimated volume becomes larger by specifying a smaller probe radius. When a larger trimming depth is used, the pore on the protein surface will be estimated at a smaller volume or even be split into smaller pores.

The performance of PoreTrace was evaluated on the largest pore of the structure 2A65. The running time increased exponentially for higher resolutions (Figure. 3.3B). On the other hand, specifying higher resolutions for PoreTrace resulted in smoother pore axes and more accurate radius profiles. According to the analyses of running times of PoreID and PoreTrace, 0.6 Å is a recommended value for the resolution by simultaneously considering the accuracy and efficiency. GateOpen is not grid-based and its running time is related to the combinations of the

rotamers of the gating residues. Usually, it takes less than 3 min to get an optimal combination of rotamers for each single gate composed of two or three residues.

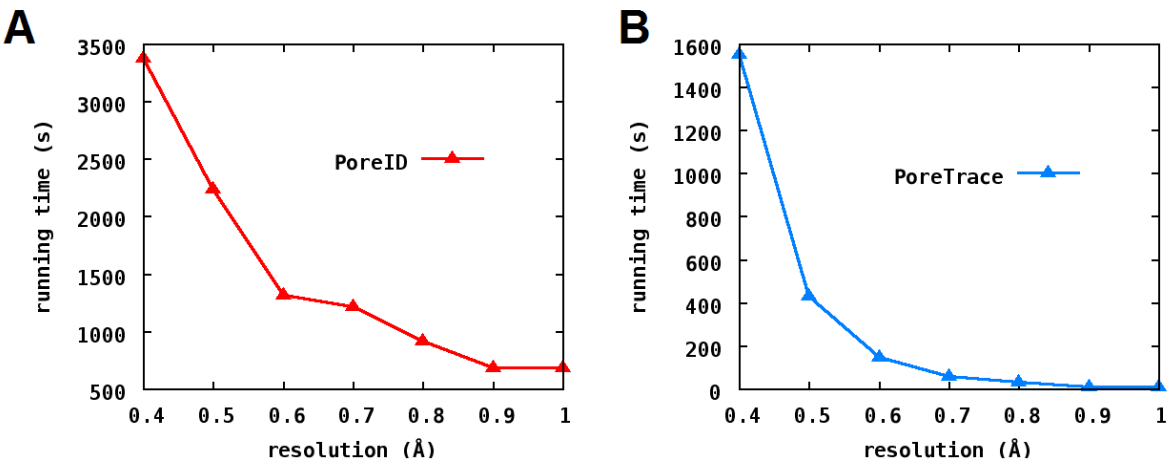

Figure 3.3 Performance of PoreID and PoreTrace. The running times of PoreID (A) and PoreTrace (B) for the leucine transporter (PDB ID: 2A65) measured for different grid resolutions.

3.4.3 Case study

- Aquaporin

The aquaporin water pores facilitate rapid and highly selective transport of water molecules across membranes. The topology of aquaporin is formed by six transmembrane helices and two short pore helices in an ‘‘hourglass’’ fold. Here, we analyzed the crystal structure of a yeast aquaporin (PDB ID: 2W1P). The water conducting channel of this structure is occluded both at the extracellular and cytoplasmic sides. The residues F92 and R227, which are conserved in the aquaporin family, function as a selective filter and block the channel at the extracellular side, whereas Y31 blocks the channel at the cytoplasmic side. [61] After pore identification by PoreID and comparison of pore-lining residues, F92 and R227 were found as the gating residues at the

extracellular side of the protein, whereas Y31 and A190 form the gate at the cytoplasmic side (Figure. 3.4A). Both gates are closed in this crystal structure so that water molecules cannot permeate through the protein. The GateOpen program was then used to open these two gates. As shown in Figure 3.4B, the χ^2 dihedral angle of the F92 side chain was rotated by 61°, and the phenyl ring was turned to face the connected pore. The side chain of R227 was flipped toward the opposite direction by rotating χ^1 by 150°. Modifying the positions of the gating residues freed the preoccupied space between the pores, so that the neighboring pores at the extracellular side became connected. At the cytoplasmic side, the χ^1 angle of Y31 was rotated about 45°. This shifted the side chain of Y31 away from A190 and opened the gate.

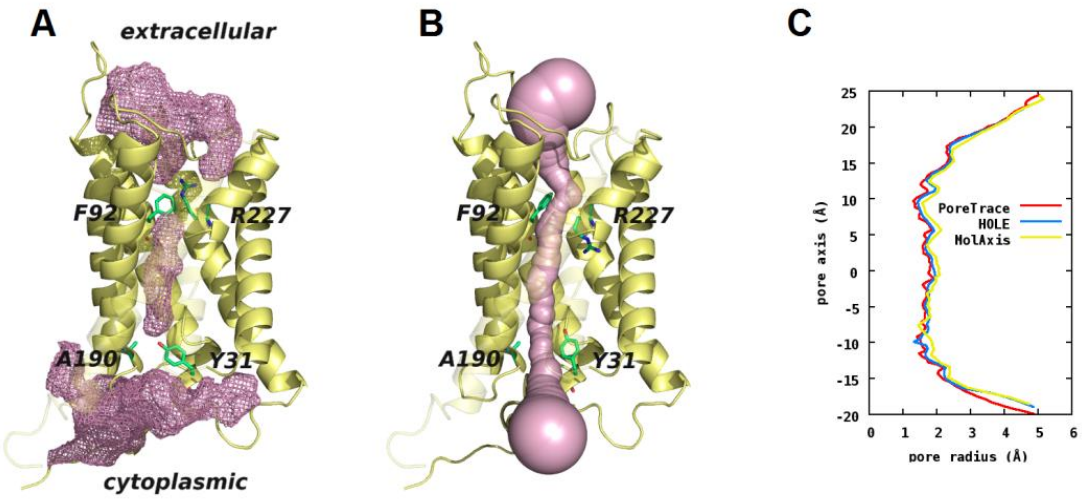

Figure 3.4 Water conducting channel of yeast aquaporin. The protein is shown in ribbon representation and the gating residues are shown as sticks. Helices at the most front layer were made transparent for clarity. (A) The pink meshes represent pores that are separated by the upper (F92 and R227) and lower gates (Y31 and A190). (B) The pink surface depicts the path of the water conducting channel. Both gates were opened by GateOpen. (C) Radius profiles of the opened water conducting channel computed by PoreTrace, HOLE, and MolAxis.

We also applied our toolkit to another member of the aquaporin family from spinach with PDB ID 2B5F [62]. In that structure, the main channel is blocked by a single gate and PoreID picked F81, H210, and R225 as gating residues. After repositioning the gating residues by GateOpen using an overlapping tolerance of 1.0 Å, F81 and H210 showed only small shifts whereas R225 was flipped inward the channel similar as found for R227 in the modified 2W1P structure (Figure 3.5). In both cases, this happened because GateOpen identified small pockets to accommodate the guanidinium group of arginine. However, the conformation of arginine seemed to be quite strained which would, in reality, result in a low occupancy unless further conformational adaptation takes place. For that reason, we reran GateOpen on the gating residues of the 2B5F structure using a more rigorous overlapping tolerance (0.4 Å). Interestingly, this resulted in a more relaxed conformation of R225 with a χ^1 angle of 53° and a similar side chain orientation as in the original structure (Figure 3.5). We note that applying such a more rigorous overlapping tolerance for repositioning the gating residue may not always result in more relaxed conformations. Yet, the degree of opening of the gate can be predicted with higher confidence.

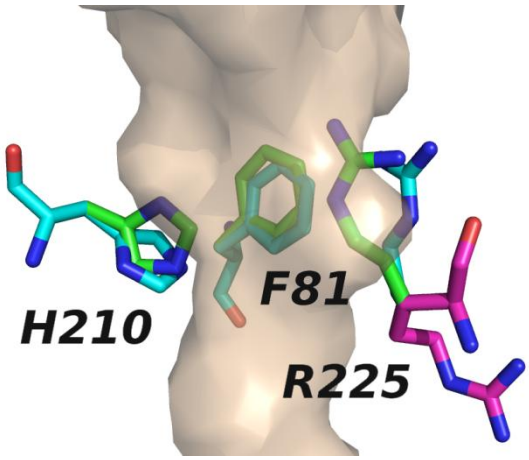

Figure 3.5 Repositioning of gating residues in spinach aquaporin. Green conformations are from the original protein structure 2B5F. The conformations repositioned by GateOpen with overlapping tolerance as 1.0 Å and 0.4 Å are shown in magenta and cyan, respectively. The surface representation depicts the volume of the water conducting channel according to the protein conformation shown in cyan.

The pore axis and radius profile of the connected pore of the modified 2W1P structure were computed by PoreTrace, HOLE, and MolAxis (Figure 3.4C). For fair comparison, we adopted the same vdW radii for protein atoms and the same grid resolution of 0.5 Å for all three programs. The PoreTrace results are similar to those of the MolAxis and HOLE programs. Especially, the trajectories of pore axes obtained from PoreTrace and MolAxis largely overlap. However, MolAxis found slightly larger radii along the pore than PoreTrace and HOLE because MolAxis approximates each protein atom by using a collection of unit spheres which are the references to estimate the pore radii. The radii of the narrower pore region (i.e., radius < 2.0 Å) computed by PoreTrace are slightly smaller than those computed by HOLE. This is attributed to the resolution of grid-based methods because the trajectory of the pore axis has to be represented by the pore grid voxels. The deviation of the pore radii at the cytoplasmic end is due to the difference in the definition of the endpoints. PoreTrace defines the endpoints first and then searches for the optimal path between the endpoints. The HOLE program tries to search the largest pore radius at every slice and stops the search when the considered pore radius last matches the given threshold. At an open area where few protein residues surround the pore, HOLE prefers to find a larger pore radius. This deviation here is not substantial because the pore radii of this region are more than twice the radius of the water molecule and the surrounding protein residues are more flexible here. However, in some complicated pore regions, the trajectory of the pore axis generated by HOLE may be guided into the branch and then go back to the direction of the start vector. This is an issue of the slice and optimization methods. When checking these positions in the protein structure, the trajectory of HOLE is

fragmented whereas the trajectories of PoreTrace and MolAxis are continuous and thus appear more reasonable for this system.

- Tryptophan synthase

The enzyme tryptophan synthase catalyzes the final two steps in the biosynthesis of L-tryptophan. This enzyme is a $\alpha_2\beta_2$ tetrameric protein complex that is formed from two dimeric $\alpha\beta$ enzyme units. [63] The α subunit catalyzes the cleavage of 3-indole-D-glycerol 3'-phosphate (IGP) to yield D-glyceraldehyde 3-phosphate (G3P) and indole. The β subunit catalyzes the replacement of the hydroxyl group of L-serine by indole. The common intermediate, indole, is directly transferred between the active sites of the α and β subunits through a 25 Å long channel that is buried inside the protein complex. [64] The substrate channeling phenomenon of tryptophan synthase presents a good example of a protein nanomachine. As of March 15, 2011, there are 50 structures of tryptophan synthase $\alpha\beta$ complexes deposited in the Protein Data Bank. In most of them, several residues are missing at the C terminus of the α subunit. To avoid that missing residues leave a large cleft that would be identified as a pore, we selected the structure 2CLI as the target for this case study, because it contains a more complete C-terminal loop of the a subunit.

First, we applied PoreID to the structure 2CLI and compared the results to those of the PASS algorithm using the same probe radius (1.8 Å) and resolution (0.7Å). For the protein interior, the results from PoreID are consistent with those from PASS. Most of the pores on the protein surface identified by PASS were also found by PoreID. However, a few pores were detected by only one method and the discrepancies happened at shallow pores (Figure 3.6). Pore

identification of PASS refers to the ‘‘burial count’’ of the probe, which is the number of protein atoms surrounding the probe. A pore surrounded only by a few protein residues may not be identified by PASS. In contrast, PoreID considers the landscape around the void grid voxel. This is why the pores identified by PoreID often cover a larger surface area or volume than those detected by PASS.

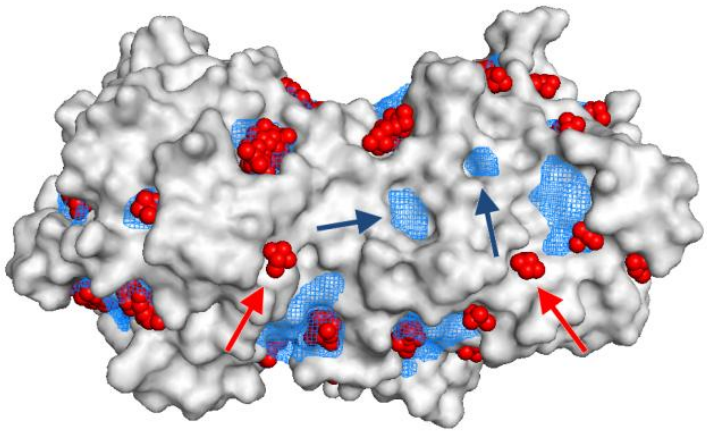

Figure 3.6 Pore identification by PoreID and PASS on the protein surface of tryptophan synthase (PDB ID: 2CLI). The protein surface is shown in gray. Blue meshes are pores identified by PoreID and red bead-clusters are pores located by PASS. Blue arrows indicate pores identified by PoreID but not by PASS. Red arrows indicate pores identified by PASS but not by PoreID.

PoreID identified the two largest pores that are responsible for forming the long interconnecting channel for indole. One of them is buried in the β subunit. The other one is mainly located in the α subunit and opens at the dimerization surface between the α and β subunits. The pore-lining residues shared by both pores, βL188 and βF280, appear to be the gate of the channel. When GateOpen was applied to βL188 and βF280, the χ^1 torsion angle of βF280 was rotated by 77°, so that the phenyl ring approached βY279. The χ^1 angle of βL188

was rotated by 90° to shift the side chain away from βF280 (Figure 3.7). The side chain rotations of βL188 and βF280 led to the opening of the gate of the interconnecting channel. Subsequently, we compared the orientations of the rotated side chains against those in the structure 2CLF. That structure determined at high ligand concentrations contains an open gate because a second IGP analog bound near βF280. [65] Due to steric hindrance by βF279 the displacement of the rotated βF280 is not as large as that in 2CLF. The side chain of βF279 is lifted a bit in 2CLF so that the phenyl ring of βF280 has more space. βF279 and βF280 were mentioned in [65] as the gating residues of the channel. In our opinion, βF280 is the gating residue and βF279 adopts the role of a door stop wedge which determines the degree of opening of the gate. In contrast to βF280, βL188 has a minor effect on the gate due to its smaller side chain. It remains almost in the same orientation in the structures 2CLI and 2CLF.

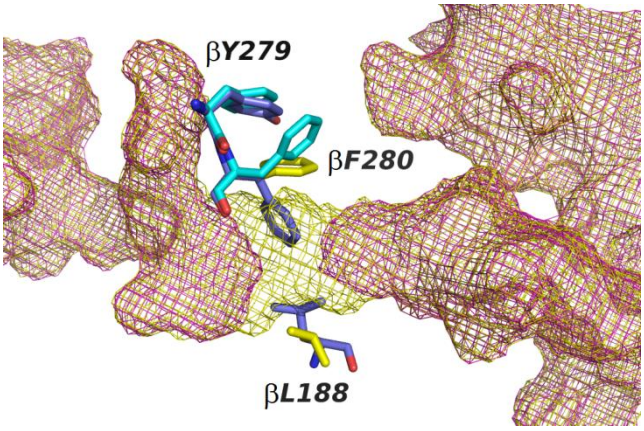

Figure 3.7 Gate opening of tryptophan synthase. The conformations of protein residues colored in blue belong to 2CLI, yellow conformations are from 2CLI after rotation by GateOpen and residues in cyan are from 2CLF. Magenta meshes are the pores identified by PoreID in the structure 2CLI. The left mesh is buried mainly in the α subunit and the right one is in the β subunit. The yellow mesh depicts the opened interconnecting channel of the structure 2CLI modified by GateOpen.

- Leucine transporter (LeuT)

The LeuT is a homolog of the neurotransmitter sodium symporters that use sodium and chloride electrical chemical gradients to transport neurotransmitters through the membranes of neuronal and glia cells. Here we used the crystal structure of the bacterial LeuT (PDB ID: 2A65) [66] as an example. LeuT is a transmembrane protein with 12 transmembrane helices. In this structure, the binding site of the substrate (L-leucine) is occluded both at the extracellular and the cytoplasmic sides. Ziegler and coworkers have recently described an alternative access model how large scale conformational changes of the channel lead to an alternative opening of the channel toward either the exoplasmic or the cytoplasmic sides. [67] In particular, it seems unfeasible to open the channel in the 2A65 conformation toward the cytoplasmic side without large scale conformational changes of the protein. This is, however, beyond the scope of this article. Here, we focused on the pores located at the extracellular side and identified the largest pore at the extracellular side and the binding pocket of leucine by PoreID. These two pores share the lining residues G26, Y108, and F253 (Figure 3.8A). Y108 and F253 were already pointed out as gating residues in [66]. Thus, we applied the GateOpen algorithm to open the gate between these two pores. Figure 3.8B shows that Y108 was only slightly twisted due to the steric hindrance of surrounding residues. Yet, the χ^1 angle of F253 was rotated by 136° to move the phenyl ring away from Y108, and this is the main modification necessary to achieve opening of the gate. The PDB also contains a LeuT structure (PDB ID: 3F3A) where the leucine binding pocket is open toward the extracellular side. That LeuT structure was also cocrystalized with L-tryptophan bound in the leucine binding pocket. [68] As shown in Figure 3.8C, the open conformation of the leucine binding pocket in 3F3A is due to several backbone rearrangements

to accommodate the bulky tryptophan ligand. The root mean square deviation of the backbones between the structures 2A65 and 3F3A is 1.18 Å. The side chain orientations of Y108 and F253 are similar between these two structures.

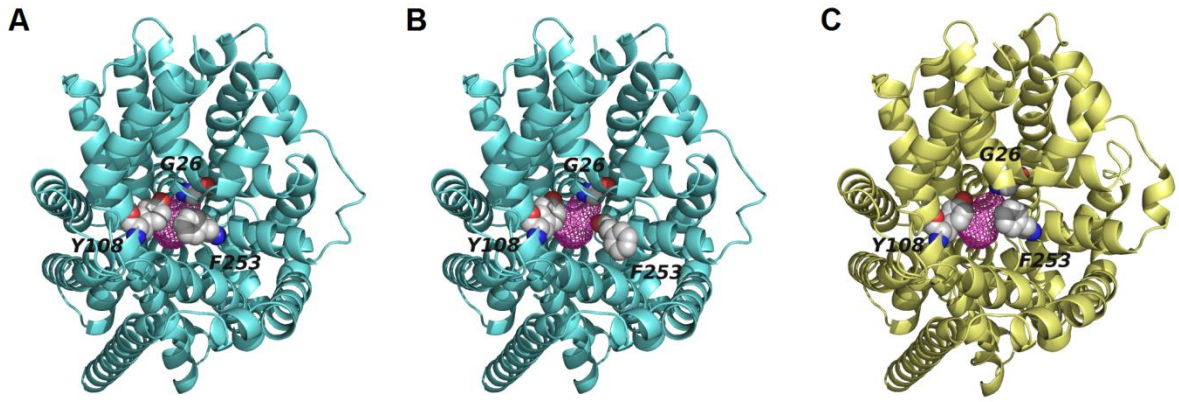

Figure 3.8 Gate opening of the leucine transporter. The extracellular side of the protein faces the reader. Gating residues are shown in CPK. The magenta meshes under the gating residues represent the leucine binding pocket. (A) The gate is closed in the structure 2A65. (B) The gate is opened by applying the GateOpen algorithm to the structure 2A65. (C) The gate is open in the structure 3F3A.

- Acetylcholinesterase (AChE)

The important enzyme AChE terminates synaptic signaling by hydrolysis of the neurotransmitter ACh into acetate and choline. The catalytic reaction of AChE proceeds at nearly diffusion controlled speed and the turnover number is about 7.4×10^5 min^{-1}. [69] However, the 3D structure of AChE surprisingly revealed that the catalytic triad is located at the bottom of a 20 Å deep gorge. From there, the products likely leave the active site through transient backdoors in the protein that were revealed in molecular dynamics simulations. [70] Clearly, the substrates and the products have to pass long distances to reach the catalytic center and leave the enzyme, respectively. The inward electrostatic field at the gorge explains

the guiding mechanism of the substrate. [71] Here, we used the structure of AChE from Torpedo californica (PDB ID: 1EA5) as target for studying the backdoor opening.

PoreID identified residues V71, D72, E73, S81, N85, P86, and L127 as shared pore-lining residues between the long gorge and the outside concavity. These residues form a thin wall in between the two pores (Figure 3.9A). The residues responsible for the backdoor (W84, V129, and G441) pointed out by the simulation study [70] are, however, totally different from these shared pore-lining residues. In the MD study, no opening event was observed in the thin wall during 119 ps of MD simulation. When manually assigning W84, V129, and G441 as gating residues for the GateOpen operation, the backdoor did not open because M83, which is located in front of W84, blocked the exit (Figure 3.9B). This suggests that M83 should be considered as a further key residue of the backdoor in agreement with [71]. Thus, we reran GateOpen by considering M83, W84, V129, and G441 as gating residues. Now the backdoor opened and the observed pore radius of the open backdoor (1.68 Å) is sufficient for a water molecule to pass through. It seems that there are two gates for the backdoor. The side chain of M83 was lifted up by a 160° rotation of χ^1 to open the first gate. The side chain of W84 was flipped inward the protein by a 153° rotation of χ^1 to open the second gate (Figure 3.9C). The χ^1 angle of V129 was rotated by 50° toward G441. However, V129 and G441 have only small effects for the opening of the backdoor because of their small side chains.

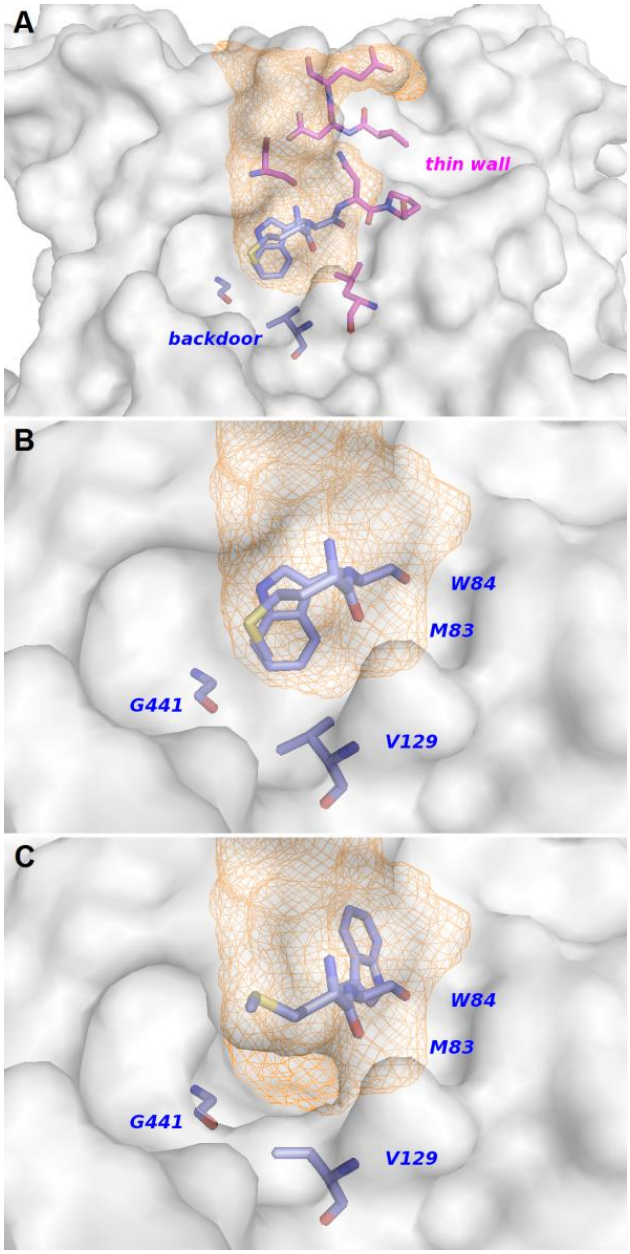

Figure 3.9 The backdoor of acetylcholinesterase. AChE is shown in transparent surface and protein residues in sticks. Orange meshes depict the long gorge of the active site. (A) Protein residues in magenta are the shared pore-lining residues between the gorge and the outside concavity. Residues in blue are key residues of the backdoor. (B) Gating residues block the backdoor. (C) Opened backdoor by rotation of the gating residues.

3.4.4 Discussion

Pore axis and radius profile are provided by PoreTrace as descriptors of an identified pore. These descriptors allow a quick look at whether the substrate can or cannot pass through the pore and where the bottleneck is. It should be stressed that pore axes are optimal solutions

from a geometric point of view and may not be the actual path for the ligand because we do not consider the physicochemical properties of the pore-lining residues during pore axes determination. On the other hand, biomolecular molecular dynamics simulations allow computing accurate potentials of mean force for the passage of solute molecules through permanent or transient channels in proteins but, of course, such simulations demand a large computational effort. [72, 73]

Gating mechanisms play an important role in channel proteins or enzymes for selectivity or preventing leakage. [74] PoreID identifies pores and pore-lining residues. Shared pore-lining residues are defined as a candidate gate between pores. We can provide this information to biologists as guidance for designing enzyme inhibitors or protein mutants. CAVER and MOLE also provide additional information about the ‘‘gorge residues’’ which are residues located at the narrowest part of the channel. In some crystal structures as in aquaporin, a continuous channel may be separated into small pores by gating residues. Sometimes, opening the gate of a protein also requires movements of the protein backbone. Mimicking such conformational dynamics by computational methods is very challenging. Much easier to model is to sample the rotations of the gating residues to sterically allowed regions. Whereas this does not guarantee that gate opening is achieved, it is an efficient way to inspect this. The conformation with rotated gating residues can be regarded as one snapshot of the conformational ensemble. A similar strategy is commonly used in modern docking algorithms that also consider partial flexibility of the receptor. Repacking of the specified side chains either by the combination of side chain conformations from a rotamer library [75] or sampling the conformational space of side chain torsion angles [45] is an efficient way to achieve the goal. GateOpen was designed

according to the same concept. In this way, we tried to introduce a limited amount of protein flexibility into pore identification and thus provide a more complete view of the pores forming inside protein structures.

4 PRIMSIPLR: Prediction of Inner-Membrane Situated Pore-Lining Residues for α-helical transmembrane proteins

This work had been published as "Nguyen D, Helms V, Lee P-H: **PRIMSIPLR: Prediction of inner-membrane situated pore-lining residues for alpha-helical transmembrane proteins**. *Proteins* 2014, in press". My own contribution of this work was to compile the non-redundant training set PH90 and testing set Test23, to perform the feature selection and the optimization of SVM, to evaluate and compare the performance of our predictor to MEMSAT-SVM, to program the standalone version of PRIMSIPLR and to write the manuscript for publication.

4.1 Motivation

Pore-lining residues (PLRs) are crucial for the function of membrane transporters and channels because they directly contact the substrates or the water shell surrounding them. They are thus involved in recognition, desolvation, binding and transportation processes of protein-substrate interactions. To our knowledge, there exists so far only the prediction method mentioned in Section 1.3 for identifying pore-lining residues developed by Nugent and Jones. [13] Their method identifies PLRs located in transmembrane helices of proteins and this prediction has been integrated into MEMSAT-SVM as an extension. MEMSAT-SVM first predicts the topology of TM helices and then identifies PLRs with respect to the predicted helices. Although most PLRs are indeed located in transmembrane helices, a certain portion of PLRs are located in loop regions or other locations. In this chapter, we present a single step method to predict PLRs of α-helical transmembrane proteins from primary amino acid sequences. This

method is based on a comprehensive data set termed PH90 and a new SVM classifier (see Materials and Methods section of this chapter), and has been tested by stringent cross-validation. With improved prediction accuracy, this method should be of great use in the annotation of genomic sequence data and also provides clues to experimental biologists working on transmembrane proteins.

4.2 Materials and Methods

4.2.1 Preparation of the data set

To collect a comprehensive data set of pore-containing α-helical transmembrane proteins, we accessed the regularly updated database PDBTM [22] (Oct 19th, 2012 version) to obtain the list of PDB IDs with chain indices of all α-helical transmembrane proteins deposited in the RCSB Protein Data Bank [76]. We retrieved PDB files according to this list and filtered out structures that were not determined by experimental techniques (i.e. theoretical models). In PDB files, the primary sequence is recorded in the "SEQRES" section. However, to facilitate purification or crystallization, the protein sequence of the determined structure sometimes differs from that of the wild-type protein due to, for example, mutations, non-native amino acids, insertions, deletions, tags, fusion protein, and so on. These modifications may be misleading when applying sequence identity-based clustering. Hence, we adopted wild-type protein sequences taken from the database Uniprot [77] according to their Uniprot IDs recorded in PDB files instead of the sequences in the "SEQRES" section.

Protein sequences of 1329 membrane protein structures were then clustered by the program BLASTClust requiring length coverage above 90% and sequence identity below 25%. Among all members of one cluster, we manually picked the protein with the visually most obvious and largest pores inside its 3D structure as the representative (exemplar) of the respective cluster. Some structures containing only one transmembrane helix or without any detectable pores inside the protein were discarded during this step.

We then used the program PoreID of our tool package PROPORES [78] to identify pores and to annotate PLRs of the representative structures. However, a pore identification software such as PoreID usually provides tens of pores for an input structure. To confirm that the identified pores are biologically functional, we reviewed the original articles where the determined structures were reported and annotated only the pores mentioned in these articles. Figure 4.1 shows an example for the glycerol facilitator protein. After the mentioned data processing steps, 92 chains from 90 protein structures were retained. This data set with the obtained structure based annotations was named “PH90” set. Their PDB IDs and protein names are listed in Table A.1 in the Appendix section.

From the OPM database and PPM server [23], clear boundaries of the water/lipid interface were estimated for each protein. With the annotation of PLRs and the water/lipid boundary of the protein structure, we then classified each residue of a protein sequence into four types according to its location. The type “P” is used for PLRs in the transmembrane region, “M” denotes all other residues in the transmembrane region, “O” is assigned to residues outside of the membrane, and “N” are those residues whose coordinates are not contained in the PDB file.

Figure 4.2 shows an example of this topological classification for the primary sequence of the bacterial glycerol facilitator (PDB ID: 1FX8). The full data set is available at http://service.bioinformatik.uni-saarland.de/PRIMSIPLR/.

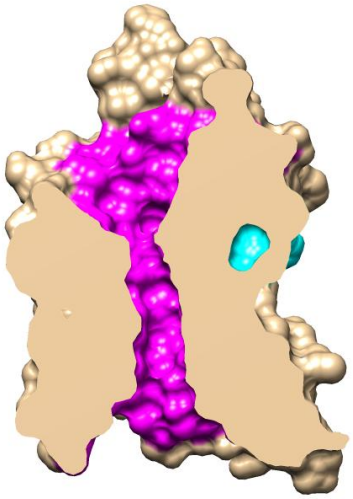

Figure 4.1 The cross section of glycerol facilitator (PDB ID: 1FX8) is shown in surface representation. Both magenta and cyan areas are pores identified by PoreID of PROPORES [78]. Only the magenta pore is considered as biological functional and annotated in our datasets because it is a glycerol conducting channel indicated in the structure determination work by Fu and his colleagues [79].

```
>sp|P0AER0|GLPF_ECOLI Glycerol uptake facilitator
protein OS=Escherichia coli (strain K12) GN=glpF PE=1
SV=1|PDB id 1FX8_A

MSQTSTLKGQCIAEFLGTGLLIFFGVGCVAALKVAGASFGQWEISVIWGLGVAMA
IYLTAGVSGAHLNPAVTIALWLFACFDKRKVIPFIVSQVAGAFCAAALVYGLYYN
LFFDFEQTHHIVRGSVESVDLAGTFSTYPNPHINFVQAFAVEMVITAILMGLILA
LTDDGNGVPRGPLAPLLIGLLIAVIGASMGPLTGFAMNPARDFGPKVFAWLAGWG
NVAFTGGRDIPYFLVPLFGPIVGAIVGAFAYRKLIGRHLPCDICVVEEKETTTPS
EQKASL

NNNNNOOOOOMMMMMMMMMPMMMMMMMPMMPOOOOOOMMPMMPMMMPMMMPMMM
PMMMPOOOOPPPPMMPMMMPOMOOOOOOOOMMMMMMMMMMMMMMMMMMMMMMO
OOOOOOOOOOOOOOOOOOOOOOPPPPPPPOOOOOMMMMMMMMMMMMMPPMMPPM
POOOOOOOOOOOMPPMMPMMMPMMMMMMMPPPPPPPPPMMPMMMMMMMMMMOOO
OOMMOOOOOMMMMMMMMMMMMMMMMMMMPOOMOOOOONNNNNNNNNNNNNNN
NNNNNN
```

Figure 4.2 Pore annotation of glycerol facilitator (PDB ID: 1FX8).The first two lines are adopted from Uniprot. The third line is the annotation of each residue according to its structural environment. “P” is for the pore-lining residue inside the membrane. “M” is for the residue inside the membrane except pore-lining residue. “O” represents the residue located outside of the membrane. N indicates that the coordinates of the residue are not contained in the PDB file.

4.2.2 Machine learning and prediction

For each single residue of the protein sequences of the training set as well as for input sequences uploaded to our webserver, we generated 24 features as input for the machine learning algorithm. The first 20 features are position-specific scores generated by PSI-BLAST [18]. This profile captures the conservation pattern of the protein from a multiple sequence alignment of homologous protein sequences. Three iterations of PSI-BLAST were run for an input sequence against the non-redundant (nr) database and using commonly used parameters (i.e. the word size was set to 3, the penalty of gap opening to 11, the penalty of gap extension to 1, and threshold to 0.001). The next three further features characterize physical properties of the 20 native amino acids, namely hydropathy, polarity and flexibility. As hydropathy scale, we used a modified version of the Kyte-Doolittle hydropathy scale proposed by Juretic *et al*. [80] In their study, the hydropathy of tryptophan was increased and that of alanine was decreased with respect to the Kyte-Doolittle scale to obtain a better prediction of transmembrane helices. The polarity indices for amino acids were taken from the study of Zimmerman *et al*. They approximated the polarity index of an amino acid as the relative electric potential generated by the dipole and the charged group of its side chain. [81] Flexibility scales typically account for the typical degree of conformational flexibility of amino acid side chains. The flexibility scale of amino acids adopted here was proposed by Vihinen *et al.* to predict continuous epitopes on proteins. [82] It was derived by averaging the B-factors of each amino acid type in the PDB files of 92 structures. The last feature is the evolutionary conservation score computed by the program Rate4Site [83]. Rate4Site implements a Bayesian approach to estimate the position specific evolutionary rate of a protein sequence.

The machine learning algorithm applied in this study is support vector classification implemented in the software package LIBSVM. [84] The SVM was implemented to classify and identify PLRs in the transmembrane region (type P of our annotation) against all other residues of the protein structure (types M, N and O). We tested several combinations of features to determine how these features affect the performance of the predictions (see Results section of this chapter). The performance of each SVM classifier was evaluated by the common measures accuracy, sensitivity, specificity and Matthews's correlation coefficient (MCC).

4.3 Results and Discussion

4.3.1 Amino acid composition

Our non-redundant PH90 dataset consists of 92 structures of α-helical transmembrane proteins with central pores. 3460 of the 36885 residues were assigned by PoreID as PLR or type "P". The remaining 33425 residues were classified to any of the other types M, N or O. The ratio between P and (M, N, O) is 1 : 9.66. N-type residues were included in the group of non-PLR residues together with M- and O-type residues. The reason for this is that most of the unresolved parts of X-ray structures of TM proteins (residues of type N) are located in N-terminal, C-terminal or flexible loop regions of transmembrane proteins. Hence, they are very unlikely to belong to the group of PLR residues.

The amino acid composition for residues of type P, M and O is shown in Figure 4.3. As expected, non-polar amino acids such as A, I, L and V are relatively abundant (> 10%) in M-type residues because they are either buried inside the protein structure or exposed to the

hydrophobic lipid bilayer. Unexpectedly, A, I, L and V are also frequently found as P-type residues (>8%). This means that non-polar residues are important components of pores. Even if the substrate is hydrophilic, non-polar residues form hydrophobic patches and facilitate substrate transportation as in aquaporin. [85] PLR positions ("P" positions) contain larger fractions of charged (D, E and R) and polar residues (H, N and Q) than M-type residues because they may be involved in contacts with buried solvent molecules in the pore or with the charged or polar transported substrates. Amino acids W and Y with aromatic side chains occur slightly more often in P-type residues than in the other two cases. These amino acids often function as gating residues of channel proteins or transporters as mentioned in Chapter 3. In contrast to membrane positions, O-type residues are substantially enriched in charged amino acids such as D, E, K and R.

When mapping structurally identified PLRs onto the protein sequences, 94.4% of the pore-lining fragments are shorter than three residues. This is due to the fact that generally only one side of pore forming α-helices faces the pore and perfect α-helices contain 3.6 residues per turn (Figure 4.4). One of the exceptions in the PH90 set is the ABC transporter with PDB ID: 3QF4 chain B. Its longest pore-lining fragment is 17 residues long. This protein is heterodimeric and contains a large channel formed between two protomers. This pore-lining fragment is part of the transmembrane helix that is not aligned with the other helices and protrudes into the central cavity.

To characterize the local environment of PLRs in the primary sequence, we also analyzed the distribution of neighboring residues of PLRs annotated by PoreID (see Figure 4.5). The pattern

of neighboring PLR peaks is consistent with the periodicity of 3.6 residues per turn of an ideal α-helix. The second and third highest frequency of PLRs was observed at the third and fourth positions downstream (+3, +4) and upstream (-3, -4) from the central PLR. This means that a PLR located in a TM helix has neighboring PLR residues one helix turn away in both directions. Although further peaks of PLR were observed at the 7th and 11th residues, the fraction of non-PLR (types M and O) in these positions is considerably larger.

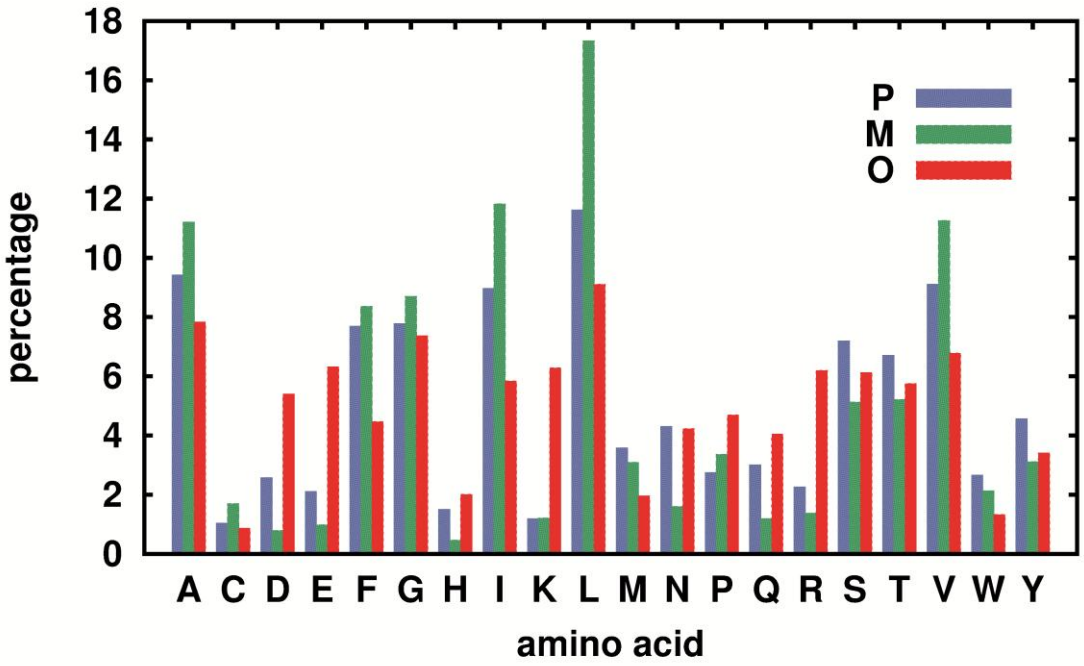

Figure 4.3 Amino acid composition of pore-lining residues (type P) and non-pore-lining residues (types M and O).

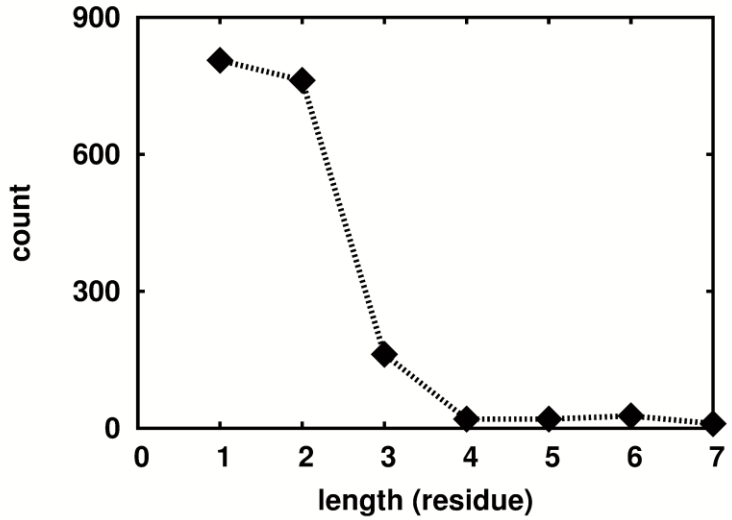

Figure 4.4 Length of pore-lining fragments when mapped on the primary sequences

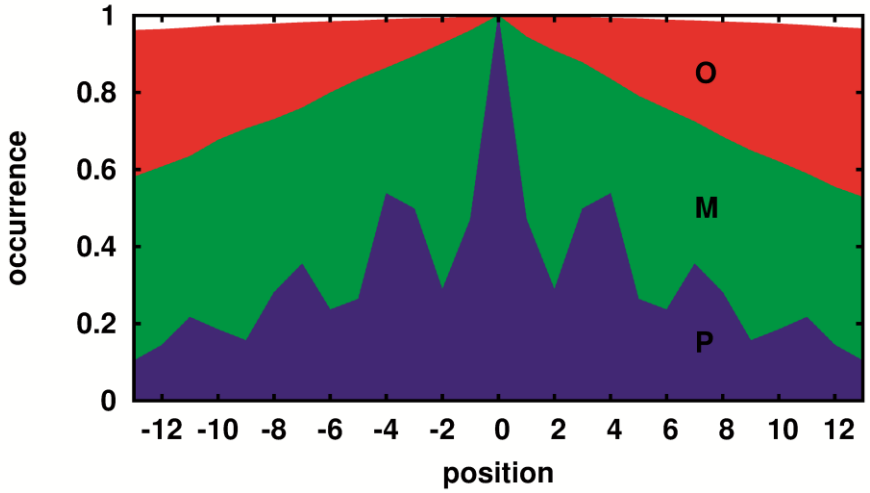

Figure 4.5 Distribution of neighboring residues around a central pore-lining residue in the PH90 set.

4.3.2 Training scheme for imbalanced data

Our PH90 set is an imbalanced data set that comprises 9.38% of positive data (3460 PLRs) and 90.62% of negative entries (33425 non-PLRs). In this study, we used the radial basis function kernel for all SVM trainings. Training an SVM on the entire PH90 set by either equal costs or weighted costs for two classes resulted in high overall prediction accuracy above 90%, a nearly perfect specificity (0.99), but a low sensitivity (about 0.3). It showed an obvious bias toward non-PLRs and is only slightly better than the naïve prediction that assigns all data points to the non-PLR class. Such a classification clearly defeats the purpose of this study which puts a higher emphasis on identifying PLRs than non-PLRs. To deal with this issue, we adopted a multiple independent random sampling and training scheme shown in Figure 4.6. An SVM was trained on all PLRs and a randomly sampled equal number of non-PLRs. Then the predictive power of the trained model was evaluated on the entire data set. Training of the random sampling data was performed 50 times and the model with highest MCC value was then taken as the model of the configuration (features and parameters) for the subsequent prediction process. The idea of this approach is to find a balanced and most representative subset of non-PLRs by multiple random sampling, training and evaluation. With this training approach, the MCC and sensitivity of classification of the imbalanced data was substantially improved whereas some sacrifice of specificity is unavoidable.

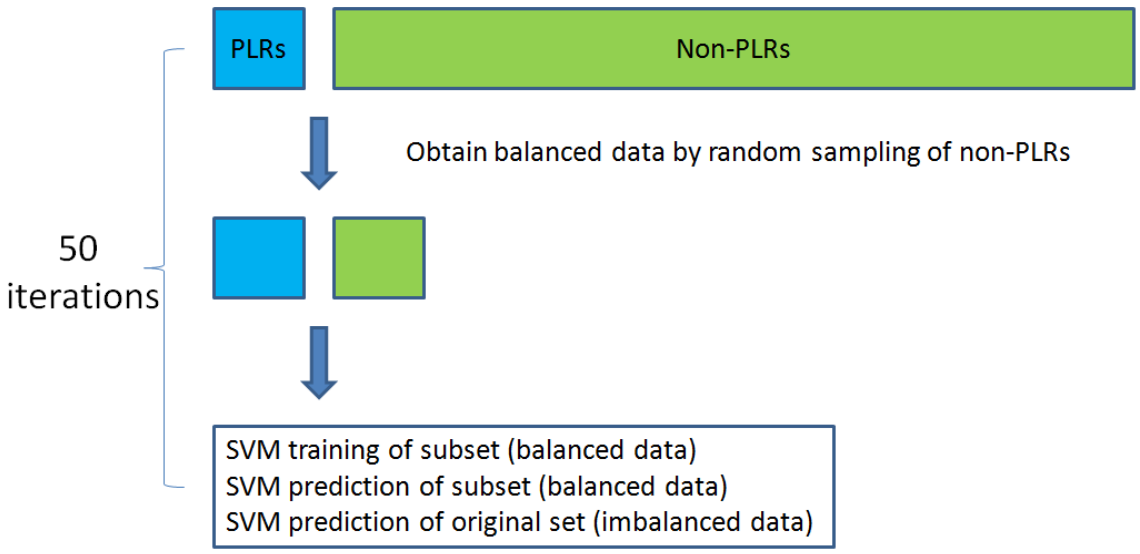

Figure 4.6 Multiple independent random sampling and training scheme for imbalanced data.

4.3.3 Feature and window size selection

The PSSM obtained from PSI-BLAST is a frequently used feature in protein structure prediction (see Section 1.3). To test whether additional features such as physicochemical properties and conservation score improve the prediction of PLRs, we performed SVM model training for different window sizes varied over a large range and different combinations of additional features and PSSM. Default values of two parameters in LIBSVM (gamma and cost) were used in these test runs. Since the parameters used here may not be the most optimal ones, we would rather like to compare the overall performance of a profile than a single window size to determine the best combination of features. Besides the PLRs annotated by PoreID, we also tested the case that includes directly adjacent residues. This strategy was suggested by Nugent and Jones to balance the amount of positive and negative data and to indirectly account for conformational dynamics that cannot be captured in crystal structures.

[13] We termed this PLR-annotation extended set as “PH90ext” to be distinguished from the original PH90 set. The PH90ext set contains 6413 PLRs and 30472 non-PLRs. In Figure 4.7, PH90ext has higher MCC values than those of PH90 under all combinations of features and window sizes. However, sensitivity and specificity are similar in both cases. This means that the performances are likewise similar for both cases. Because the number of positive data (PLRs) in PH90ext is almost twice as much as in PH90, the higher MCC in PH90ext is mainly due to the relatively higher ratio between true-positive predictions and all residues. In both datasets, MCC values increased quickly for longer windows up to a window size of 13 where they started to gradually converge. Figure 4.7 shows that adding the conservation score computed by Rate4Site substantially improves the performance of the SVM predictor for the PH90 and PH90ext sets. This is likely the case because Rate4Site considers the relationship between the homologous sequences, which is not captured by PSSMs. Rate4Site builds a phylogenetic tree for homologous sequences and estimates the evolutionary rates of each position from the topology and branch lengths of the phylogenetic tree. On the other hand, the three physicochemical properties that were used at the same time to train the SVM models showed no clear effect for PH90 (Figure 4.7A) and only a slight improvement for PH90ext. This suggests that the PSSMs already contain most information about these three properties in an implicit way. The configurations with best MCC were then taken to be further optimized for the gamma and cost parameters. For the PH90 set, the best configuration is a window size of 23 with the combinations of PSSM and conservation score. For the PH90ext case, a smaller window size of 19 with all features including PSSM, conservation scores and physicochemical properties was used as the start for the optimization of SVM parameters.

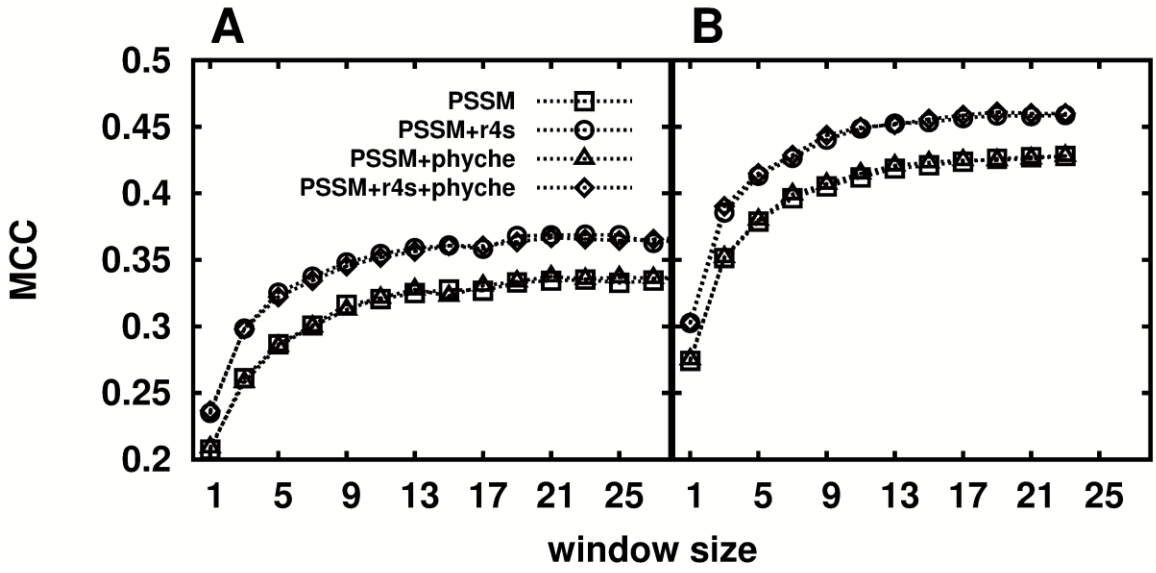

Figure 4.7 Performance of (A) the PH90 set and (B) the PH90ext set under different combinations of features and for different window sizes. "r4s" stands for conservation scores generated by Rate4Site. "phyche" is used as abbreviation for physicochemical properties, which comprises of hydropathy, polarity and flexibility in this study.

4.3.4 Grid search and cross-validation of optimal SVM training parameters

After determining the best window size and features of the training data, we optimized the two parameters of SVMs, gamma and cost. Gamma is an adjustable parameter of the radial basis function kernel (RBF kernel) that was used for mapping the feature space in the SVM. The radial basis function is defined as exp(-gamma*$|u - v|^2$) where u and v are two feature vectors. Gamma can be set as a positive value no greater than 1. The cost is the penalty of misclassification for each data point in the SVM. For both the PH90 and the PH90ext sets, we performed a wide ranged grid search for both parameters. The results of the training evaluations (MCC values) are shown in Figure 4.8. The results for the two sets follow a similar

trend. Larger gamma with higher cost leads to better performances. The least successful separation of training data happened when gamma was set between 2^{-4} and 1 with cost between 2^{-10} and 2^{-2}. In contrast, the red areas show nearly perfect classification when gamma is 2^{-2} and cost is larger than 1. Although Figure 4.8 shows the training accuracies of the entire data set, we then estimated the true performance of the predictor by sixfold cross-validation. The frame areas in Figure 4.8 mark the (gamma, cost) pairs with MCC larger than 0.5 for the PH90 set and larger than 0.6 for the PH90ext set. These pairs were further evaluated by sixfold cross-validation. (Note that the MCC values after sixfold cross-validations are not shown in Figure 4.8.) For both data sets, we found that the averaged MCCs for the previously perfect classification area of entire data set (red areas in Figure 4.8) are only around 0.15 after sixfold cross-validation. This means that the model over-fitted the features of the training data and this model could not accurately predict cases whose features deviate from those of the training data. For the PH90 set, the best prediction emerged when (gamma, cost) were set to (2^{-5}, 2^{8}). The averaged per residue level MCC is 0.39, accuracy is 0.78, sensitivity is 0.82 and specificity is 0.77. For the PH90ext set, the best averaged MCC of sixfold cross-validation was obtained for (2^{-5}, 2^{6}) with averaged per residue level MCC, accuracy, sensitivity and specificity as 0.49, 0.78, 0.83 and 0.77, respectively. As mentioned in the previous section, the prediction performances of both data sets are similar because of close accuracy, sensitivity and specificity. The higher MCC of the PH90ext set is due to its higher true positive ratio. Besides, we then checked the classification of the entire data sets by a SVM with optimized parameters. We found that the percentage of the false positives that are due to type M residues being classed as type P is about 74% in both PH90 and PH90ext (about 24% due to O-type residues and 2% from N).

Finding more specific and representative features to distinguish these two types of residues (P and M) remains a tough challenge for future work.

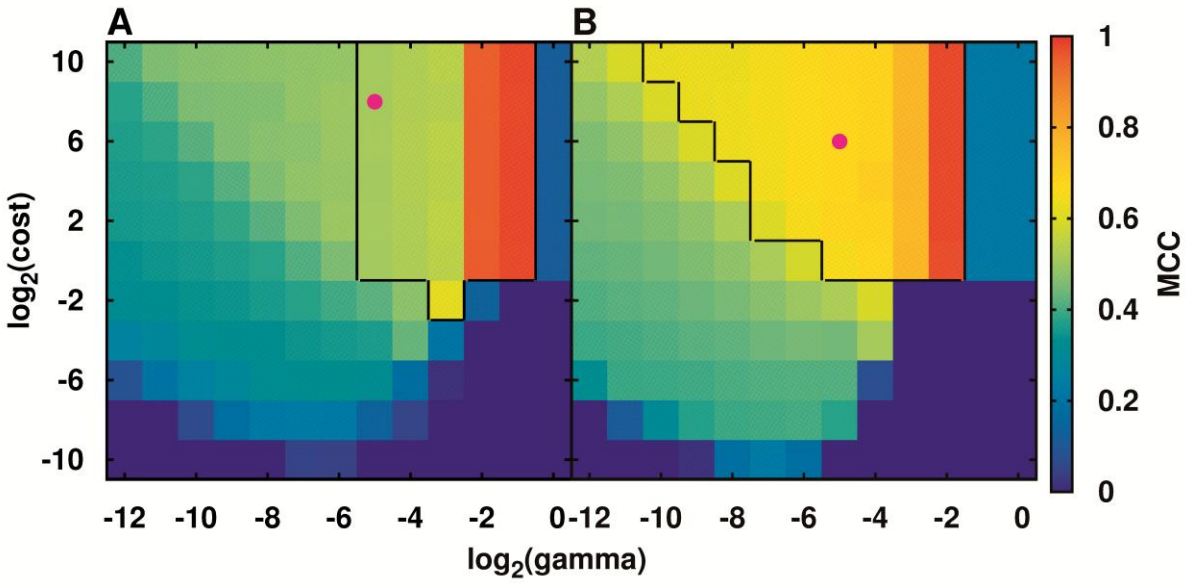

Figure 4.8 Grid search of gamma and cost parameters used in SVM training for (A) PH90 set and (B) PH90ext set. MCC values indicate the performance of an SVM that was trained on the entire data set. For the (gamma, cost) pairs in the framed areas, performance was also evaluated by sixfold cross-validation. Those MCC values after sixfold cross-validations are not shown here except for the magenta points that show the optimized parameters with highest averaged MCCs after cross-validation.

For imbalanced data as was used here, the specificity of 0.77 obtained by cross-validation contains a considerable fraction of false positives compared to the true positives. Thus, we introduced a confidence index for each positive prediction by converting the probabilistic output provided by LIBSVM into a discrete scale from 5 to 9. This mapping of an SVM output into a probability by a sigmoid function was proposed by Platt. [86] Lin *et al*. [87] then improved the algorithm and implemented it in LIBSVM.. As shown in Figure 4.9, when the threshold of

positive prediction was set to tighter values according to the confidence index, accuracy and specificity gradually increased whereas the sensitivity sharply decreased. For the PH90 set with a threshold of 7, MCC was slightly raised to 0.41 with sensitivity as 0.61 and specificity as 0.89. For the same threshold and the PH90ext set, MCC, sensitivity and specificity were 0.48, 0.63 and 0.88, respectively. Generally, a larger confidence index implies a higher ratio of true positives to false positive predictions and provides more reliable predictions for experimental biologists whose studies are related to PLRs.

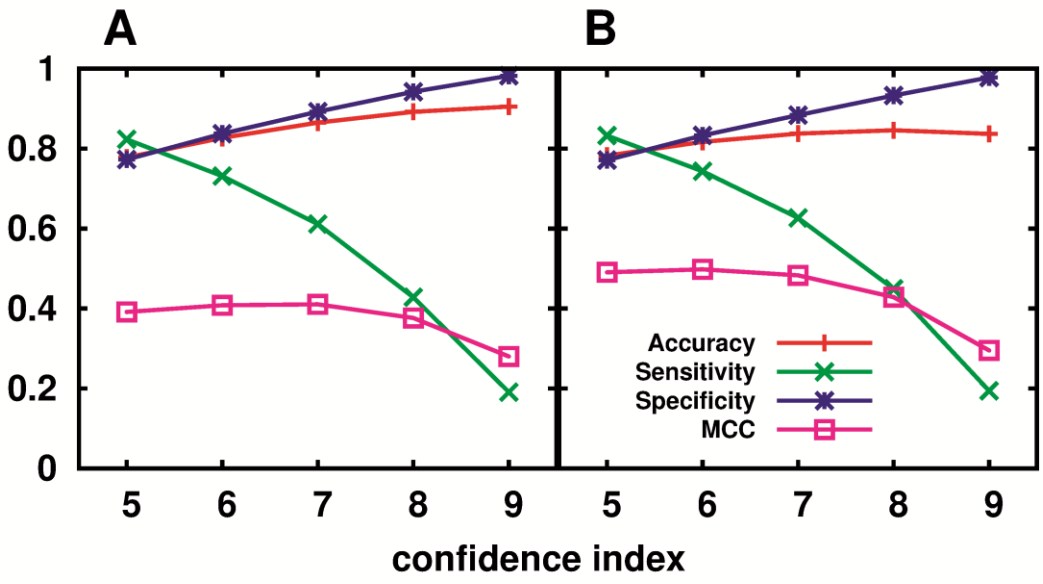

Figure 4.9 The relationship between confidence index and averaged performance of prediction in sixfold cross-validation. (A) PH90 set, (B) PH90ext set.

4.3.5 Evaluation on novel protein structures

To evaluate the performance of our predictor on a set of novel protein structures that were not used during the training of the method, we collected a test set including only structures that were submitted to the PDB databank after composing the PH90 set. We compared two

lists of α-helical transmembrane proteins obtained from PDBTM [22] either on October 19th, 2012, or on July 26th, 2013, and processed the novel protein sequences and structures by the same procedure mentioned in "Materials and Methods" section of this chapter. This resulted in 23 structures that are only contained in the latter list and that share less than 25% sequence identity with any member of the training set. They contain 10251 residues with 693 assigned as PLR and 9558 as non-PLR (the ratio is 1:13.8). We named this test set Test23 (details shown in Table A.2 of Appendix) and evaluated the predictive power of our PRIMSIPLR method on this set. When the threshold of confidence score was set to 7, we obtained a similar performance to the cross-validation results reported before with higher sensitivity and slightly lower specificity and MCC value (Table 4.1).

To have a fair comparison with MEMSAT-SVM, we derived the modified Test23ext set from Test23 by extending the PLR labels to directly adjacent residues, as was done for the PH90ext set. The same threshold of confidence score was applied when testing our model on the Test23ext set. According to Tables 4.1 and 4.2, the optimized models of PRIMSIPLR showed equivalent performance for both Test23 and Test23ext. When considering the set of residues predicted as PLR (TPs + FPs), about one in four predicted PLR residues is correct for Test23. When including next-neighbors of PLRs as is done in Test23ext, about one in two predicted PLRs is correct (TP). In addition, our method outperformed MEMSAT-SVM and yielded more than 10% improved sensitivity and MCC. When considering the performance of each individual protein, PRIMSIPLR has better MCC values than MEMSAT-SVM for 19 out of 23 proteins. One of the cases where MEMSAT-SVM made better predictions than PRIMSIPLR is the protein Magnesium transporter (PDB ID: 4EV6) that was not included in their training set. Remarkably,

MEMSAT-SVM gave a nearly perfect prediction with a MCC value of 0.94 compared to 0.65 with PRIMSIPLR.

Table 4.1 Performance of PRIMSIPLR evaluated on the Test23 set

	TP	TN	FP	FN	accuracy	sensitivity	specificity	MCC
PRIMSIPLR	469	7862	1221	216	0.85	0.68	0.87	0.37

TP, TN, FP, FN stand for true positive, true negative, false positive and false negative. MCC is the abbreviation of Matthews's correlation coefficient.

Table 4.2 Performance of PRIMSIPLR and MEMSAT-SVM evaluated on the Test23ext set

	TP	TN	FP	FN	accuracy	sensitivity	specificity	MCC
PRIMSIPLR	962	7358	1079	461	0.84	0.68	0.87	0.48
MEMSAT-SVM	791	7187	1250	632	0.81	0.56	0.85	0.35

TP, TN, FP, FN stand for true positive, true negative, false positive and false negative. MCC is the abbreviation of Matthews's correlation coefficient.

5 Coarse-grained Brownian dynamics simulations of protein translocation through nanopores

This work had been published as “Lee PH, Helms V, Geyer T: **Coarse-grained Brownian dynamics simulations of protein translocation through nanopores**. *J Chem Phys* 2012, **137**(14):145105”. My own contribution of this work was to conduct all simulations of this project, to analyze the simulation data and to write the manuscript for publication.

5.1 Background

Porous membranes have a wide range of applications in medicine, biology, or physics, where they are used, for example, for filtration, liquid phase separation, biosensing, or drug delivery. [88] As an application for the translocation of proteins through artificial membranes, porous membrane filtration is being used extensively for protein separation in dairy industry. [89] In that case, microorganisms are filtered out due to their large size compared to proteins and pore diameters. Different types of proteins can also be separated by adjusting the pH value of the solution or by modifications of the surfaces of the pores in the membrane. Some recent applications of protein translocation are biosensing or single molecule analysis. [90] The perturbations of the current through a single nanopore within an electrochemical cell exhibit distinguishable patterns for different protein types that are correlated with both charge and size of the proteins.

In biological cells, a very important process is the translocation of newly synthesized proteins across cell membranes via the SecYEG and Sec61 translocons. [5] This biological process

includes the recognition and delivery of the nascent protein chain as well as energy coupling, and requires several molecular machines. [91] Ongoing technological improvements now allow the fabrication of artificial porous membranes with similarly small pore diameters of a few nanometers as the translocon system. [92] For example, free standing solid state artificial membranes can nowadays be produced with a thickness down to 15 nm and average pore diameters from 5 to 25 nm. The properties of the membrane surface can further be changed by chemical modifications. Thus, artificial membranes can act as simple and more stable models for biological membranes containing specialized translocation pores.

Actually, polymer translocation is of fundamental interest in polymer physics and chemistry and has thus been intensely studied in the fields of statistical and chemical physics. For example, several theoretical, experimental, and simulation studies have addressed the translocation of DNA strands through nanopores. [93-95] In these studies, the length of the DNA molecule was much longer than the membrane thickness, while the diameters of the pore and of the DNA fragment were of comparable size. Thus, the relatively stiff DNA molecules could pass the pore only in a linear way. It was found that the translocation time τ is related to the length of the DNA fragment L by a power law dependence $\tau \approx L^{\alpha}$, where different values of the exponent α were observed for different setups. The scaling behavior is also influenced by the hydrodynamic drag resulting from the external driving force and the length of the DNA molecule.

Another focus has been driven polymer translocation, where the driving force can be due to an applied external electric field [96, 97] or due to the binding of chaperone proteins on the *cis*-side that prevent the back-sliding of the translocating protein. [98] Most work has concentrated on long, disordered polymers. The scaling of their translocation dynamics as a function of the

chain length for pores of different lengths and diameters was studied by analytical models [99, 100] as well as by Langevin dynamics simulations similar to the ones reported here. [101, 102] In contrast, fewer works have addressed the translocation of protein-like particles through nanopores. Compared to DNA polymers, protein molecules are relatively small with respect to the dimensions of currently available artificial nanopores and thus can translocate both in their folded and unfolded states, as well as in charged or neutral forms depending on the solution conditions. For example, molecular simulations using an HP-model have studied the orientation and conformational stability of idealized peptides inside of a pore [103] and their diffusional properties. [104] In a pioneering study, Moussavi-Baygi and colleagues employed Brownian dynamics simulations to study the passage of a decorated cargo particle through a coarse-grained model of the nuclear pore complex. [105]

To further investigate the behavior of proteins permeating through porous membranes, we present here results from coarse-grained Langevin dynamics simulations of model proteins translocating across a porous membrane. The analysis of these simulations shows and explains how the translocation rate is influenced by the characteristics of the model system. The investigated parameters include the folding state of the proteins, the pore dimensions, and the interaction between the proteins. This study is organized as follows. First, the protein models and the propagation algorithm used in the simulations are described. Then, the system setup and the corresponding results are presented and we discuss how the diffusion behavior of the proteins is influenced by the characteristics of the proteins and the porous membrane.

5.2 Materials and Methods

For the pore translocation simulations of this project, we used a setup that resembled a typical experimental scenario where a nano-porous silica membrane separates two reservoirs that hold buffer solution with different densities of a water soluble protein. For convenience, such experiments often utilize the small electron carrier cytochrome *c* that can easily be detected by light absorption in the visible regime, can be purchased, is relatively spherical and stable, and does not agglomerate. Its biological function is to transfer electrons in the respiratory or photosynthetic system from the cytochrome *bc*1 complex to the cytochrome *c* oxidase or the reaction center, respectively.

We used a number of different representations for this protein. The most simple one consisted of a spherical bead with a radius of 1.66 nm corresponding to the hydrodynamic radius of the folded cytochrome *c* and the respective, experimentally determined diffusion coefficient of D = 1.48 × 10^5 nm^2/ps. [106] These particles were termed “Normal” for normal sized particle. To investigate particle size effects, also larger model proteins were used with a three-fold increased radius, labeled “Triple” in the following, and “Tiny” particles with a thousand-fold smaller radius. For these differently sized particles, however, the same diffusion coefficient as for the “Normal” particles was used to make the obtained flow rates directly comparable. This, of course, is not the scaling behavior that one would have in a real experiment. However, in this way the time that the proteins need to diffuse through a pore of a given length is independent of the protein size so that the fluxes can be directly compared without rescaling. The actual numbers for any experimental system can then be obtained by rescaling time and lengths. Furthermore, as we will see further down, the pores are

characterized by their aspect ratio, i.e., the ratio between their length and their radius. Rescaling only the protein size thus allows us to directly compare the size effects in pores of different aspect ratios.

Besides the 1-bead model, bead-spring polymers both for the folded and for the unfolded states of the protein cytochrome *c* were implemented, too (see Figure 5.1). We set up a 2-bead model of the unfolded protein and 6-bead models of the folded and the unfolded cytochrome *c*. The diffusion coefficients of the individual beads were adjusted to again obtain the same center-of-mass diffusion coefficient as the 1-bead models, both for the folded and for the unfolded representations. Hydrodynamic interactions between the beads were neglected because for the diffusive long-range transport investigated here only the static shape and overall orientation of the bead chains is important, but not the short-time internal dynamics or the coupling between translation and rotation. [107, 108] The 6-bead polymer model of the folded proteins was held in shape by additional springs such that its radius of gyration R_g resembled the radius of the 1-bead model. For the unfolded models, the springs were tuned such that in the simulations their radius of gyration corresponded to the experimental value of R_g of about 3.0 nm, obtained in cytochrome *c* denaturation experiments. [109] With a spring constant of 800 kJ/(mol nm^2), the best agreement was obtained when the springs between the beads were set to 5.96 and 2.85 nm for the 2-bead and 6-bead unfolded models, respectively. These multi-bead models were pre-equilibrated from an initially extended conformation and a random snapshot was used as a template for the particle insertion interface (see below).

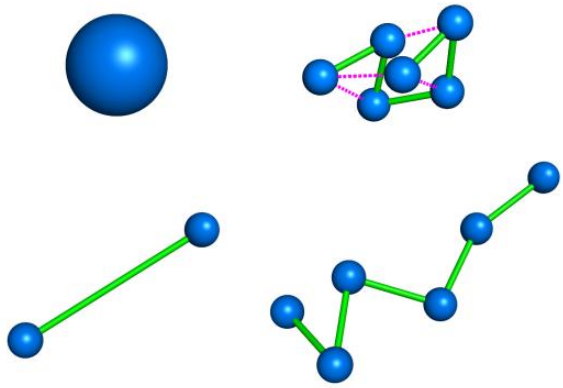

Figure 5.1 Multi-bead protein models. The first row shows the one- and six-bead models for a folded cytochrome *c*. The green sticks represent the springs of the bead-spring polymer, while the dashed magenta lines indicate additional harmonic bonds that keep the polymers in their folded states. The second row gives examples in two different representations with two and six beads, respectively, of their unfolded counterparts.

The effective short-range repulsion between the individual beads was modeled with a Lennard-Jones potential as given in Eq. 5.1, where r is the distance between the surfaces of two beads. The constant x_0 specifies the length scale of the potential, while dr is set such that E_{LJ} = 1 k_BT for r = 0, i.e., when the spheres touch. For the interactions between the proteins and the membrane, the same potential was used. In this case, however, r is the distance between the surfaces of the protein and of the membrane.

$$E_{LJ} = C_{12}\left(\frac{x_0}{r + dr}\right)^{12} \tag{5.1}$$

For the bead-spring models, the beads had a radius of 0.55 nm and the parameters were C_{12} = 9 kJ/mol and x_0 = 3.0 nm for all protein models, requiring dr = 3.34 nm. The total charge of +7.5 e of horse-heart cytochrome *c* was modeled with a single centered charge on the 1-bead models [106], whereas this charge was distributed equally among the beads for the multi-bead models. The repulsion between these point charges was described by a shielded Coulomb

interaction according to Debye-Hückel theory [110] with a Debye length κ = 1 nm corresponding to physiological ion strengths (Eq. 5.2).

$$E_{es} = \frac{q_i q_j}{4\pi\epsilon\epsilon_0 \left(1 + \kappa \frac{B_{ij}}{2}\right)^2 r_{ij}} exp\left[-\kappa\left(r_{ij} - B_{ij}\right)\right] \tag{5.2}$$

Here, q_i and q_j are point charges on beads i and j, and r_{ij} is the distance between these two charges. B_{ij} is the sum of the burial depths b_i and b_j of the point charges in their respective beads, which were set to the respective bead radii. κ is the inverse Debye length, and ϵ_0 and ϵ are the absolute and relative dielectric constants for vacuum and aqueous solution, respectively. To connect the beads of the multi-bead models, we used harmonic springs (Eq. 5.3) with a spring constant k_{ij} and a rest length L_{ij}, while the distance between the centers of the beads i and j is denoted by R_{ij}.

$$E_{spring} = \frac{k_{ij}^2}{2}\left(R_{ij} - L_{ij}\right)^2 \tag{5.3}$$

Apart from the bond lengths that determine how large the multi-bead protein models are on average, the actual values of the interaction parameters are not crucial. We ran tests with different short-ranged interaction strengths, spring constants, or with and without charges on the beads and always obtained very similar results. The used parameter values are a compromise between numerical efficiency at soft interactions and an accurate definition of radii and pore sizes when hard interactions were used.

To propagate the particles, we used the Langevin propagator recently introduced by Winter and Geyer. [111] In this implicit-solvent approach, a particle of mass m experiences three types of forces, namely the external force(s), random kicks, and a friction force, which together determine how the particle velocity v changes.

$$m\frac{dv}{dt} = f_{ext} + f_r - \gamma v \tag{5.4}$$

In Eq. 5.4, the external force term f_{ext} combines the short range, electrostatic, and harmonic interactions between the beads and between the beads and the membrane as described above, whereas the other two contributions are effective interactions that mimic the (implicit) solvent molecules. [112] The random kicks f_r due to the thermal motion of the solvent molecules are distributed according to Eq. 5.5, where k_B is the Boltzmann constant and δ is the Dirac delta function. [112] The strength of the random kicks is related to the friction coefficient γ of the particles. For the friction term, which describes how the solvent molecules are pushed away by the moving protein, a linear Stokesian form is used.

$$\langle f_r(t)\rangle = 0 \; and \; \langle f_r(t) f_r(t')\rangle = 2\gamma k_B T \delta(t - t') \tag{5.5}$$

For the derivation of the Langevin propagator, it is more convenient to express the random kicks in terms of forces [111] instead of the usually given displacements [113]. The Langevin propagator is then obtained by an analytical integration of the Langevin equation Eq. 5.4 under the assumption that the total force $F = f_{ext} + f_r$ is constant over a short time interval Δt. This

yields the velocity $v(\Delta t)$ and the displacement $\Delta x(\Delta t)$ after one time step Δt when the particle had the velocity v_0 at the beginning of the time step.

$$v(\Delta t) = \frac{F}{\gamma} + \left(v_0 - \frac{F}{\gamma}\right) exp\left[-\frac{\gamma \Delta t}{m}\right] \tag{5.6a}$$

$$\Delta x(\Delta t) = \frac{F}{\gamma}\Delta t - \frac{m}{\gamma}\left(\frac{F}{\gamma} - v_0\right)\left(1 - exp\left[-\frac{\gamma \Delta t}{m}\right]\right) \tag{5.6b}$$

As the simulations shown here only contained large enough particles of only one size per run, we could have also used the conventional Brownian dynamics algorithm that is obtained from Eq. 5.6b in the limit of a vanishing velocity relaxation time $\tau = m/\gamma$. The Langevin dynamics (LD) algorithm, however, is numerically not more expensive but allows running the simulations with larger integration time steps, because the trajectories are smoothed due to the inertia of the particles.

The simulation box was constructed as shown in Figure 5.2. Seen from the top, the box is quadratic with a side length of 80 nm. The pore-containing membrane is placed at half of the height of the box. The thickness of the membrane, and thus the length of the pores, was varied from 5 to 40 nm with pore radii of 4, 8, or 16 nm. The heights of the *cis* and *trans* compartments were 30 nm each. At bulk densities similar to typical experimental conditions, the simulation volume contained only a handful of particles. In all simulations, the distances between the pores or between the pores and the side walls of the simulation box were much longer than the Debye length of about 1 nm or the range of the effective short range interactions. Consequently, using periodic boundary conditions was not necessary as tests

showed. The large volumes of the *cis* and *trans* compartments of an experiment above and below the membrane were mimicked with our constant density interfacing algorithm [114] at the top and the bottom walls of the simulation box. This interfacing algorithm is based on the idea of a virtual boundary, placed in the middle of a volume with Brownian particles, where the probability of a particle passing through this virtual boundary at a given density yields the probability for injecting particles into the simulation box at a real wall. In the other direction, particles that jump across the wall from the inside are removed from the simulation. In this way, the particles in the simulation behave as if the simulation box was extended beyond the wall. With such particle injection/removal interfaces at the top and at the bottom boundaries of the simulation box, the concentration gradient between the *cis* and the *trans* compartments could be kept constant even with our relatively small simulation volumes above and below the membrane. This resulted in large savings of computational effort while still allowing for good statistics. A typical simulation contained 10 to 20 particles in the simulation volume on average at one time, but the simulations were usually run until about 1000 particles had translocated through the pores all the way from the *cis* to the *trans* side of the simulation box.

The simulations were started with an initially empty simulation box, into which particles were inserted from the constant density boundary interfaces. The analysis of the simulations was started after the density had equilibrated throughout the simulation volume.

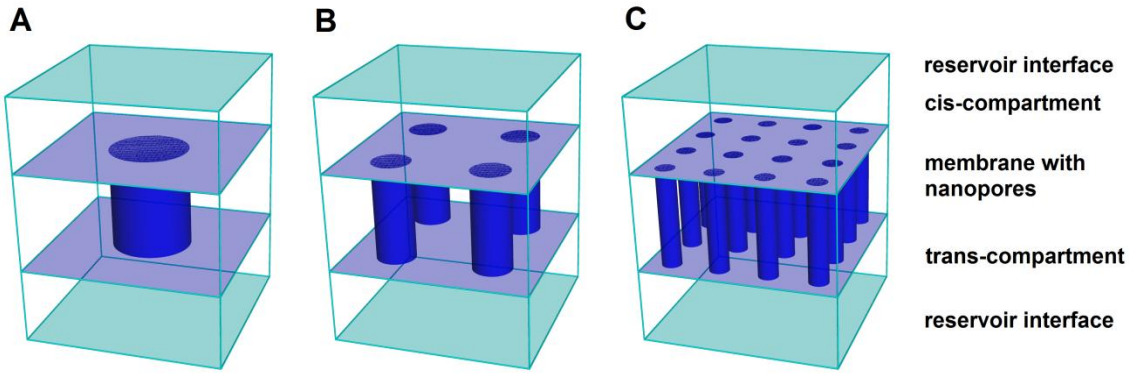

Figure 5.2 Cartoon representations of simulation boxes. At the reservoir interfaces, constant particle densities were set to model the large bulk volumes of an experiment. Usually, the concentration at the *cis* side was higher such that the resultant diffusive flow of particles developed towards the *trans* side, where the proteins were "swallowed up" and counted at the *trans* side reservoir interface. The figure shows three setups with (A) a single pore with a radius of 16 nm in the membrane, (B) four pores of radius 8 nm, and (C) 16 pores of radius 4 nm. In all cases, the membrane area was 80 × 80 nm^2, the heights of the *cis* and *trans* compartments were 30 nm, and, in this figure, the membrane had a thickness of 40 nm.

5.3 Results and Discussion

5.3.1 Verifying the linear regime

In a macroscopic picture of non-interacting, freely diffusing particles in the above described system of a porous membrane between two reservoirs, the particle flow rate from one reservoir to the other linearly depends on the concentration in this reservoir. The flow rate Φ_{ct} from the *cis* to the *trans* reservoir, e.g., then depends on the concentration c_{cis} in the *cis* compartment as $\Phi_{ct} = a\, c_{cis}$, where the rate a combines the physical parameters such as the diffusion coefficient of the particles, the number and area of the pores, and the interaction between particles and pores. For a symmetric setup as investigated here, this parameter is the

same on both sides of the membrane. An analogous relation exists for the reverse flow rate Φ_{tc}. The resulting change of the concentration in the *trans* compartment, c_{trans}, then also depends on its volume V_{trans}.

$$\frac{dc_{trans}}{dt} = \frac{1}{V_{trans}}(ac_{cis} - ac_{trans}) \tag{5.7}$$

With the average concentration c_0 throughout the complete volume $V_0 = (V_{cis} + V_{trans})$, we can write

$$c_{cis} = \frac{V_0}{V_{cis}}c_0 - \frac{V_{trans}}{V_{cis}}c_{trans} \tag{5.8}$$

and obtain the following differential equation for c_{trans}:

$$\frac{dc_{trans}}{dt} = \frac{aV_0}{V_{cis}V_{trans}}(c_0 - c_{trans}) \tag{5.9}$$

which describes how the concentration equilibrates exponentially towards c_0 from $c_{trans}(t = 0) = 0$:

$$c_{trans}(t) = c_0\left(1 - exp\left[-\frac{aV_0}{V_{cis}V_{trans}}t\right]\right) \tag{5.10}$$

The interesting quantity is the pore-translocation rate a that depends non-trivially on the pore and on the particle properties when the particle diameters are comparable to the pore dimensions as we will show in the following.

When deriving the exponential relaxation curve for $c_{trans}(t)$, we assumed that the diffusive flow rate through the pores scales linearly with the concentration difference across the membrane and that the net translocation is the sum of the forward and backward flows. Under these conditions, the complete relaxation behavior of a given pore setup can be extrapolated from the simulated translocation rate at a single density difference. With finite sized particle, however, crowding will occur at higher densities and the relation between concentration difference and translocation rate will become nonlinear. To verify that our simulation parameters are within the linear, non-crowding regime, we ran simulations with a fixed pore setup and varying combinations of concentrations on the *cis* and the *trans* sides. In these tests, the membrane of 20 nm thickness contained a single pore of 16 nm radius.

To determine the flow rate Φ_{ct} from the *cis* to the *trans* reservoir, we monitored the trajectories of those particles that were inserted into the simulation at the constant density interface of the *cis* side and counted how many of those were taken out of the simulation at the *trans* side interface. For the reverse flow rate Φ_{tc}, those particles were counted that were inserted at the *trans* interface and left the simulation volume at the *cis* reservoir interface. The flow rates were then determined from a straight line fit to the respective counts vs. the simulation time to account for the initial equilibration phase and to smoothen out the statistical fluctuations of the counts.

The relaxation of the particle densities was probed at various intermediate concentration ratios (represented as c_{cis}:c_{trans}) of (100:0), (90:10), (80:20), (70:30), (60:40), and (50:50), where 100% corresponded to a bulk protein concentration of 1.5 g/L, which is a typical value in protein translocation experiments [P. Huber, private communication]. This set of simulations can be regarded as the specific conditions at six instantaneous points in a protein translocation experiment. The first point (c_{cis}:c_{trans} = 100:0) corresponds to the initial preparation of the experiment when protein solution is injected into the *cis* compartment while the *trans* compartment contains pure buffer solution. The net flow rate at this time point is the highest and then it gradually slows down over time as the concentration difference relaxes. Finally, the flows from both directions of the membrane become equal so that the net flow vanishes. Shown in Figure 5.3 are the simulated individual flow rates from *cis* to *trans* and from *trans* to *cis* as well as the resulting net flow rates across the membrane for the selected density ratios. The individual flow rates Φ_{ct} and Φ_{tc} show a linear dependence on the particle concentration of the departure reservoir. By balancing each anti-flow pair, we found that also the net flow rate Φ_{net} is linearly proportional to the concentration difference between the reservoirs as expected. The results confirm that we can predict the net flow rate of the whole translocation process from simulations at any non-equilibrium concentration ratio. Thus, for simplicity, in all subsequent simulations of the particle translocation in different system setups, the concentration ratio was set to (100:0).

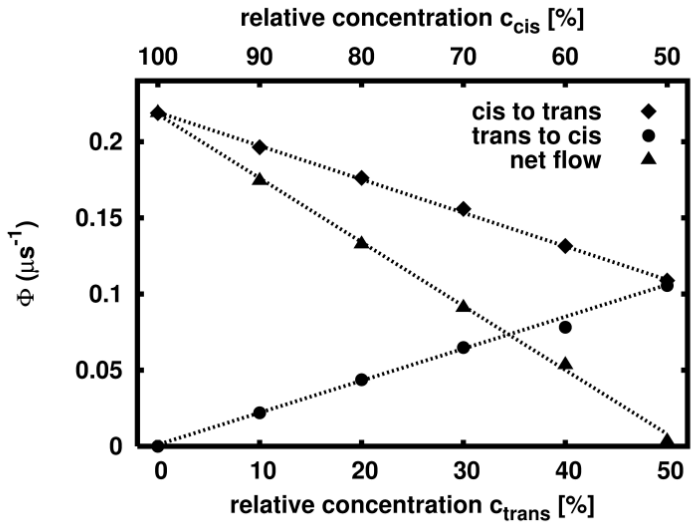

Figure 5.3 Diffusive flow rate vs. concentration gradient obtained from simulations with a single pore of 16 nm radius and 20 nm length for the "Normal" sized model proteins. The flow from *cis* to *trans* counts all particles that were inserted at the *cis* interface and left the simulation on the opposite *trans* interface, while the *trans* to *cis* flow counts translocations in the opposite direction. The net flow is the difference between these two individual flows. The concentrations in the *cis* and *trans* compartments are given in percent of the initial concentration of 1.5 g/L. Dashed lines are linear fits. The statistical errors of all data points are about 3% so that the error bars would be covered by the symbols.

5.3.2 Pore translocation of folded proteins

To investigate how the pore geometry affects the protein translocation rate, we ran simulations with a single pore on the membrane (Figure 5.2A) where the pore length L_{pore} was varied from 5 to 40 nm in 5 nm increments and its radius R_{pore} was set to 4, 8, or 16 nm, respectively. For the diffusing particles, we used the three differently sized single bead models described in Section 5.2. To reduce the complexity of the system, the interactions between the protein particles were ignored such that the simulations were effectively run at vanishing protein concentration.

In the simplest one-dimensional model for the simulated scenario, the bulk densities extend up to the two ends of the pore and a linear gradient develops inside the pore. In that case, the resulting diffusive flow rate decreases inversely proportional to the pore length, and increases with the pore area $A_{pore} = \pi R_{pore}^2$ and the bulk diffusion coefficient D of the particles:

$$\Phi_{\text{simple}} = A_{pore} D \nabla c = A_{pore} \frac{D(c_{cis} - c_{trans})}{L_{pore}} \tag{5.11}$$

Here, A_{pore} is a scaling factor for the overall system size and D determines the time-scale. Obviously, for very short pores (with $L_{pore} \rightarrow 0$), the flow rate diverges. In those cases, one can expect that the three-dimensional transport from the bulk to the pore entrance and away from the pore exit will determine the translocation rate. This is described in a more realistic model that was derived by Brunn *et al.* [115] via a Green's function description. This analytical model gives the stationary particle density and the resulting flow through a single membrane pore in an infinite membrane between two infinite half-space reservoirs. Replacing the dimensionless variables of this model by the variables of our setup yields the following equation for the total flow rate:

$$\Phi_{\text{theo}} = A_{pore} \frac{D(c_{cis} - c_{trans})}{L_{pore} + \frac{\alpha\pi}{2} R_{pore}} \tag{5.12}$$

Here, α is a constant close to unity whose exact value depends on L_{pore}. For our purposes, it can be approximated as $\alpha = 1$ with a negligible overall error in Φ_{theo}. Similar to the simple model of Eq. 5.11, the diffusive flow rate through the pore scales linearly with the pore area, the

concentration difference, and the diffusion coefficient of the particles. The interesting prediction of this explicitly three-dimensional model is that the diffusive flow rate Φ_{theo} decreases not only with the pore length, but that there is an additional radius dependent offset. This effect takes care of the fact that, especially for short pores, the density gradient is not confined to the pore interior but that the particle density above the ends of the pore is affected on a length scale comparable to the pore radius. Consequently, for very short pores, the flow rate scales with A_{pore}/R_{pore}, i.e., it is proportional to the pore radius and not to the pore area. Only for very thin or long pores with $L_{pore} >> R_{pore}$, the region above the pore can be neglected and both models (Eqs. 5.11 and 5.12) give the same flow rate. For wider pores, the three-dimensional model of Brunn *et al.* [115] predicts a smaller finite particle current for short pores and a slower-than-inverse decrease of the flux with the pore length.

To compare the results from our simulations to the theoretical models, we fitted the obtained particle flow rates with the following simplified form of Eq. 5.12 where all scaling factors are summarized into C_1 and a potential length offset is described by $C_2 R_{pore}$.

$$\Phi_{\text{fit}} = \frac{C_1}{L_{pore} + C_2 R_{pore}} \tag{5.13}$$

In the ideal case when the simulations reproduce the theoretical model of Eq. 5.12, C_2 would be close to $\pi/2 \approx 1.57$ whereas any deviation from this value indicates that our simulations with the finite sized particles do not behave exactly like the theoretical continuum model.

The results from the simulations are summarized in Figure 5.4, which shows how the particle flow rate Φ decreases with the pore length for the different pore radii and particle sizes. For

each combination of pore radius and particle size, the decrease of Φ with L_{pore} can be fitted well by Eq. 5.13. The fitting results are given in Table 5.1 together with the values of the expected fit constants from Eq. 5.12. There is no result for the "Triple" particles and the small 4 nm pores, because in this case the particles are larger than the pore opening. The first observation is that the scaling constants C_1 obtained from the simulations are, except for one case, larger than the theoretically expected values. This is interesting insofar as one might expect that Φ is determined by the smaller effective pore area $\pi(R_{pore} - R_{particle})^2$, whereas in our simulations a larger particle radius only slightly reduced the observed flow rate. We note that the analytical model assumes a continuous density what corresponds to many infinitely small particles. In contrast, the particles used in our simulations had radii not much smaller than the pores. For the "Tiny" particles having negligibly small radii compared to the pore radius, the obtained fit parameters and the flow rates are indeed closer to the theoretical results than for the larger particles.

Table 5.1 Fit parameters for the particle flow rates according to Eq. 5.13 for simulations with the three single bead models and pores of various lengths and radii. The theoretical values were obtained from Eq. 5.12 by inserting the parameters of our system setups.

R_{pore}(nm)	16		8		4	
Particle size	C_1	C_2	C_1	C_2	C_1	C_2
Normal	11	1.9	2.8	2.50	0.47	2.5
Tiny	9.8	1.52	2.4	1.80	0.56	1.8
Triple	10.3	2.0	2.15	2.90	N/A	N/A
Theoretical value	8.56	1.57	2.15	1.57	0.54	1.57

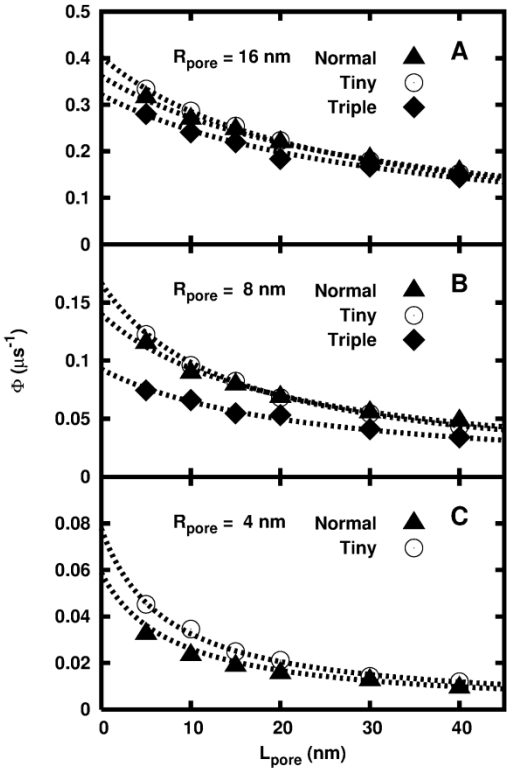

Figure 5.4 Decrease of the diffusive flow rate with the pore length for pores of R_{pore} = 16 nm (A), 8 nm (B), and 4 nm (C), and the three differently sized protein models. The dashed lines are fitting curves according to Eq. 5.13. The statistical errors of all data points are about 3% so that the error bars would be covered by the symbols.

A similar trend is obtained for the effective length correction factor C_2. The values of C_2 are generally larger than predicted from the analytical model and the deviations increase with particle size. This again means that our simulations on the one hand recover the analytical model in the limit of vanishing particle size whereas, on the other hand, they provide evidence for finite-size effects for small pores on biologically relevant length scales.

As mentioned above, one might suspect that the simulation results can be fitted better when the particle size is taken into account via an effective pore radius $R_{eff} = R_{pore} - R_{particle}$ and an effective pore length $L_{eff} = L_{pore} + 2R_{particle}$.We therefore also tried to fit the obtained flow rates with various combinations of the actual and the effective pore radii and lengths. Although the overall results gave a qualitatively similar picture, the parameters were less consistent. These results are therefore not shown here.

5.3.3 Density profiles along the pores

When Brunn and his colleagues derived their analytical continuum solution to the diffusive current [115], they assumed that the concentration profiles were symmetric with respect to the central plane of the membrane. As shown in Figure 5.5, this was not the case in our simulations with the finite sized particles. These concentration profiles were determined from the trajectories by counting the particles in disk shaped bins of radius 11 nm and 3 nm for R_{pore} = 16 nm and R_{pore} = 8 nm, respectively, stacked along the pore axis. We only analyzed these inner parts of the simulation volume so that the inner pore walls and the membrane surface would not affect how the radius had to be chosen at each height. According to Figure 5.5, the density gradients are roughly linear inside the pore and in the two bulk regions, albeit with different slopes. Whereas the analytical theory predicts a continuous and monotonic decrease of the concentration from the *cis* to the *trans* compartment, in our simulations, interestingly, (i) the density around the pore entrance on the *cis* side was increased and (ii) the resulting steeper gradient inside the pore extended slightly beyond the pore exit into the *trans* compartment. These “discontinuities” were smaller for the “Tiny” particles and the wider pores, and more pronounced with the larger “Triple” model proteins or a narrower pore. This clearly shows that the analytical theory, which assumes a continuum of infinitely small non-interacting particles, breaks down when the dimensions of proteins and pore become comparable. In the simulations used for Figure 5.5, the particles did not interact, but also with a repulsive interaction between the particles we got the same density profiles, at least at the low densities used here. Also when the particle-pore potential was changed, the picture did not change qualitatively. This effect must therefore be related to the relative sizes of the proteins and the pores.

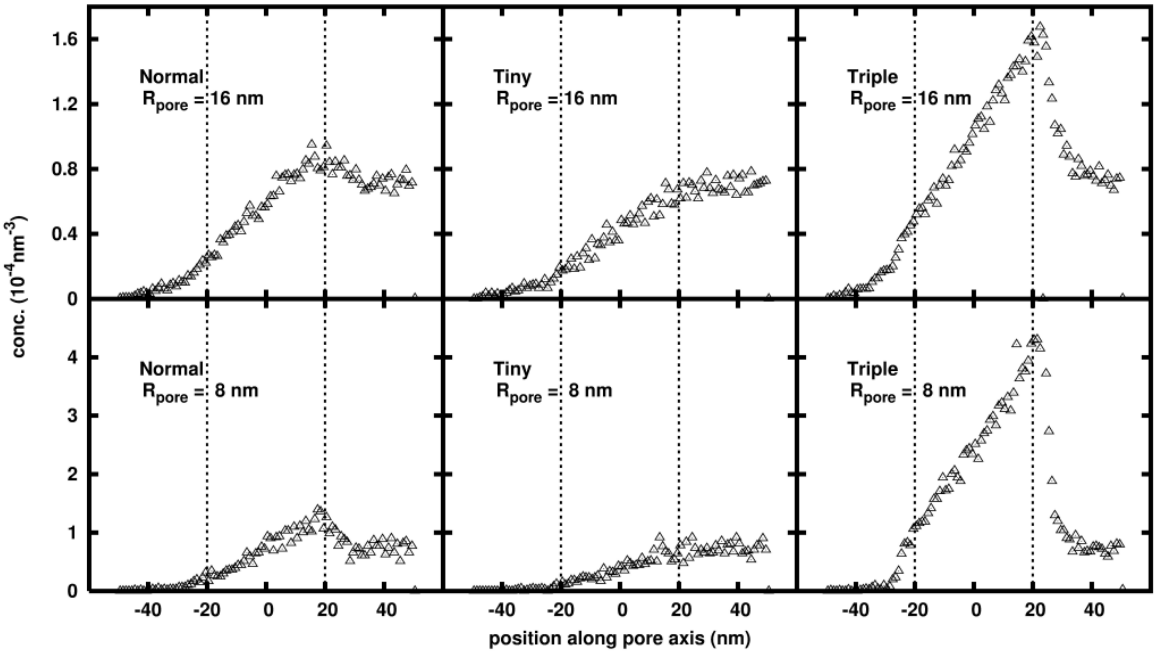

Figure 5.5 Concentration profiles along the pore axis for proteins of different radii and for pores with R_{pore} = 16 nm (upper row) and 8 nm (lower row). In all cases, the pore length was L_{pore} = 40 nm, i.e., the pore extends from –20 to 20 nm, the *cis*-compartment from 20 to 50 nm, and the *trans*-compartment is from –50 to –20 nm. The concentrations were determined within cylindrical volumes with a radius of 11 and 3 nm, respectively, centered along the pore axis.

One possible explanation of this density peak around the pore entrance might be that, even without any attraction between the proteins and the membrane, the model proteins accumulate close to the membrane and then slide into the pore parallel to the membrane. To investigate whether this is the case, we determined the radial density distributions of the proteins perpendicular to the pore axis from the simulations. Figure 5.6 presents the results from a pore of L_{pore} = 40 nm and R_{pore} = 16 nm and the three different particle sizes. Panel (A) indicates that the "Normal" particles started to feel the pore wall when their centers were more than 14 nm away from the pore axis while this occurred for the larger "Triple" particles

already at r = 11 nm [panel (C)]. Only the very small "Tiny" particles could make use of the full pore diameter [panel (A)]. These three density distributions illustrate that the density peak around the pore entrance is not due to an accumulation of the proteins on the membrane surface laterally from the pore entrance, but that the density peak is limited to the volume above the pore entrance. A similar particle accumulation can be observed at the pore exit. Here, the particles may diffuse away from the membrane more easily towards the c_{trans} = 0 interface than on the *cis* side, where the higher bulk density pushes them into the pore. Consequently, the step at the pore exit is less pronounced.

To verify that this is not an artifact of our setup or a bug in the simulation software, we also ran simulations in which the values of c_{cis} and c_{trans} were swapped. From these simulations, we obtained the reversed profiles as expected.

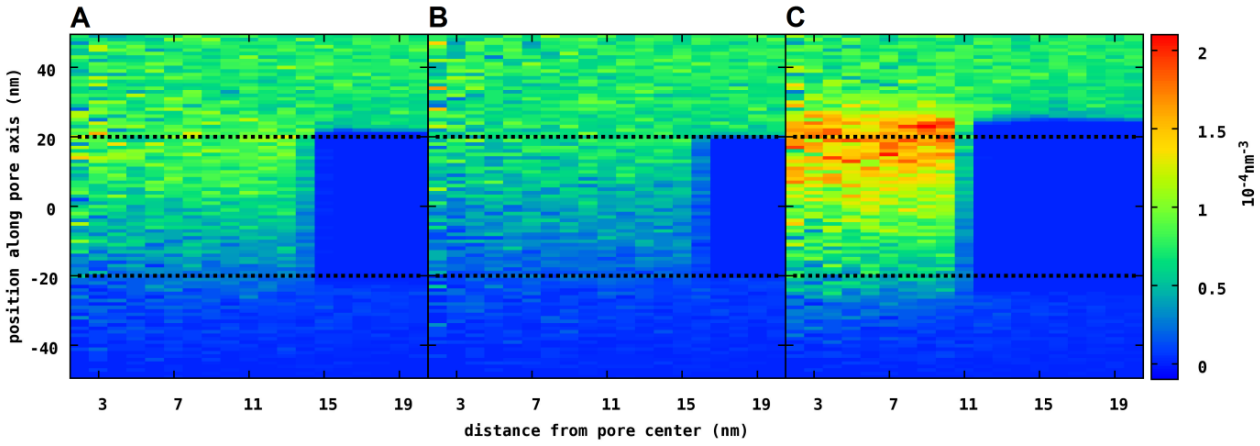

Figure 5.6 Radial distributions of the particle concentrations through the simulation box along the pore axis. Here, the pore again had R_{pore} = 16 nm and L_{pore} = 40 nm. The protein models for the three panels are (A) Normal, (B) Tiny, and (C) Triple. The partition along the pore axis (vertical axis here) is the same as the horizontal axis of Figure 5.5. The small volume of the innermost bins near the pore center resulted in poor statistics, thus the densities are given only for radii greater than 2 nm.

We suspected that the increased particle density could result from a locally reduced diffusion coefficient. To test this hypothesis, we computed the diffusion coefficients of the particles along and perpendicular to the axis of an infinitely long pore. The diffusivity of the particles along the pore axis, which is an unconfined dimension in this setup, also represents the diffusive behavior of the model proteins in the bulk. In an infinite bulk scenario or under periodic boundary conditions, our LD propagation algorithm indeed reproduces the long-time diffusion coefficient that was assigned to the particles. For pores of R_{pore} = 8 and 16 nm radius, we compared the one dimensional diffusion coefficients $D_x = \Delta x^2(\Delta t)/2\Delta t$ along the pore axis to D_{yz} (which is the average of the extracted D_y and D_z) perpendicular to the pore axis for varying observation time intervals Δt. One expects that the confinement will become visible for longer Δt, whereas for short observation intervals, the particles should behave as in the bulk. As shown in Figure 5.7, for short Δt < 100 ps, the particle motion is in the weakly damped, still-ballistic Langevin regime where the diffusivity increases with the length of the observation interval (see Eq. 5.6). For comparison, the velocity relaxation time of the model cytochrome *c* is τ = 1.3 ps and the simulation time step was 1 ps for these tests. When the observation interval was longer than 100 ps, the particle motion was in the overdamped regime so that the diffusion coefficients converged to the long-time bulk value of 1.48×10^{-5} nm^2ps^{-1}. In the unconfined direction along the pore axis, D_x remained constant for arbitrarily long intervals Δt. However, in the confined directions perpendicular to the pore axis, the diffusion coefficient decreased with Δt because the particles bumped into the inner wall of the pore, effectively bouncing back and forth in these directions several times during a long observation interval. This effect sets in around $\Delta t \approx 10^4$ ps for the larger pore with R_{pore} = 16 nm and is more pronounced for the larger

particles (Figure 5.7A) due to the smaller effective radius (see Figure 5.6C). For the smaller pore with R_{pore} = 8 nm, D_{yz} decreased already from $\Delta t \approx 10^3$ ps on Figure 5.7B. In this smaller pore, the particle size makes an even larger difference.

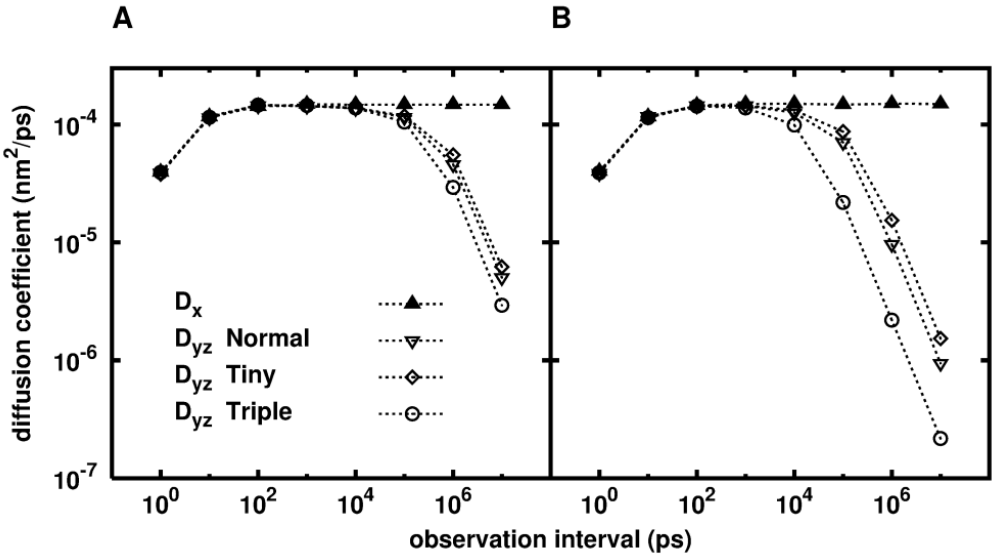

Figure 5.7 Diffusion coefficients of the particles inside infinitely long pores of R_{pore} = 16 nm (A) and R_{pore} = 8 nm (B) vs. the observation time interval. The one-dimensional diffusion coefficient D_x in the direction along the pore axis (filled triangles) is the same for both pore radii and all particle sizes. The other three curves (open symbols) give diffusion coefficients D_{yz} (which are the averages of the measured D_y and D_z) perpendicular to the pore axis for the three differently sized protein models.

For very long observation intervals, the motion of the particles inside the pore can thus be considered an effectively one-dimensional diffusion since the diffusivity in the perpendicular directions approaches zero. On the other hand, the membrane with the pore is still a three-dimensional system with a finite length. Consequently, the local particle densities are still related to the 3D diffusion coefficient $D = (D_x + D_y + D_z)/3$, which with $D_y = D_z = 0$ shrinks to a third of its bulk value in the central regions of the long pores. In other words, when a particle enters the pore from the bulk compartment, the diffusivity in the perpendicular direction is

reduced while the diffusivity along the pore axis remains unchanged. This reduction of the overall diffusion coefficient effectively traps the particle in the pore, and the local concentration increases. This does not occur in the analytical model where the particles are negligibly small compared to the pore.

We thus find that the geometric confinement by the pore leads to a reduced diffusivity of the permeating particles, resulting in an accumulation of particles at the pore entrance. This accumulation, in turn, leads to a steeper concentration gradient along the pore and thus to an increased diffusive flow rate. This effect is more pronounced for larger particles and longer pores. This explains why in our simulation setup the diffusive flow rate decreases not as strong for longer pores as predicted by the analytical theory. Another way to view this behavior is that the pores appear to be wider than they really are.

In our simulations, we had neglected hydrodynamic interactions (HI) between the particles and between the particles and the pores. While the omission of the interparticle HI can be easily justified from the very low densities of less than 20 particles in the complete simulation volume, including the hydrodynamic interactions between the particles and the membrane and its pores would certainly affect the results. While there is a formula available for how the diffusion coefficient changes next to an infinite planar wall [116], no analytical description exists yet for particles inside a cylinder of a finite length. When a particle approaches a fixed wall, its diffusion coefficient is reduced due to the no-slip boundary conditions. Similarly, its mobility inside of a pore is reduced more strongly for narrow pores. Using the same arguments as above, the hydrodynamic interactions between the particles and the membrane surface should reduce the probability to first hit the pore opening, whereas the even lower diffusion

coefficient inside the pore should lead to an even more pronounced accumulation of particles at the pore opening and a corresponding steeper gradient. We expect that this would lead to an even stronger compensation for longer pores. Thus, with HI, we expect that the flow rate is lower for short pores but that the transmission of longer pores is higher relative to the analytical expectations. HI is actually a finite-size effect, too. The diffusion coefficient of a particle above the wall is affected on a length scale of about its diameter. This means that the infinitely small particles of a continuum description can come arbitrarily close to the membrane and the pore walls before they feel any hydrodynamic influence from them.

5.3.4 Translocation of unfolded proteins: Multi-bead models

In this section, we show how the folding state of a protein affects its translocation through a pore. For this, we performed similar Langevin dynamics simulations as above but with bead-spring polymers instead of the single bead models. Again, the membrane had a size of 80 × 80 nm^2 and the pore radii were set to 4, 8, or 16 nm. Both to increase the diffusive flow in the simulations with the narrow pores (e.g., R_{pore} = 4 and 8 nm) and to have the same porosity of the membrane with the different pore sizes, this time we placed 16 pores (arranged as a 4 × 4 grid) into the membrane when R_{pore} was set to 4 nm, 2 × 2 = 4 pores for R_{pore} = 8 nm, and a single pore for R_{pore} = 16 nm. In these simulations, also non-bonded electrostatic and short-ranged repulsive interactions between the particles were included.

The protein translocation rates obtained from the simulations were again fitted against the pore lengths using Eq. 5.13. The respective results are given in Figure 5.8 and Table 5.2. The results with the 1-bead model [panel (A)] are, as expected, very similar to the results from

Section 5.3.2 with the "Normal" sized particles. For the large 16 nm pore, the same fit constants were obtained, whereas for the smaller 8 and 4 nm pores, C_2 was slightly larger due to the inter-particle interactions. The folded 6-bead model has a slightly larger outer radius, correspondingly C_1, that determines the total scaling of the flow rate, is lower than for the 1-bead representation. Interestingly, for the more flexible 6-bead model C_2 is closer to the theoretical value than with the non-deformable 1-bead proteins.

Table 5.2 Fit parameters of Eq. 5.13 for the multi-bead models. In these simulations, the same total pore area was used, but the fit parameter C_1 is normalized per pore for comparison with the single bead results given in Table 5.1.

R_{pore} (nm)	16		8		4	
Particle size	C_1	C_2	C_1	C_2	C_1	C_2
1 bead	11.2	1.9	2.9	3.2	0.47	3.4
6 beads folded	10.0	1.7	2.5	2.7	0.28	2.1
2 beads unfolded	10.0	1.7	1.7	1.6	0.15	0.82
6 beads unfolded	10.4	2.0	1.7	1.8	0.089	0.81

A different picture was obtained for the bead-spring models of the "unfolded" protein that were assigned the same center-of-mass diffusion coefficients as the folded models [panels (C) and (D)]. Whereas for the large 16 nm pore the diffusive flow rate was about the same as for the "folded" particles, the flow rate was strongly reduced for narrower pores. But not only the values of C_1 are smaller than for the compact, folded models, also C_2 is smaller. For the narrowest 4-nm pores, C_2 is even smaller than the $C_2 = \pi/2$ prediction of the continuum model. The value of C_2 = 0.8 for these narrow pores implies that here the transport of the unfolded

protein chain is an essentially one-dimensional process once the protein is inside the pore, while the small values of C_1 indicate that it is difficult for the Gaussian-chain like proteins to thread into the narrow pore opening. For comparison, the 2-bead model of the unfolded protein has a length of about 7 nm, whereas the 6-bead model can be extended to a length of nearly 15 nm (five bonds of 2.85 nm each plus two bead radii). Consequently, these unfolded bead-spring models do not fit into the small 4 nm pores in arbitrary orientations and the obtained flow rates are thus three- to fivefold lower compared to the wider pores.

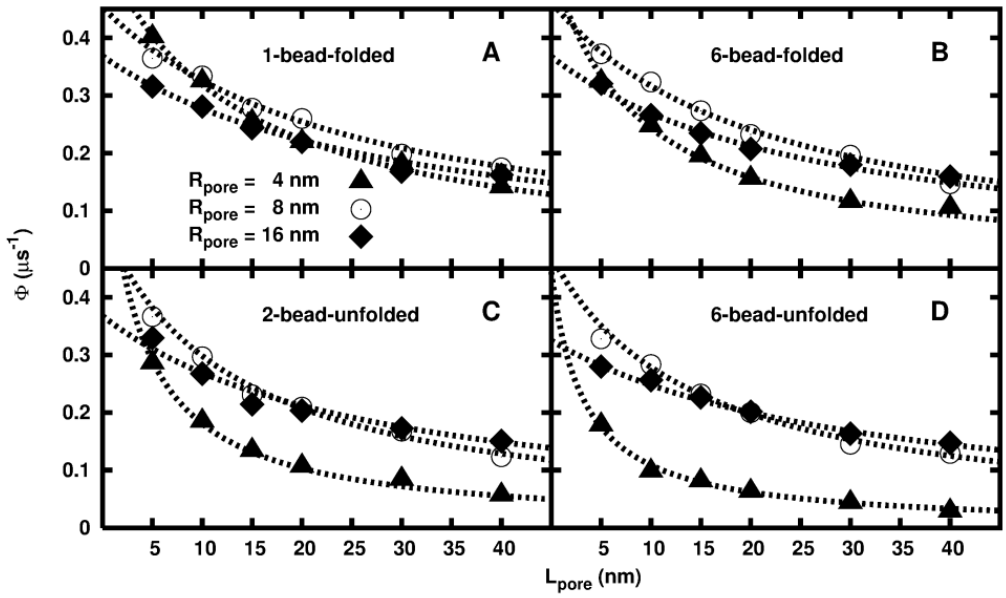

Figure 5.8 Diffusive flow rate vs. pore length for various pore sizes and protein models. In all scenarios, the same total pore area of a 16-nm pore was used, i.e., 16 pores with 4 nm radius and 4 pores of radius 8 nm. The protein models are the folded "Normal"-sized 1-bead model [panel (A)] and the 6-bead folded (B), the 2-bead-unfolded (C), and the 6-bead unfolded (D) bead-spring representations. The dashed lines are fitting curves according to Eq. 5.13. The statistical errors of all data points are about 3% so that the error bars are smaller than the symbols.

When the extended length of the unfolded protein is long compared to the pore diameter then the polymer chain has to align along the pore. This could be observed clearly in our simulations for the 6-bead unfolded model. We found that its averaged radius of gyration of about 3 nm did not change substantially even inside the smallest pores of R_{pore} = 4 nm. Although the model had a similar averaged radius of gyration in the bulk and inside the pore, this polymer chain model needed more time to find a suitable conformation and orientation to enter a pore. Moreover, the geometric confinement inside the pore also affected the orientation of the unfolded proteins when they moved through the pore.

The orientation of the bead-spring polymer models can be described by the vector pointing from the first to the last bead. Figure 5.9 displays the probabilities for a certain angle between the pore axis and the vector from the first to the last bead for the 6-bead model in the three differently sized pores. In the bulk volume far away from the membrane (at heights between 25 nm and 40 nm), the polymer chains were oriented randomly. Directly above the membrane, the orientation of the polymers was mainly perpendicular to the pore axis because of the membrane surface that confines their conformations when they are searching for the pore. Then, once inside the pore, the polymer chains were aligned mainly parallel to the pore axis. As shown in Figure 5.9, this behavior could be observed for all pore sizes, but was most prominent in the narrow 4-nm pores [panel (C)] and hardly noticeable in the wide 16 nm-radius pores [panel (A)]. This again shows that for long polymers and narrow pores, the entrance into the pore greatly affects the translocation rate.

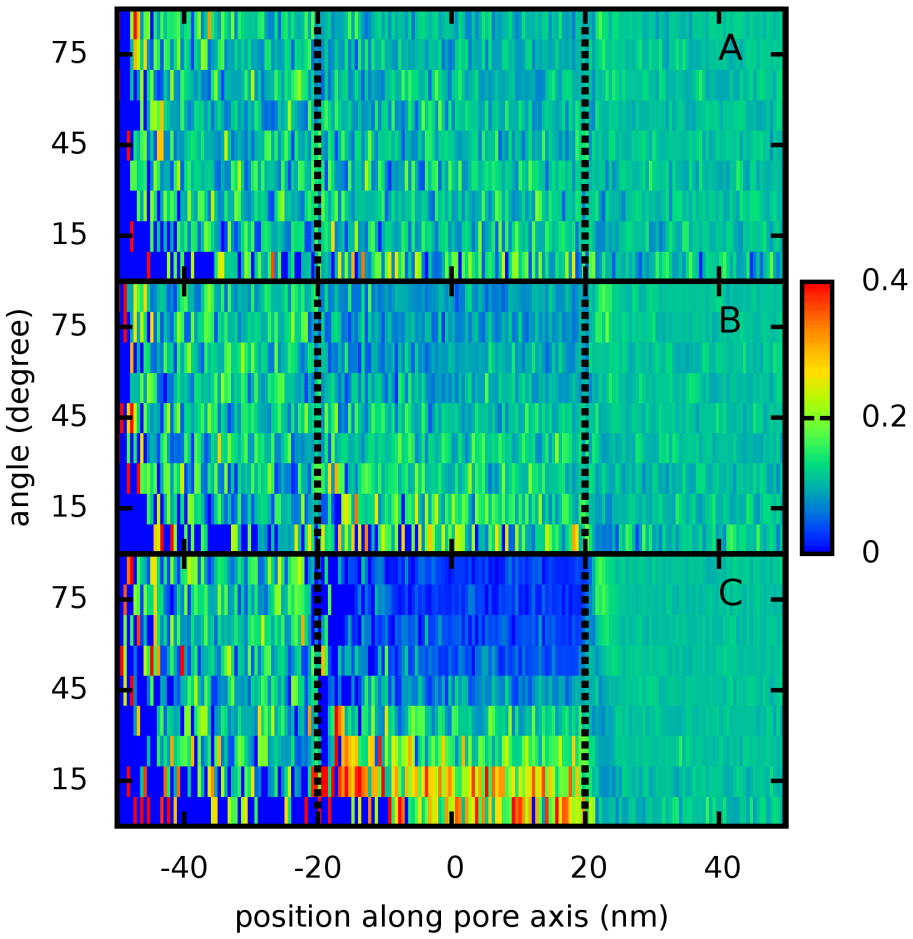

Figure 5.9 Probability distributions of the orientational preference of the 6-bead unfolded protein model inside the pores. The plots give the color coded probability to observe a given angle between the pore axis and the vector from the first to the last bead of the protein model (vertical axis) vs. the height in the simulation volume (horizontal axis). For each height bin, the probabilities were normalized independently. In these simulations, the membrane had a thickness of 40 nm and the two compartments above and below the membrane were 30 nm high with the *trans* side on the left and the *cis* compartment on the right side of the plots. The pore radii for the three panels are (A) 16 nm, (B) 8 nm, and (C) 4 nm. The relative fluctuations are larger in the *trans*-compartment because of the lower density.

6 Putative cholesterol-binding sites in human immunodeficiency virus (HIV) coreceptors CXCR4 and CCR5

This work had been published as "Zhukovsky MA, Lee PH, Ott A, Helms V: **Putative cholesterol-binding sites in human immunodeficiency virus (HIV) coreceptors CXCR4 and CCR5**. *Proteins* 2013, **81**(4):555-567". My own contribution of this work was to design and perform all docking calculations and to write part of the manuscript (explaining the docking method and the corresponding results) for publication.

6.1 Background

Human immunodeficiency virus Type 1 (HIV-1) and Type 2 (HIV-2), which are etiologic agents of the acquired immunodeficiency syndrome (AIDS), infect human cells via fusion of the virus membrane with a host cell membrane. The HIV envelope (Env) glycoprotein is organized as trimer. Each protomer consists of two noncovalently bound subunits, a surface subunit (SU) and a transmembrane (TM) subunit. The SU and TM subunits of HIV-1 Env are named gp120 and gp41, respectively, whereas gp125 and gp36 are the respective SU and TM subunits of HIV-2. The fusion process is initiated by interaction of HIV Env with the cellular membrane receptor CD4 and one of several coreceptors, usually CXCR4 or CCR5. CXCR4-using HIV strains are called X4-tropic, and those using CCR5 are called R5-tropic. [9]

CXCR4 and CCR5 are chemokine receptors. Chemokines are a multigene family of cytokines, which are small signaling protein molecules that are secreted by cells. CXCR4, CCR5, and other chemokine receptors belong to the rhodopsin family of seven-transmembrane G protein-

coupled receptors (GPCR). CXCR4 is a receptor for the chemokine CXCL12 (also known as SDF-1). CXCR4 plays a role in trafficking of naive lymphocytes, fetal development, migration of several types of neural cells, mobilization of hematopoietic stem cells, and synaptic transmission. CXCR4 has been implicated in rheumatoid arthritis, in the WHIM (Warts, Hypogammaglobulinemia, Infection, and Myelokathexis) syndrome, in different types of cancer, and in the immune response during fungal asthma. [9, 117] A crystal structure of human CXCR4 has been determined recently. [118] CCR5 is a receptor for chemokines CCL3 (also known as MIP-1α), CCL4 (MIP-1β) and CCL5 (RANTES). CCR5 is involved in the migration of Th1 cells, NK cells, and monocytes toward an increasing concentration of β chemokines. CCR5 alleles play a role in many diseases, including infectious and inflammatory diseases as well as cancer. [9, 117] However, the crystal structure of CCR5 was just resolved recently after we conducted this study. [119]

Cholesterol is an essential component of eukaryotic membranes and has been reported to have a modulatory role on the structure and functional activity of many membrane proteins. It can modulate membrane receptor function either by direct binding to the respective protein and thereby possibly inducing further conformational changes, or in an indirect way by altering the physical properties (e.g., fluidity) of the membrane in which the protein is embedded, or by a combination of both effects. [120]

Depletion of cholesterol from target cells inhibits entry of both X4 and R5 strains of HIV-1 and membrane fusion mediated by Env of both X4-tropic and R5-tropic HIV-1. However, the fusion activity could be recovered by adding back cholesterol to these cells. Furthermore, binding of CXCL12 and of a monoclonal conformation-dependent anti-CXCR4 antibody 12G5 to CXCR4-

expressing cells was inhibited by cholesterol extraction from these cells. Cholesterol extraction from CCR5-expressing cells inhibited binding of CCL4 and anti-CCR5 antibodies to CCR5. Reloading cells with cholesterol restored binding of these molecules to CXCR4 and CCR5. Loss in ligand binding after cholesterol depletion was suggested to be likely due to conformational changes in CXCR4 and CCR5. [121-123]

Moreover, oxidation of cholesterol to 4-cholesten-3-one by cholesterol oxidase (CO) inhibits binding of CXCL12 and CCL4 to CXCR4 and CCR5 on the cell surface, respectively, resulting in the inhibition of chemokine-mediated intracellular calcium mobilization and chemotaxis. [124] CO treatment inhibited HIV-1 infection through CXCR4. Moreover, CO treatment induced conformational changes in CXCR4 and CCR5 as detected by differential loss in binding of epitope-specific monoclonal antibodies. Interestingly, the loading of treated cells with cholesterol after CO treatment does not restore the binding of monoclonal antibodies or chemokines to chemokine receptors CXCR4 and CCR5. [124] This may be expected, if both cholesterol and 4-cholesten-3-one have relatively high affinities and low off rates for chemokine receptors in the membrane, and cholesterol stabilizes the native conformations of CXCR4 and CCR5, whereas 4-cholesten-3-one does not. Studies on the influence of cholesterol oxidation on CXCR4 and CCR5 strongly suggests that cholesterol does not influence these receptors through an indirect way by disrupting lipid rafts or by changing membrane fluidity, but instead show direct binding of cholesterol to CXCR4 and CCR5. Taken together, all these prior data demonstrate that cholesterol is essential for the conformational stability and function of CXCR4 and CCR5. [124]

In this study we report the presence of a putative cholesterol-binding site located in a groove between transmembrane domains (TMD) 1 and 7 of CXCR4, near the inner membrane-water interface. Sequence alignment showed that a similar putative cholesterol-binding site may also be present in CCR5. We suggest that these cholesterol-binding sites in CXCR4 and CCR5 are responsible for the influence of cholesterol on various properties of these receptors.

6.2 Materials and Methods

6.2.1 Sequence alignment

The amino acid sequences of CXCR4 and CCR5 orthologs were retrieved from the GenBank [125] and Ensembl [126] databases. The amino acid sequences given in Table 6.3, Tables A.3 and A4 (in Appendix) were aligned by the web version of the Clustal Omega software [127] at http://www.ebi.ac.uk/Tools/msa/clustalo/. The amino acid sequences of CXCR4 and CCR5 orthologs represent different species belonging to diverse taxa that possess genes for these chemokine receptors. Generally, species in Tables A.3 and A.4 are presented in order from less divergent (on the top) to more divergent (on the bottom) from *Homo sapiens*. The putative CRAC motifs [128] in chemokine receptors were manually identified in these alignments.

6.2.2 Docking analysis

To find the optimal binding site and binding pose of a single cholesterol molecule on the surface of CXCR4, the software AutoDock4 (version 4.2) [45] was used for the docking calculations in the rigid protein-flexible ligand scheme. AutoDock4 explores and evaluates

ligand conformations according to the receptor structure during the docking simulation. First, the structure of the ligand cholesterol was completed by applying the program PRODRG [129] to add hydrogen atoms. To determine the partial atomic charges, quantum mechanical calculations were performed with the GAUSSIAN 03 program [130] at the Hartree-Fock level with the 6-31G* basis set. The program Antechamber from the Amber9 package [131] was used to compute the RESP (Restrained ElectroStatic Potential [132]) partial charges of the ligand. The web server PDB2PQR (version 1.8) [133] was used to add hydrogen atoms on the X-ray protein structures and to assign the partial charges of each atom according to the Amber parm94 force field. [134] The python scripts prepare_ligand4.py and prepare_receptor4.py from the package AutoDockTools4 [45] were then used to create the united atom models for the ligand and protein and prepare PDBQT files for the subsequent docking calculation.

It is generally not efficient to perform docking calculations in a huge grid box that encloses the whole protein, due to the excessive run time caused by the vast search space. Instead, we adopted a divide-and-conquer scheme to solve this problem for all protein docking cases in this study. We used a grid box of 50 × 66 × 40 grid points along the three Cartesian axes and a spacing between neighboring grid points of 0.375 Å. The only exception is Na^+-K^+ transporting ATPase where a larger grid box (60 × 70 × 50) was adopted due to its larger size. Grid boxes were arranged in three layers with each layer containing 12 grid boxes (Figure 6.1). The boxes overlap with adjacent boxes and cover the entire transmembrane lipid accessible surface of the protein. Because of the relatively small size of each grid box, the docking runs finished in a reasonable time (10–20 hours on a 2.8 GHz AMD Opteron core).

The Lamarckian genetic algorithm (LGA) was applied for the optimization processes throughout the docking calculation. The population size was set as 250 and the optimization was terminated after 5×10^7 energy evaluations which equal about 2.4×10^5 generations. Sixty LGA runs were carried out for each grid box and then the obtained ligand conformations were clustered using a threshold of 2.0 Å RMSD.

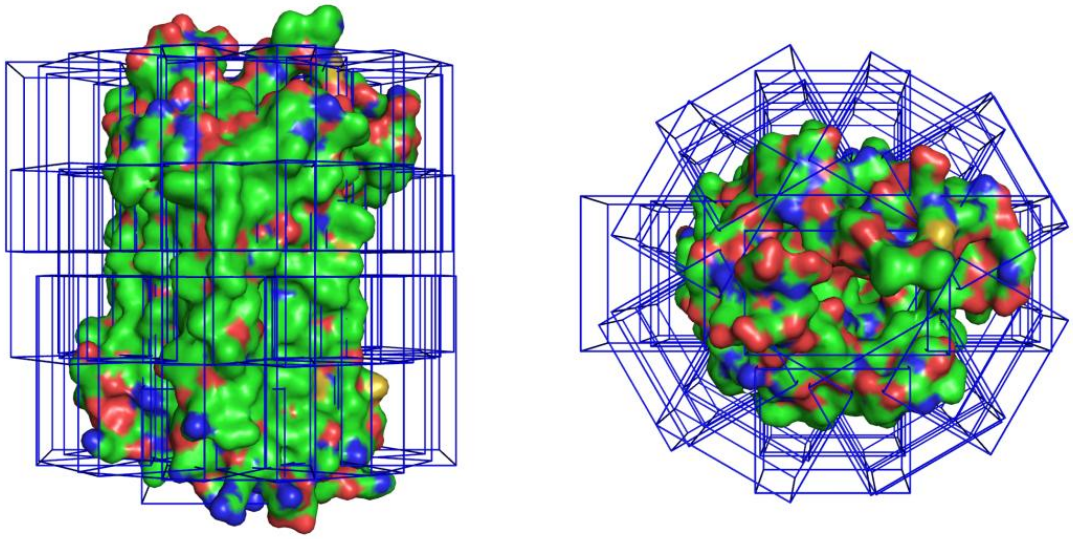

Figure 6.1 Global docking scheme. CXCR4 is shown in surface representation. The structure of the attached T4 lysozyme is not shown for clarity. Thirty-six partially overlapping grid boxes bounded by frames cover the entire target surface of CXCR4. The left figure shows a side view and the right one the top view.

6.3 Results and Discussion

6.3.1 CRAC motifs in CXCR4 and CCR5

An important cholesterol binding motif in proteins is the cholesterol recognition/interaction amino acid consensus (CRAC) motif. This motif is defined by the presence of the pattern L/V-X_{1-}

$_5$-Y-X_{1-5}-R/K, where $X_{1–5}$ represents between one and five residues of any amino acid. [128] It has been proposed that the aromatic ring of tyrosine could stack with one of the hydrocarbon rings of cholesterol, and the positively charged residue (arginine or lysine) may interact with the hydroxyl group of cholesterol. This motif is present in various proteins that are targeted to lipid rafts. CRAC motifs were recently identified in various GPCRs. [135] A conserved CRAC motif LWYIK is present in the gp41 subunit of the HIV-1 envelope glycoprotein, adjacent to its transmembrane helix, and mutations to this motif arrest HIV infection at the hemifusion stage. The LWYIK pentapeptide was shown to interact with cholesterol, whereas a modified IWYIK pentapeptide that does not contain the CRAC motif exhibited no preferential interaction with cholesterol. [136, 137]

We also attempted to identify the CRAC motif in the HIV receptor CD4 and major HIV coreceptors, CXCR4 and CCR5. We found that human CD4 does not contain any CRAC motif. Motif analysis of major HIV coreceptors identified three CRAC motifs in human CXCR4 and one in human CCR5. The CRAC motif that we identified in CCR5 is located in the TMD5 and involves residues L208/V209/V211-Y214-K219. The three CRAC motifs determined in CXCR4 are: (1) CRAC1 motif contains V59/L61/V62-Y65-K67/K68/R70 and is located partly in TMD1 and partly in the intracellular loop 1 that consists of residues 65 to 71; (2) CRAC2 motif is formed by V214/L216-Y219-K225 in TMD5; (3) CRAC3 motif involving L297-Y302-K308 is located partly in TMD7 and partly in the intracellular C-terminus that starts at residue A303.

However, the molecular docking analysis performed in this study showed that the three CRAC motifs in CXCR4 do not give rise to energetically favorable docking sites and are, therefore,

most probably, not functional. Instead, we identified a putative cholesterol-binding site in CXCR4 that is formed ‘‘collectively’’ by the CRAC1 and CRAC3 motifs (see Section 6.3.3).

6.3.2 Reproducing known interaction modes of cholesterol with α-helical transmembrane proteins

The AutoDock4 scoring function we used [138] has been parameterized on globular, soluble proteins that bind ligands in an aqueous environment. Here, however, the cholesterol binding occurs in the immediate vicinity of a membrane which is a different case. Before applying AutoDock4 in a predictive docking scenario to CXCR4, it was necessary to perform a thorough benchmark for its ability of reproducing known interaction modes of cholesterol with α-helical transmembrane proteins. For this, we scanned the PDBTM database [22] for structures of α-helical transmembrane proteins that were co-crystallized with cholesterol. Such a test of repredicting known interactions is known in the protein-ligand docking field as a ‘‘redocking’’ scenario. PDBTM (latest updated on June 28, 2012) contained only 11 structures of α-helical transmembrane proteins in complex with cholesterol. Among these 11 structures are 6 structures of β_2-adrenergic receptor, 3 of Na^+-K^+ transporting ATPase, and one each for *Acetabularia* rhodopsin and mouse μ-opioid receptor. For the redocking test, we selected one case out of the four different groups; their PDB IDs are 2RH1 [139, 140], 3KDP [141], 3AM6 [142], and 4DKL [143], respectively (Figure 6.2). The 2RH1 contains three bound cholesterol molecules per protein protomer; 3AM6 contains two bound cholesterol molecules. In both cases, we compared the results of our redocking benchmark against the cholesterol conformation in the co-crystal having most contacts or interactions with the protein.

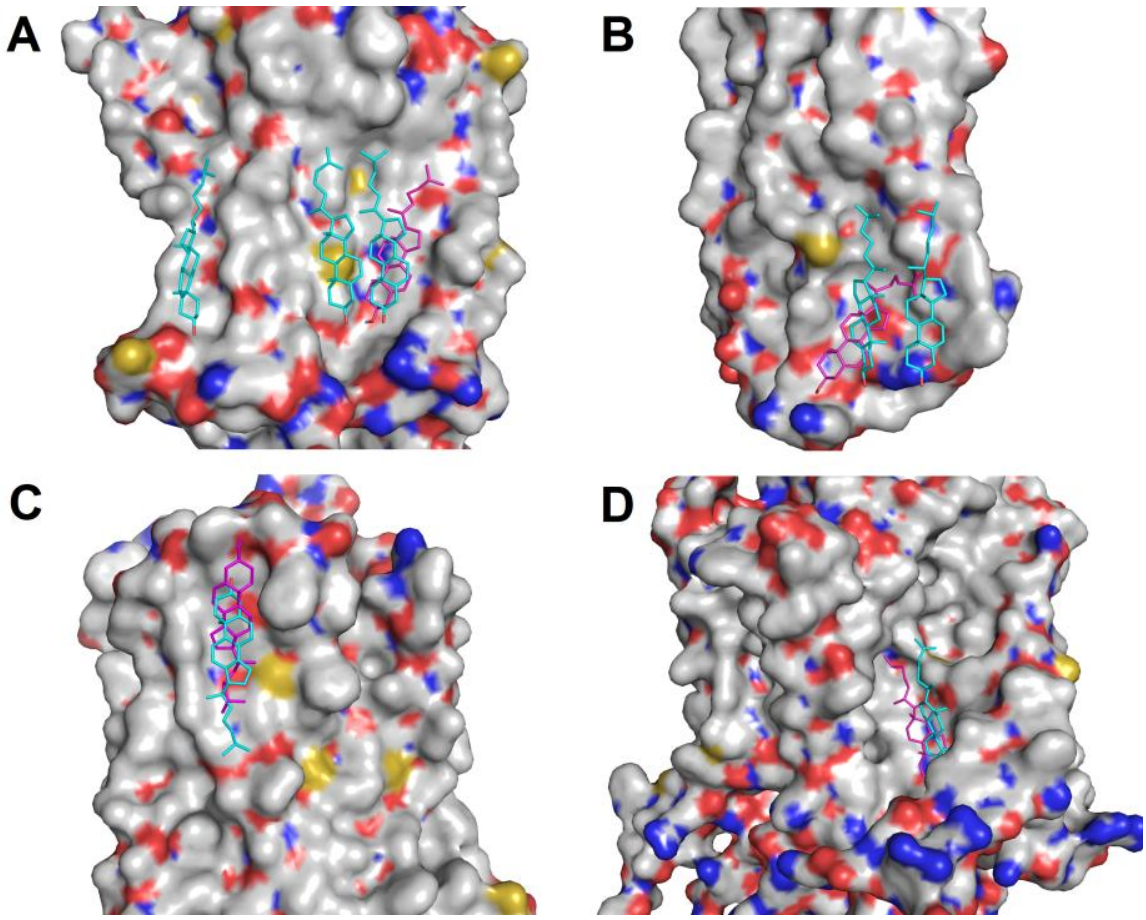

Figure 6.2 Redocking to cholesterol-binding sites found at four α-helical transmembrane proteins. The protein models in each panel are (A) β_2-adrenergic receptor (2RH1), (B) *Acetabularia* rhodopsin (3AM6), (C) μ-opioid receptor (4DKL), and (D) Na^+-K^+ transporting ATPase (3KDP). The proteins are shown in a surface representation and the cholesterol molecules as sticks. The cholesterol molecules colored in magenta are the conformations with the lowest RMSD produced by our docking calculation, whereas the cholesterol molecules colored in cyan were co-crystallized with the proteins.

Table 6.1 Redocking of cholesterol to four α-helical transmembrane proteins

PDB ID	Protein name	Global docking		Conformation with the lowest RMSD	
		Number of conformation	Distribution of energies (kcal/mol)	Rank / Energy (kcal/mol)[a]	$RMSD_{full}$ (Å)/ $RMSD_{ring}$ (Å)[b]
2RH1	β_2-adrenergic receptor (GPCR)	123	-8.31 ~ -4.43	12 / -7.79	3.08 / 1.50
3AM6	*Acetabularia* rhodopsin (GPCR)	119	-9.44 ~ -4.37	7 / -7.17	4.90 / 4.17
4DKL	μ-opioid receptor (GPCR)	131	-8.73 ~ -3.94	6 / -8.31	4.27 / 4.19
3KDP	Na^+-K^+ transporting ATPase	143	-10.02 ~ -4.36	17 / -7.93	4.10 / 3.34

[a]Rank among all conformations obtained by global docking according to estimated binding free energies.
[b]$RMSD_{full}$ considers all heavy atoms of the cholesterol molecule whereas $RMSD_{ring}$ concerns only the fused ring of cholesterol.

The redocking results are listed in Table 6.1. Also, the cholesterol conformations with the lowest root mean square deviation (RMSD) are shown in Figure 6.2. The RMSD is a measure of the difference between the target and the reference structures. In each case, more than one hundred conformations were generated within the 36 grid boxes of the global docking. The estimated binding free energies range between -4 and -10 kcal/mol. Table 6.1 shows that for 3KDP the conformation of lowest RMSD value against the co-crystallized cholesterol was ranked among the best 12% of all conformations according to the estimated binding free energy found by global docking. For 4DKL, the conformation with lowest RMSD was even among the best 5%. In the cases of 2RH1 and 3KDP, several conformations with lower estimated binding free energies than the one with smallest RMSD were docked in ‘‘improper’’ orientations where the hydroxyl group of cholesterol is buried inside the membrane or the cholesterol is perpendicular to the membrane normal.

Although these results show very convincingly that AutoDock4 is able to correctly identify the rough positioning of cholesterol in complex with different α-helical transmembrane proteins, the lowest RMSD values of 3.1 to 4.9 Å are larger than the values that are typically found when redocking small ligands to binding pockets of globular, soluble protein (often less than 2 Å). One explanation is that cholesterol has an aliphatic tail and this hydrophobic tail can be stabilized by either protein surface or the surrounding lipid environment. As shown in Figure 6.2, the aliphatic tails of co-crystallized cholesterol molecules are always positioned roughly parallel to the normal direction of the membrane in contrast to several docking conformations. Thus, smaller RMSDs of 1.5 to 4.2 Å were obtained when considering only the docking position of the fused ring of cholesterol (Table 6.1). Another explanation concerns the orientation of the fused ring of cholesterol. In the crystal structures used here, the two protruding carbon atoms (C18 and C19) in the fused ring can either be directed at the protein surface (in the cases of 3AM6, 4DKL, and 3KDP) or at the surrounding lipid (in the case of 2RH1). In the docked conformations, however, these two protruding atoms always point at the lipid surrounding (Figure 6.2). It appears a tough challenge for AutoDock to capture these features at the moment. Due to the optimization process of finding the lowest interaction energy between protein and ligand, and possibly due to the lacking representation of the lipid environment, the resulting cholesterol conformations tend to attach to the protein surface by its flat side but not the protruding side because more contacts are formed with the protein in this way. This is likely also the reason why the redocking of 2RH1 has the lowest RMSD value among the four cases because, here, also the crystal orientation contains outward-pointing C18 and C19 atoms.

We also performed two further tests of docking cholesterol to apo-conformations of β_2-adrenergic receptors (3P0G [144] and 3SN6 [145]) that had no cholesterol bound. The docked ligand conformations obtained for these two apoproteins were compared against the cholesterol conformation in holo-protein 2RH1 after structural alignment of the proteins. The results are shown in Table 6.2 and Figure 6.3. The energy and the ranking are similar to the results for 2RH1. To our surprise, the lowest RMSD values were even lower than obtained for 2RH1.

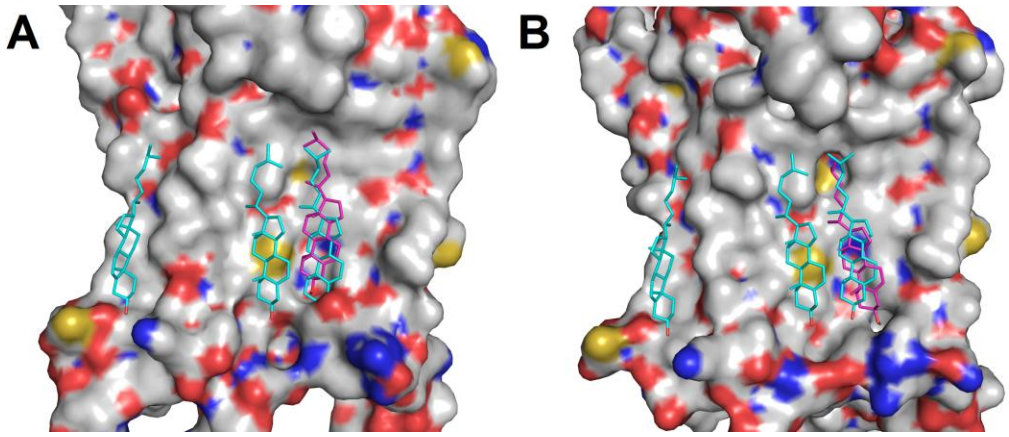

Figure 6.3 Redocking of cholesterol to two apo-conformations of β_2-adrenergic receptor without cholesterol bound. The PDB IDs are (A) 3P0G and (B) 3SN6. β_2-adrenergic receptor is shown in a surface representation and the cholesterol molecules as sticks. The cholesterol molecules colored in magenta are the conformations with the lowest RMSD. For comparison, cholesterol molecules colored in cyan were mapped from the conformations in the co-crystal structure 2RH1 according to the structural alignment of proteins.

Table 6.2 Redocking of cholesterol to the apo-form of β_2-adrenergic receptor containing no bound cholesterol molecule.

		Global docking		Conformation with the lowest RMSD	
PDB ID	Protein name	Number of conformation	Distribution of energies (kcal/mol)	Rank / Energy (kcal/mol)[a]	$RMSD_{full}$ (Å)/ $RMSD_{ring}$ (Å)[b]
3P0G	β_2-adrenergic receptor (GPCR)	107	-9.05 ~ -4.98	16 / -7.39	2.12 / 2.02
3SN6	β_2-adrenergic receptor (GPCR)	143	-8.83 ~ -4.00	13 / -7.85	1.71 / 1.49

[a]Rank among all conformations obtained by global docking according to estimated binding free energies.
[b]$RMSD_{full}$ considers all heavy atoms of the cholesterol molecule whereas $RMSD_{ring}$ concerns only the fused ring of cholesterol.

As mentioned before, the scoring function of AutoDock4 was parameterized based on the interactions of ligands and biomolecules in aqueous solution whereas the co-crystallized cholesterol molecules on the protein surface are stabilized not only by the protein but also by the hydrophobic solvent molecules present in purification and crystallization. When a transmembrane protein is functional in the membrane, these cholesterol molecules are stabilized by the surrounding lipid molecules. Thus the deviation between the co-crystallized cholesterol and the docking result may be attributed to such additional stabilizing interactions that are not captured by the scoring function used here. Even though, the AutoDock4 docking reproduced the important ligand-receptor interactions and predicted binding positions in close proximity to the crystal structures. This encouraging result confirms the suitability of the docking analysis for positioning cholesterol on the lipid accessible surface of membrane proteins. In the next step, we thus employed the same analysis for predicting putative interactions between CXCR4 and cholesterol.

6.3.3 Identification of a cholesterol-binding site in CXCR4 using docking analysis

The protein structure of human CXCR4 was obtained from the PDB as 3ODU.pdb [118]. To identify the binding site and key residues involving in cholesterol binding, we also implemented a global docking scheme that covered the entire surface of the CXCR4 structure (see Section 6.2.2). After clustering, AutoDock4 found 145 cholesterol conformations in the 36 grid boxes surrounding CXCR4. The binding free energies ranged from -10.44 to -4.34 kcal/mol. Cholesterol molecules were mainly docked in the concavities formed between transmembrane helices because these provided good structural complementarity (Figure 6.4). The three CRAC motifs mentioned above turned out not to be favorable binding sites because of their low computed binding affinities. From the structural point of view, the side chains of $L61^{1.55}$, $Y65^{1.59}$ (superscripts refer to the Ballesteros-Weinstein numbering), and K68 of the CRAC1 motif align roughly in one line and form a ridge that is energetically unfavorable for binding cholesterol. The side chains of $Y219^{5.58}$ and K225 of the CRAC2 motif point to different sides of TMD5, so that it seems impossible for cholesterol to reach these two residues simultaneously. For the CRAC3 motif, K308 resides on the intracellular C-terminus and is located too far away from the other two residues ($L297^{7.48}$ and $Y302^{7.53}$) to form a feasible binding site for cholesterol.

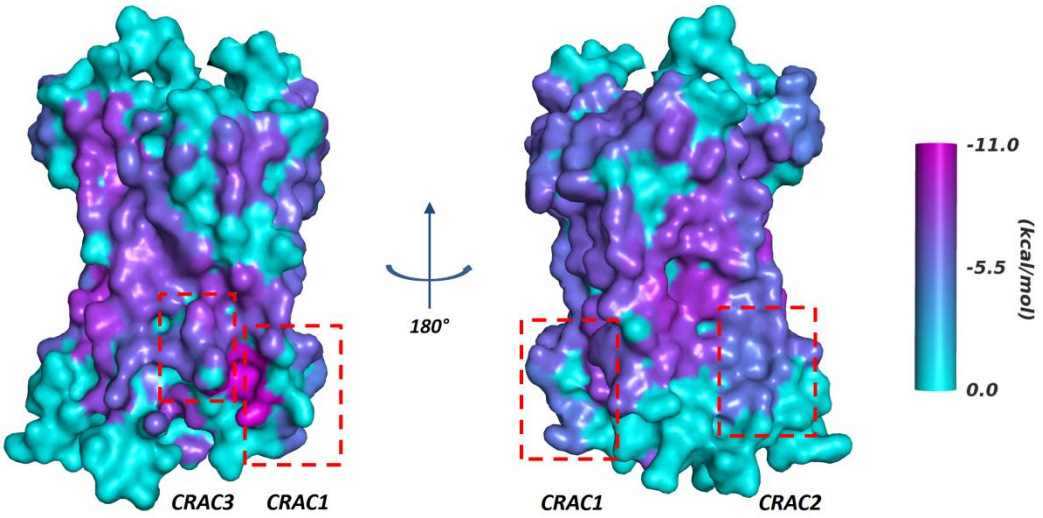

Figure 6.4 The distribution of predicted binding affinities for cholesterol on the surface of CXCR4. No cholesterol molecules were docked to cyan regions. Other surface regions are colored according to the most favorable docking scoring for cholesterol. The C terminus of CXCR4 (from F304 to Q328) is not shown for clarity.

The binding site with the highest affinity predicted by AutoDock4 is located in the groove between TMD1 and TMD7 near the inner membrane-water interface (Figure 6.5A). In this position, the ε-amino group of K67$^{1.61}$ establishes a hydrogen bond with the hydroxyl group of cholesterol, Y302$^{7.53}$ stacks with cholesterol by its aromatic side chain, and a number of residues (T51$^{1.45}$, G55$^{1.49}$, L58$^{1.52}$, V59$^{1.53}$, V62$^{1.56}$, M63$^{1.57}$, C296$^{7.47}$, P299$^{7.50}$, I300$^{7.51}$, A303$^{7.54}$) form hydrophobic contacts with cholesterol (Figure 6.5). Interestingly, these three factors (hydrogen bonding, aromatic stacking, and hydrophobic contacts) are consistent with the pattern of the CRAC motif. Although these residues do not align in a localized sequence as required by the CRAC motif, the binding site is formed ‘‘collectively’’ by the CRAC1 (V59$^{1.53}$, V62$^{1.56}$, and K67$^{1.61}$) and CRAC3 (Y302$^{7.53}$) motifs. These four residues reside on TMD1 and TMD7, but they are direct neighbors in three-dimensional space where they shape a feasible

binding groove for cholesterol. Besides that, the computed binding free energy with the unbound structure of the CXCR4 is -10.44 kcal/mol for this ligand conformation that is much more favorable than the -6.89 kcal/mol computed for the β_2AR-cholesterol complex based on the bound X-ray crystal structure (2RH1). Also, this conformation can be further stabilized by the membrane because the hydroxyl group of cholesterol would be located at the same height as the hydrophilic groups of the lipid molecules, and the aliphatic part of cholesterol would contact the hydrophobic tails of the lipids. All these factors demonstrate the high plausibility of the found binding site for cholesterol in CXCR4.

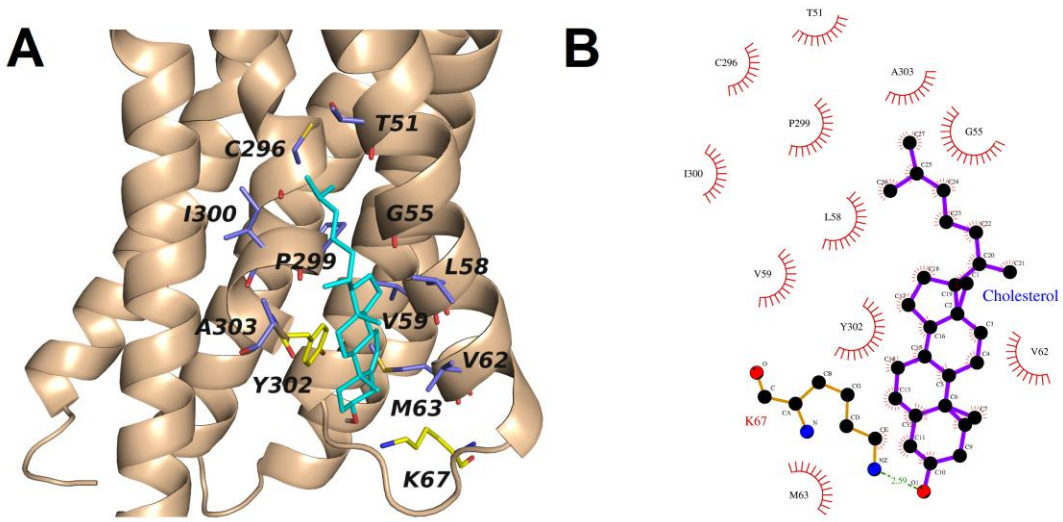

Figure 6.5 The best predicted binding pose and interaction map for cholesterol on CXCR4. (A) The protein is shown as ribbon. Cholesterol and the residues interacting with cholesterol are represented by sticks. The residues involved in hydrophobic contacts with cholesterol are colored in slate blue. The C terminus of CXCR4 (from F304 to Q328) is not shown for clarity. (B) Shows the same conformation as in (A) in the representation generated by LigPlot [146]. Residues represented as red eyelashes are involved in hydrophobic contacts. A tight ionic hydrogen bond is formed between atoms NZ of $K67^{1.61}$ and O of cholesterol with a distance of 2.59 Å.

The second best binding site at the lipid exposed surface of the protein with an estimated binding affinity of -8.39 kcal/mol is located at the concavity between TMD5 and TMD6. In this pocket, cholesterol forms favorable hydrophobic interactions with several pocket lining residues (including L208$^{5.47}$, I209$^{5.48}$, G212$^{5.51}$, I215$^{5.54}$, L216$^{5.55}$, V242$^{6.38}$, I245$^{6.41}$, L246$^{6.42}$, and F249$^{6.45}$). However, the hydroxyl group of the cholesterol docked into this concavity is positioned near the center of the lipid bilayer so that the hydroxyl group would be surrounded by hydrophobic lipid tails. Both the higher binding affinity and the more favorable orientation of the hydroxyl group favor the first binding site for cholesterol in CXCR4.

6.3.4 Identification of a putative cholesterol-binding site in human CCR5 using sequence alignment

Sequence alignment between human chemokine receptors shows that out of twelve CXCR4 residues forming contacts with cholesterol, nine (G$^{1.49}$, L$^{1.52}$, V$^{1.53}$, K$^{1.61}$, C$^{7.47}$, P$^{7.50}$, I$^{7.51}$, Y$^{7.53}$, A$^{7.54}$) are present also in human CCR5, see Table 6.3. Three CXCR4 residues (T51$^{1.45}$, V62$^{1.56}$, M63$^{1.57}$) that are not identical to the corresponding residues in CCR5 form hydrophobic contacts with cholesterol. In CCR5, the corresponding residues (F43$^{1.45}$, I54$^{1.56}$, L55$^{1.57}$) are hydrophobic and, hence, are also able to form hydrophobic contacts with cholesterol. Therefore we suggest that in CCR5, similar to CXCR4, the putative cholesterol-binding site is present in the groove between TMD1 and TMD7 near the inner membrane-water interface.

The putative cholesterol-binding sites in CXCR4 and CCR5 are highly conserved among various species as shown in Tables A.3 and A.4 that are multiple sequence alignments of CXCR4 orthologs and CCR5 orthologs, respectively.

Table 6.3 Aligned residues in the putative cholesterol-binding sites in human CXCR4 and CCR5

Residues (Ballesteros-Weinstein numbering)												
1.45	1.49	1.52	1.53	1.56	1.57	1.61	7.47	7.50	7.51	7.53	7.54	Receptor
T51	G55	L58	V59	**V62**	**M63**	K67	C296	P299	I300	Y302	A303	CXCR4
F43	G47	L50	V51	**I54**	**L55**	K59	C291	P294	I295	Y297	A298	CCR5

Residues that are not identical in two receptors are shown in boldface.

7 Conclusions

In Chapter 3, we presented a novel toolkit PROPORES to analyze the concavities of biomolecular structures automatically. The toolkit consists of three separate programs including PoreID for identifying pores (pockets, cavities, and channels), PoreTrace for determining the axes and radius profile of the channel, and GateOpen for opening the gate between two neighboring pores. PROPORES was designed as a general method for all classes of proteins and all kinds of concavities. PoreID and PoreTrace are grid-based methods whose computing time will grow exponentially for finer resolution as shown in Figure 3.3. A possible solution is to parallelize the implementation of PoreID. In Section 3.3.1, steps (b), (c) and (d) work on scanning each voxels for PSP events, scanning vectors for redundancy and for perpendicularity, respectively. These scanning processes are independent and are suitable to be parallelized. However, it will not work for PoreTrace because Dijkstra's algorithm implements in a serial manner due to the dependency of neighboring nodes.

PROPORES provides several types of useful information of pores (volume, lining residues, trajectory of pore axis and radius) for known protein structures. One can use the physicochemical properties or conservation scores of pore-lining residues as features to predict ligand binding sites [26] or analyze the evolution of the pocketome during the course of a molecular dynamics simulation of a protein [147]. What we have done is to annotate transmembrane pore-lining residues for a non-redundant set of α-helical transmembrane proteins as described in Chapter 4.

In Chapter 4, we presented the new method PRIMSIPLR for predicting of PLRs from the sequences of α-helical transmembrane proteins. This method was developed on a

comprehensive data set containing 90 protein structures. The amino acid composition of the data set reflects the expected characteristics of residues in different environments. Residues outside the membrane prefer to be charged and polar, whereas pore-lining amino acids have a higher content of charged and polar side chains compared to the other residues embedded in the membrane. Amino acids with aromatic side chains are crucial for PLRs by virtue of their function as gate. The typical length of a PLR stretch and the periodic pattern of PLRs are related to the structural trait of the α-helix.

We trained an SVM with a multiple independent random sampling scheme to account for the imbalanced nature of our data set. The evolutionary conservation score calculated by Rate4Site [83] gave substantial improvements in the prediction results whereas the three physicochemical properties had no apparent effects. Furthermore, we provide a confidence index for each positive prediction. A higher confidence index implies a more reliable prediction. Our predictor outperforms MEMSAT-SVM on 19 out of 23 non-redundant novel protein structures. The most challenging issue is to distinguish 'P' and 'M' residue types inside the membrane. More characteristic and representative features are necessary and the key issue to be overcome in future work. Additionally, some structures of transporters such as Leucine transporter [68, 148], glutamate/GABA antiporter [149] were determined in one of several alternative ligand transportation states. For the Leucine transporter, the complete view of PLRs can be captured by including multiple structures resolved in different states (i.e., 3F3A [68] is in outward-open conformation and 3TT3 [148] is in inward-open conformation). However, the glutamate/GABA antiporter [149] was crystallized either in a blocked or in an inward-open state

so that the identification of PLRs of these structures cannot present all possible residues that form contacts with the substrate. This may have caused some errors in the prediction.

In Chapter 5, we presented results of how protein models translocate through a membrane with small nanopores based on coarse-grained Brownian dynamics simulations. The latest developments of artificial porous membrane indicate that soon artificial membranes will become available with comparable size of pores as in biological membranes. Whereas it is difficult to control all experimental aspects of protein translocation through biological membranes, this can be achieved much more easily with the reproducible properties of artificial membranes. Thus, our simulations focused on the behavior of the proteins translocating through artificial porous membranes. These non-equilibrium steady-state simulations consisted of two constant density interfaces on opposite ends of the simulation volume and a membrane of variable thickness in between containing pores. Our model protein resembles the electron carrier cytochrome *c*, which is often used in experiments and simulations, both in the folded and unfolded state.

When considering a single pore, our simulations reproduced the prediction of an analytical continuum model (Eq. 5.12) particularly well for small particles that had a radius thousand fold smaller than the pore radius. For the larger particles, however, we found that the decrease of the flow rate with the pore length was slower in the simulations than in the analytical model. The translocation rate through the smaller pores was larger than expected. This effect could be explained by an increased protein density around the entrance of the pore, what leads to a steeper gradient through the pore and thus to an increased diffusive current. This density increase is due to the geometric confinement of the protein motion in the narrow pores that

reduces the overall diffusion coefficient. This is a purely geometric effect and is not related to an increased hydrodynamic friction in a confined geometry. In a second set of simulations, we found that the longer polymer models in the smaller pore showed a strong orientation along the pore axis, which further restricted their motion and thus reduced the translocation rate.

In a realistic system, the flow rate and the behavior of the proteins can be affected by many more factors such as the surface roughness inside the pore, particle adsorption on the membrane and inside the pores, or geometric effects such as the tortuosity of the pores or interconnections between adjacent pores. Starting from the simple setup presented here, these factors can now be added to the model one by one and investigated in future projects. Furthermore, to model translocation processes through structured biological pores and channels, additional beads with positive and negative charges can be placed on the pore surface or inside the channel to mimic the transmembrane pore proteins in more detail. Also, the crowded conditions inside a cell could be described by additional fixed or mobile spheres in the bulk regions. Furthermore, the protein models can be refined by adding charges or by sequence specific bead sizes and interactions. The model system presented in Chapter 5 is thus a starting point to explore the mechanisms of protein translocation and also serves as a first step towards the biological translocon mediated transport.

In Chapter 6, we report the presence of a putative cholesterol-binding site in the groove between TMD1 and TMD7 near the inner membrane-water interface of HIV coreceptor CXCR4 based on molecular docking studies. The corresponding residues in CCR5 show a similar arrangement of cholesterol-binding ability as found in CXCR4. Multiple sequence alignment among various species demonstrates that most residues belonging to the cholesterol-binding

sites in CXCR4 and CCR5 are highly conserved in orthologous proteins. We suggest that cholesterol molecules occupying these binding sites in CXCR4 and CCR5 are crucial for the function of these receptors due to the conformational stability. We also propose that mutations of the putative cholesterol-binding residues will make CXCR4 and CCR5 insensitive to membrane cholesterol concentration. Steroid drugs, that compete with cholesterol for the binding sites but do not maintain the native conformation of CXCR4 or CCR5, may become useful tools in the therapeutic control of AIDS and other diseases in which CXCR4 and CCR5 involve.

8 Appendix

Table A.1 List of PDB IDs with chain indices of the PH90 dataset

PDB ID and chain index	Protein name
1. 2WLM_A	KIR channel (K^+ channel)
2. 3E86_A	NaK channel (K^+ channel)
3. 3IFX_A	KcsA channel (K^+ channel)
4. 3UKM_A	K2P1 channel (K^+ channel)
5. 1LNQ_A	MthK channel (K^+ channel)
6. 1ORQ_A	KvAP channel (K^+ channel)
7. 3LUT_A	Kv1.2 channel (K^+ channel)
8. 4DXW_A	NaChBac channel (Na^+ channel)
9. 3BEH_A	MlotiK1: cyclic nucleotide-regulated channel
10. 3RQU_A	ELIC channel
11. 2XQ3_A	ELIC channel
12. 2IUB_A	CorA: metal ion transporter
13. 3RHW_A	GluC1: glutamate-gated chloride channel
14. 3UM7_A	TRAAK channel
15. 3PJZ_A	TrkH channel
16. 3S3W_A	ASIC1: acid sensing ion channel
17. 3I5D_A	ATP-gated $P2X_4$ ion channel
18. 2YVX_A	MgtE: Mg^{2+} transporter
19. 3J1Z_P	YiiP: Zn^{2+} transporter
20. 3M73_A	SLAC1 anion channel
21. 2L0J_A	Influenza A proton channel
22. 2KIX_A	Influenza B proton channel
23. 3B8C_A	P-type proton pump
24. 3AQP_A	SecDF
25. 4FG6_A	CLC Cl^-/H^+ exchange transporters
26. 1ZCD_A	NhaA: Na^+/ H^+ antiporter
27. 3ZUX_A	ASBT bile acid/Na^+ symporter
28. 2X79_A	Na^+-Hydantoin Transporter Mhp1
29. 3QNQ_A	Phosphorylation-coupled saccharide transporter
30. 4GBY_A	D-xylose/H^+ symporter XylE
31. 2R6G_F	MalF subunit of maltose transporter
32. 2R6G_G	MalG subunit of maltose transporter
33. 2XQ2_A	Na^+/glucose transporter
34. 3O7P_A	Fucose transporter
35. 2Y5Y_A	Lactose permease
36. 1PW4_A	Glycerol-3-phosphate transporter
37. 4F35_A	Na^+-dependent citrate transporter (NaCT)
38. 3BHS_A	Ammonia transporter
39. 3C1I_A	AmtB: ammonia channel
40. 3HD6_A	RhCG: ammonia transporter

41. 4EZC_A	Urea transporter
42. 1FX8_A	GlpF: glycerol facilitator
43. 3TDO_A	Hydrosulphide ion channel (formate/nitrite transport family)
44. 2WSX_A	CaiT: carnitine/butyrobetaine antiporter
45. 3QE7_A	UraA: uracil transporter
46. 2C3E_A	ADP/ATP translocase
47. 2YXQ_A	SecY subunit of translocon
48. 2ZQP_Y	SecY subunit of translocon
49. 4APS_A	Oligopeptide transporter
50. 3F3A_A	Leucine transporter
51. 4DJI_A	Glutamate-GABA antiporter
52. 2NWW_A	Aspartate transporter
53. 3OB6_A	AdiC: arginine/agmatine antiporter
54. 3TUJ_A	MetNI: methionine importer (ABC transporter)
55. 3QF4_A	TM287 subunit of heterodimeric ABC transporter
56. 3QF4_B	TM288 subunit of heterodimeric ABC transporter
57. 2NQ2_A	Putative metal-chelate-type ABC transporter
58. 2ONK_C	Molybdate transporter (ABC transporter)
59. 3P5N_A	RibU S-component of riboflavin transporter (ECF type)
60. 3RLB_A	ThiT S-component (ECF type)
61. 4DVE_A	BioY S-component biotin-specific of transporter (ECF type)
62. 3G5U_A	P-glycoprotein (multidrug resistence)
63. 2GFP_A	EmrD multidrug transporter
64. 3MKU_A	NorM MATE transporter (multidrug resistence)
65. 2GIF_A	AcrB transporter (multidrug resistence)
66. 2OAU_A	MscS: Small-conductance mechanosensitive channel
67. 2OAR_A	MscL: Large-conductance mechanosensitive channel
68. 2J58_A	Wza
69. 2WCD_A	Cytolysin A
70. 2LCK_A	UCP2: Mitochondrial uncoupling protein 2
71. 1ZLL_A	Phospholamban
72. 2ZW3_A	Connexin 26 gap junction channel
73. 4AW6_A	ZMPSTE24: zinc metalloprotease
74. 3KP9_A	Vitamin K epoxide reductase
75. 3B4R_A	S2P metalloprotease
76. 3DWW_A	MPGES1: Microsomal prostaglandin E synthase 1
77. 4A01_A	H^+-PPases: H^+-translocating pyrophosphatase
78. 4A2N_B	ICMT: Methyltransferase
79. 2PNO_A	LTC_4S: LTC_4 synthase
80. 2BG9_A	Membrane-associated acetylcholine receptor
81. 3KG2_A	AMPA-subtype glutamate receptor
82. 2Q7R_A	FLAP: 5-lipoxygenase–activating protein
83. 4EJ4_A	δ-opioid receptor (GPCR)

84. 3RZE_A	Histamine H_1 receptor (GPCR)
85. 4DAJ_A	M3 muscarinic acetylcholine receptor (GPCR)
86. 4AMJ_A	β_1AR (GPCR)
87. 2RH1_A	β_2AR (GPCR)
88. 3EML_A	A_{2A} AR (GPCR)
89. 3PBL_A	Dopamine D3 Receptor (GPCR)
90. 3V2W_A	$S1P_1$ receptor (GPCR)
91. 3OE8_A	CXCR4 (GPCR)
92. 2LNL_A	CXCR1 (GPCR)

Table A.2 List of PDB IDs with chain indices of the Test23 dataset

PDB ID and chain index	Protein name
1. 4BW5_A	Trek2 channel (K^+ channel)
2. 4H33_A	Voltage-gated K^+ channel
3. 4EV6_A	CorA: Mg^{2+} transporter
4. 4HKR_A	Orai: calcium release–activated calcium channel
5. 4K1C_A	Ca^{2+}/H^+ exchanger
6. 4KPP_A	Ca^{2+}/H^+ exchanger
7. 4HYT_A	Na^+/K^+-transporting ATPase subunit □1
8. 3UX4_A	Urea channel
9. 4IU8_A	Nitrate transporter
10. 4J05_A	Phosphate transporter
11. 3J41_A	aquaporin
12. 3ZOJ_A	aquaporin
13. 4IA4_A	aquaporin
14. 3WBN_A	MATE transporter (multidrug resistance)
15. 4HG6_A	Cellulose synthase subunit a
16. 3ZMH_A	Rhomboid Protease
17. 4IL3_A	CAAX protease Ste24p
18. 4GRV_A	NTS1: neurotensin receptor
19. 3VW7_A	Human protease-activated receptor 1 (GPCR)
20. 4IAQ_A	$5\text{-}HT_{1B}$ receptor (GPCR)
21. 4IB4_A	$5\text{-}HT_{2B}$ receptor (GPCR)
22. 4JKV_A	Human smoothened receptor (GPCR)
23. 4L6R_A	Human glucagon receptor (GPCR)

Table A.3 Multiple sequence alignment of the putative cholesterol-binding sites in CXCR4 from various species.

Residues (Ballesteros-Weinstein numbering)													Isoform	Database ID	Species
1.45	1.49	1.52	1.53	1.56	1.57	1.61	7.47	7.50	7.51	7.53	7.54				
T	G	L	V	V	M	K	C	P	I	Y	A			NP_003458.1	*Homo sapiens*
T	G	L	V	V	M	K	C	P	I	Y	A			AAC03718.1	*Pan troglodytes*
T	G	L	V	V	M	K	C	P	I	Y	A			AAF89352.1	*Gorilla gorilla*
T	G	L	V	V	M	K	C	P	I	Y	A			AAF89351.1	*Pongo pygmaeus*
T	G	L	V	V	M	E	C	P	I	Y	A			AAF42991.1	*Hylobates lar*
T	G	L	V	V	M	K	C	P	I	Y	A			AAF89349.1	*Hylobates lar*
T	G	L	V	V	M	K	C	P	I	Y	A			AAC39834.1	*Cercocebus atys*
T	G	L	V	V	M	K	C	P	I	Y	A			O08565.1	*Rattus norvegicus*
T	G	L	V	V	M	K	C	P	I	Y	A			NP_034041.2	*Mus musculus*
T	G	L	V	V	M	K	C	P	I	Y	A			ABX59689.1	*Oryctolagus cuniculus*
T	G	L	V	V	M	K	C	P	I	Y	A			NP_001009826.1	*Felis catus*
T	G	L	V	V	M	K	C	P	I	Y	A			ACH54079.1	*Panthera leo*
T	G	L	V	V	M	K	C	P	I	Y	A			NP_001041491.1	*Canis lupus familiaris*
T	G	L	V	V	M	K	C	P	I	Y	A			XP_002927722.1	*Ailuropoda melanoleuca*
T	G	L	V	V	M	K	C	P	I	Y	A			NP_776726.1	*Bos taurus*

T	G	L	V	V	M	K	C	P	I	Y	A			NP_998938.1	*Sus scrofa*
T	G	L	V	V	M	K	C	P	I	Y	A			XP_001490215.1	*Equus caballus*
T	G	L	V	V	M	K	C	P	I	Y	A			XP_003405904.1	*Loxodonta africana*
T	G	L	V	V	M	K	C	P	I	Y	A			XP_001370420.2	*Monodelphis domestica*
T	G	L	V	V	M	K	C	P	I	Y	A			XP_001510648.1	*Ornithorhynchus anatinus*
T	G	L	V	V	M	K	C	P	I	Y	A			XP_002198314.1	*Taeniopygia guttata*
T	G	L	V	V	M	K	C	P	I	Y	A			ENSAPLT00000001873	*Anas platyrhynchos*
T	G	L	V	V	M	K	C	P	I	Y	A			NP_989948.1	*Gallus gallus*
T	G	L	V	V	M	K	C	P	I	Y	A			XP_003207773.1	*Meleagris gallopavo*
T	G	L	V	V	M	K	C	P	I	Y	A			ENSACAT00000000709	*Anolis carolinensis*
L	G	L	V	V	M	K	C	P	I	Y	A			NP_001090831.1	*Xenopus tropicalis*
L	G	L	V	V	M	K	C	P	I	Y	A			BAA32797.1	*Cyprinus carpio*
M	G	L	V	V	M	K	C	S	I	Y	A			CAB60252.1	*Acipenser ruthenus*
L	G	L	V	V	M	K	C	P	I	Y	A			ACS45337.1	*Ictalurus punctatus*
L	G	L	V	V	M	K	C	P	I	Y	A			ABP48751.1	*Scophthalmus maximus*
L	G	L	V	V	L	R	C	P	L	Y	A		CXCR4a	ENSORLT00000014201	*Oryzias latipes*
L	G	L	V	V	M	K	C	P	I	Y	A		CXCR4b	ENSORLT00000025393	*Oryzias latipes*
L	G	L	V	V	M	K	C	P	I	Y	A			ACN10355.1	*Salmo salar*
L	G	L	V	V	M	K	C	P	I	Y	A		CXCR4a	NP_571957.2	*Danio rerio*

L	G	L	V	V	M	K	C	P	I	Y	A		CXCR4b	NP_571909.1	*Danio rerio*
L	G	L	V	V	M	K	C	P	I	Y	A		CXCR4c	XP_002663363.1	*Danio rerio*
L	G	L	V	V	M	K	C	P	I	Y	A		CXCR4d	XP_002664268.1	*Danio rerio*
L	G	L	V	V	M	K	C	P	I	Y	A			NP_001117814.1	*Oncorhynchus mykiss*
M	G	L	V	V	L	R	C	P	L	Y	A		CXCR4a	SINFRUP00000137823	*Takifugu rubripes*
L	G	L	V	V	V	K	C	P	I	Y	A		CXCR4b	GENSCAN00000006261	*Takifugu rubripes*
V	G	L	V	V	L	H	C	P	L	Y	A		CXCR4a	ENSTNIT00000010907	*Tetraodon nigroviridis*
L	G	L	V	V	M	K	C	P	I	Y	A		CXCR4b	ENSTNIT00000001795	*Tetraodon nigroviridis*
L	G	L	V	V	L	R	C	P	L	Y	A			ADR10240.1	*Epinephelus coioides*
L	G	L	I	I	L	T	C	P	V	Y	A			AAO21209.1	*Petromyzon marinus*

Table A.4 Multiple sequence alignment of the putative cholesterol-binding sites in CCR5 from various species.

Residues (Ballesteros-Weinstein numbering)													Isoform	Database ID	Species
1.45	1.49	1.52	1.53	1.56	1.57	1.61	7.47	7.50	7.51	7.53	7.54				
F	G	L	V	I	L	K	C	P	I	Y	A			NP_000570.1	*Homo sapiens*
F	G	L	V	I	L	K	C	P	I	Y	A			AAB62557.1	*Pan troglodytes*
F	G	L	V	I	L	K	C	P	I	Y	A			AAD47648.1	*Gorilla gorilla*
F	G	L	V	I	L	K	C	P	I	Y	A			AAD47664.1	*Pongo pygmaeus*
F	G	L	V	I	L	K	C	P	I	Y	A			AAD47714.1	*Macaca mulatta*
F	G	L	V	I	L	K	C	P	I	Y	A			ADM13528.1	*Cercocebus atys*
F	G	L	V	I	L	E	C	P	I	Y	A			AAD47792.1	*Cercopithecus mona*
F	G	L	V	T	L	K	C	P	V	Y	A			AAD47763.1	*Lemur catta*
F	G	M	V	I	L	K	C	P	V	Y	A			NP_446412.2	*Rattus norvegicus*
F	G	M	V	I	L	K	C	P	V	Y	A			NP_034047.2	*Mus musculus*
F	G	L	V	I	L	K	C	P	V	Y	A			ABD79048.1	*Oryctolagus cuniculus*
S	G	L	V	I	L	K	C	P	I	Y	A			NP_001009248.1	*Felis catus*
F	G	L	V	I	L	K	C	P	I	Y	A			NP_001012342.2	*Canis lupus familiaris*
F	G	L	V	I	L	K	C	P	I	Y	A			NP_001011672.2	*Bos taurus*
F	G	L	V	I	L	K	C	P	I	Y	A			NP_001001618.1	*Sus scrofa*
F	G	L	V	I	L	K	C	P	I	Y	A			ACJ46497.1	*Ovis aries*
F	G	L	V	I	L	K	C	P	I	Y	A			AEA91533.1	*Capra hircus*
S	G	L	V	V	L	K	C	P	V	Y	A			ABG01973.1	*Equus caballus*
S	G	L	V	V	L	K	C	P	I	Y	A			ABP65313.1	*Equus grevyi*
S	G	L	V	V	L	K	C	P	V	Y	A			ABP65312.1	*Equus asinus*

F	G	L	V	T	L	K	C	P	I	Y	A			ABS86966.1	*Loxodonta africana*
F	G	L	V	V	L	K	C	P	V	Y	A			XP_001379621.2	*Monodelphis domestica*
S	G	L	V	I	L	K	C	P	V	Y	A			XP_001517287.1	*Ornithorhynchus anatinus*
F	G	L	V	I	L	K	C	P	V	Y	A			XP_002198821.1	*Taeniopygia guttata*
F	G	L	V	I	L	K	C	P	V	Y	A		CCR5a	ENSACAT00000014112	*Anolis carolinensis*
F	G	L	V	I	L	K	C	P	V	Y	A		CCR5b	ENSACAT00000021030	*Anolis carolinensis*

9 References

1. Brown BS: **Biological membranes**: Biochemical Society; 1996.
2. Hedin LE, Illergard K, Elofsson A: **An Introduction to Membrane Proteins**. *J Proteome Res* 2011, **10**(8):3324-3331.
3. Marinissen MJ, Gutkind JS: **G-protein-coupled receptors and signaling networks: emerging paradigms**. *Trends Pharmacol Sci* 2001, **22**(7):368-376.
4. Rees DC, Johnson E, Lewinson O: **ABC transporters: the power to change**. *Nat Rev Mol Cell Bio* 2009, **10**(3):218-227.
5. Zimmermann R, Eyrisch S, Ahmad M, Helms V: **Protein translocation across the ER membrane**. *BBA-Biomembrane* 2011, **1808**(3):912-924.
6. Miller C: **An overview of the potassium channel family**. *Genome Biol* 2000, **1**(4):REVIEWS0004.
7. King LS, Kozono D, Agre P: **From structure to disease: The evolving tale of aquaporin biology**. *Nat Rev Mol Cell Bio* 2004, **5**(9):687-698.
8. Yu EW, Aires JR, Nikaido H: **AcrB multidrug efflux pump of Escherichia coli: Composite substrate-binding cavity of exceptional flexibility generates its extremely wide substrate specificity**. *J Bacteriol* 2003, **185**(19):5657-5664.
9. Alkhatib G: **The biology of CCR5 and CXCR4**. *Curr Opin Hiv Aids* 2009, **4**(2):96-103.
10. Tsirigos KD, Hennerdal A, Kall L, Elofsson A: **A guideline to proteome-wide alpha-helical membrane protein topology predictions**. *Proteomics* 2012, **12**(14):2282-2294.

11. Viklund H, Elofsson A: **OCTOPUS: improving topology prediction by two-track ANN-based preference scores and an extended topological grammar**. *Bioinformatics* 2008, **24**(15):1662-1668.

12. Nugent T, Jones DT: **Transmembrane protein topology prediction using support vector machines**. *Bmc Bioinformatics* 2009, **10**.

13. Nugent T, Jones DT: **Detecting pore-lining regions in transmembrane protein sequences**. *Bmc Bioinformatics* 2012, **13**.

14. Park Y, Hayat S, Helms V: **Prediction of the burial status of transmembrane residues of helical membrane proteins**. *Bmc Bioinformatics* 2007, **8**.

15. Hayat S, Walter P, Park Y, Helms V: **Prediction of the Exposure Status of Transmembrane Beta Barrel Residues from Protein Sequence**. *J Bioinform Comput B* 2011, **9**(1):43-65.

16. Fuchs A, Kirschner A, Frishman D: **Prediction of helix-helix contacts and interacting helices in polytopic membrane proteins using neural networks**. *Proteins* 2009, **74**(4):857-871.

17. Nugent T, Jones DT: **Predicting Transmembrane Helix Packing Arrangements using Residue Contacts and a Force-Directed Algorithm**. *Plos Comput Biol* 2010, **6**(3).

18. Altschul SF, Madden TL, Schaffer AA, Zhang JH, Zhang Z, Miller W, Lipman DJ: **Gapped BLAST and PSI-BLAST: a new generation of protein database search programs**. *Nucleic Acids Res* 1997, **25**(17):3389-3402.

19. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE: **The Protein Data Bank**. *Nucleic Acids Res* 2000, **28**(1):235-242.

20. Kelley LA, Sternberg MJE: **Protein structure prediction on the Web: a case study using the Phyre server**. *Nat Protoc* 2009, **4**(3):363-371.

21. Simons KT, Kooperberg C, Huang E, Baker D: **Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions**. *J Mol Biol* 1997, **268**(1):209-225.

22. Tusnady GE, Dosztanyi Z, Simon I: **Transmembrane proteins in the Protein Data Bank: identification and classification**. *Bioinformatics* 2004, **20**(17):2964-2972.

23. Lomize AL, Pogozheva ID, Lomize MA, Mosberg HI: **Positioning of proteins in membranes: A computational approach**. *Protein Sci* 2006, **15**(6):1318-1333.

24. Levitt DG, Banaszak LJ: **POCKET: a Computer-Graphics Method for Identifying and Displaying Protein Cavities and Their Surrounding Amino-Acids**. *J Mol Graphics* 1992, **10**(4):229-234.

25. Hendlich M, Rippmann F, Barnickel G: **LIGSITE: Automatic and efficient detection of potential small molecule-binding sites in proteins**. *J Mol Graph Model* 1997, **15**(6):359-363.

26. Huang BD, Schroeder M: **LIGSITE(csc): predicting ligand binding sites using the Connolly surface and degree of conservation**. *Bmc Struct Biol* 2006, **6**.

27. Raunest M, Kandt C: **dxTuber: Detecting protein cavities, tunnels and clefts based on protein and solvent dynamics**. *J Mol Graph Model* 2011, **29**(7):895-905.

28. Ho BK, Gruswitz F: **HOLLOW: Generating Accurate Representations of Channel and Interior Surfaces in Molecular Structures**. *Bmc Struct Biol* 2008, **8**.

29. Voss NR, Gerstein M: **3V: cavity, channel and cleft volume calculator and extractor**. *Nucleic Acids Res* 2010, **38**:W555-W562.

30. Petrek M, Otyepka M, Banas P, Kosinova P, Koca J, Damborsky J: **CAVER: a new tool to explore routes from protein clefts, pockets and cavities**. *Bmc Bioinformatics* 2006, **7**.

31. Coleman RG, Sharp KA: **Finding and Characterizing Tunnels in Macromolecules with Application to Ion Channels and Pores**. *Biophys J* 2009, **96**(2):632-645.

32. Dijkstra EW: **A note on two problems in connexion with graphs**. *Numer Math* 1959, **1**(1):269-271.

33. Weisel M, Proschak E, Schneider G: **PocketPicker: analysis of ligand binding-sites with shape descriptors**. *Chem Cent J* 2007, **1**.

34. Petrek M, Kosinova P, Koca J, Otyepka M: **MOLE: A Voronoi diagram-based explorer of molecular channels, pores, and tunnels**. *Structure* 2007, **15**(11):1357-1363.

35. Yaffe E, Fishelovitch D, Wolfson HJ, Halperin D, Nussinov R: **MolAxis: Efficient and accurate identification of channels in macromolecules**. *Proteins* 2008, **73**(1):72-86.

36. Edelsbrunner H, Facello M, Liang J: **On the definition and the construction of pockets in macromolecules**. *Discrete Appl Math* 1998, **88**(1-3):83-102.

37. Brady GP, Stouten PFW: **Fast prediction and visualization of protein binding pockets with PASS**. *J Comput Aid Mol Des* 2000, **14**(4):383-401.

38. Laskowski RA: **SURFNET: a Program for Visualizing Molecular-Surfaces, Cavities, and Intermolecular Interactions**. *J Mol Graphics* 1995, **13**(5):323-330.

39. Smart OS, Neduvelil JG, Wang X, Wallace BA, Sansom MSP: **HOLE: A program for the analysis of the pore dimensions of ion channel structural models**. *J Mol Graph Model* 1996, **14**(6):354-360.

40. Pellegrini-Calace M, Maiwald T, Thornton JM: **PoreWalker: A Novel Tool for the Identification and Characterization of Channels in Transmembrane Proteins from Their Three-Dimensional Structure**. *Plos Comput Biol* 2009, **5**(7).

41. Cristianini N, Shawe-Taylor J: **An introduction to support vector machines and other kernel-based learning methods**: Cambridge university press; 2000.

42. Coffey W, Kalmykov YP, Waldron JT: **The Langevin equation: with applications to stochastic problems in physics, chemistry, and electrical engineering**, vol. 14: World Scientific; 2004.

43. Huey R, Morris GM, Olson AJ, Goodsell DS: **A semiempirical free energy force field with charge-based desolvation**. *J Comput Chem* 2007, **28**(6):1145-1152.

44. Morris GM, Goodsell DS, Halliday RS, Huey R, Hart WE, Belew RK, Olson AJ: **Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function**. *J Comput Chem* 1998, **19**(14):1639-1662.

45. Morris GM, Huey R, Lindstrom W, Sanner MF, Belew RK, Goodsell DS, Olson AJ: **AutoDock4 and AutoDockTools4: Automated Docking with Selective Receptor Flexibility**. *J Comput Chem* 2009, **30**(16):2785-2791.

46. Huang SY, Grinter SZ, Zou XQ: **Scoring functions and their evaluation methods for protein-ligand docking: recent advances and future directions**. *Phys Chem Chem Phys* 2010, **12**(40):12899-12908.

47. Kitchen DB, Decornez H, Furr JR, Bajorath J: **Docking and scoring in virtual screening for drug discovery: Methods and applications**. *Nat Rev Drug Discov* 2004, **3**(11):935-949.

48. Yuriev E, Ramsland PA: **Latest developments in molecular docking: 2010-2011 in review**. *J Mol Recognit* 2013, **26**(5):215-239.

49. Solis FJ, Wets RJ-B: **Minimization by random search techniques**. *Math Oper Res* 1981, **6**(1):19-30.

50. Janson S, Merkle D, Middendorf M: **Molecular docking with multi-objective particle swarm optimization**. *Appl Soft Comput* 2008, **8**(1):666-675.

51. Korb O, Stützle T, Exner TE: **An ant colony optimization approach to flexible protein–ligand docking**. *Swarm Int* 2007, **1**(2):115-134.

52. Lang PT, Brozell SR, Mukherjee S, Pettersen EF, Meng EC, Thomas V, Rizzo RC, Case DA, James TL, Kuntz ID: **DOCK 6: Combining techniques to model RNA-small molecule complexes**. *Rna* 2009, **15**(6):1219-1230.

53. Jones G, Willett P, Glen RC, Leach AR, Taylor R: **Development and validation of a genetic algorithm for flexible docking**. *J Mol Biol* 1997, **267**(3):727-748.

54. deGroot BL, vanAalten DMF, Scheek RM, Amadei A, Vriend G, Berendsen HJC: **Prediction of protein conformational freedom from distance constraints**. *Proteins* 1997, **29**(2):240-251.

55. Lin J-H, Perryman AL, Schames JR, McCammon JA: **Computational drug design accommodating receptor flexibility: the relaxed complex scheme**. *J Am Chem Soc* 2002, **124**(20):5632-5633.

56. Bentley JL: **Multidimensional Binary Search Trees Used for Associative Searching**. *Commun Acm* 1975, **18**(9):509-517.

57. Desmet J, Demaeyer M, Hazes B, Lasters I: **The Dead-End Elimination Theorem and Its Use in Protein Side-Chain Positioning**. *Nature* 1992, **356**(6369):539-542.

58. Goldstein RF: **Efficient Rotamer Elimination Applied to Protein Side-Chains and Related Spin-Glasses**. *Biophys J* 1994, **66**(5):1335-1340.

59. Richards FM: **Areas, Volumes, Packing, and Protein-Structure**. *Annu Rev Biophys Bio* 1977, **6**:151-176.

60. Weiser J, Shenkin PS, Still WC: **Approximate atomic surfaces from linear combinations of pairwise overlaps (LCPO)**. *J Comput Chem* 1999, **20**(2):217-230.

61. Fischer G, Kosinska-Eriksson U, Aponte-Santamaria C, Palmgren M, Geijer C, Hedfalk K, Hohmann S, de Groot BL, Neutze R, Lindkvist-Petersson K: **Crystal Structure of a Yeast Aquaporin at 1.15 angstrom Reveals a Novel Gating Mechanism**. *Plos Biol* 2009, **7**(6).

62. Tornroth-Horsefield S, Wang Y, Hedfalk K, Johanson U, Karlsson M, Tajkhorshid E, Neutze R, Kjellbom P: **Structural mechanism of plant aquaporin gating**. *Nature* 2006, **439**(7077):688-694.

63. Miles EW: **Tryptophan synthase: structure, function, and subunit interaction**. *Adv Enzymol Relat Areas Mol Biol* 1979, **49**:127-186.

64. Hyde CC, Ahmed SA, Padlan EA, Miles EW, Davies DR: **3-Dimensional Structure of the Tryptophan Synthase Alpha-2-Beta-2 Multienzyme Complex from Salmonella-Typhimurium**. *J Biol Chem* 1988, **263**(33):17857-17871.

65. Ngo H, Harris R, Kimmich N, Casino P, Niks D, Blumenstein L, Barends TR, Kulik V, Weyand M, Schlichting I *et al*: **Synthesis and characterization of allosteric probes of substrate channeling in the tryptophan synthase bienzyme complex**. *Biochemistry-Us* 2007, **46**(26):7713-7727.

66. Yamashita A, Singh SK, Kawate T, Jin Y, Gouaux E: **Crystal structure of a bacterial homologue of Na+/Cl--dependent neurotransmitter transporters**. *Nature* 2005, **437**(7056):215-223.

67. Tsai CJ, Khafizov K, Hakulinen J, Forrest LR, Kramer R, Kuhlbrandt W, Ziegler C: **Structural Asymmetry in a Trimeric Na+/Betaine Symporter, BetP, from Corynebacterium glutamicum**. *J Mol Biol* 2011, **407**(3):368-381.

68. Singh SK, Piscitelli CL, Yamashita A, Gouaux E: **A Competitive Inhibitor Traps LeuT in an Open-to-Out Conformation**. *Science* 2008, **322**(5908):1655-1661.

69. Wilson IB, Harrison MA: **Turnover Number of Acetylcholinesterase**. *J Biol Chem* 1961, **236**(8):2292-2295.

70. Gilson MK, Straatsma TP, Mccammon JA, Ripoll DR, Faerman CH, Axelsen PH, Silman I, Sussman JL: **Open Back Door in a Molecular-Dynamics Simulation of Acetylcholinesterase**. *Science* 1994, **263**(5151):1276-1278.

71. Ripoll DR, Faerman CH, Axelsen PH, Silman I, Sussman JL: **An Electrostatic Mechanism for Substrate Guidance down the Aromatic Gorge of Acetylcholinesterase**. *P Natl Acad Sci USA* 1993, **90**(11):5128-5132.

72. Aird A, Wrachtrup J, Schulten K, Tietz C: **Possible pathway for ubiquinone shuttling in Rhodospirillum rubrum revealed by molecular dynamics simulation**. *Biophys J* 2007, **92**(1):23-33.

73. Henin J, Tajkhorshid E, Schulten K, Chipot C: **Diffusion of glycerol through Escherichia coli aquaglyceroporin GlpF**. *Biophys J* 2008, **94**(3):832-839.

74. Zhou HX, McCammon JA: **The gates of ion channels and enzymes**. *Trends Biochem Sci* 2010, **35**(3):179-185.

75. Hartmann C, Antes I, Lengauer T: **Docking and scoring with alternative side-chain conformations**. *Proteins* 2009, **74**(3):712-726.

76. Rose PW, Bi CX, Bluhm WF, Christie CH, Dimitropoulos D, Dutta S, Green RK, Goodsell DS, Prlic A, Quesada M *et al*: **The RCSB Protein Data Bank: new resources for research and education**. *Nucleic Acids Res* 2013, **41**(D1):D475-D482.

77. Apweiler R, Bairoch A, Wu CH: **Protein sequence databases**. *Curr Opin Chem Biol* 2004, **8**(1):76-80.

78. Lee PH, Helms V: **Identifying continuous pores in protein structures with PROPORES by computational repositioning of gating residues**. *Proteins* 2012, **80**(2):421-432.

79. Fu DX, Libson A, Miercke LJW, Weitzman C, Nollert P, Krucinski J, Stroud RM: **Structure of a glycerol-conducting channel and the basis for its selectivity**. *Science* 2000, **290**(5491):481-486.

80. Juretić D, Lučić B, Zucić D, Trinajstić N: **Protein transmembrane structure: recognition and prediction by using hydrophobicity scales through preference functions**. *Theor Comp Chem* 1998, **5**:405-445.

81. Zimmerma.Jm, Eliezer N, Simha R: **Characterization of Amino Acid Sequences in Proteins by Statistical Methods**. *J Theor Biol* 1968, **21**(2):170-201.

82. Vihinen M, Torkkila E, Riikonen P: **Accuracy of Protein Flexibility Predictions**. *Proteins* 1994, **19**(2):141-149.

83. Mayrose I, Graur D, Ben-Tal N, Pupko T: **Comparison of site-specific rate-inference methods for protein sequences: Empirical Bayesian methods are superior**. *Mol Biol Evol* 2004, **21**(9):1781-1791.

84. Chang CC, Lin CJ: **LIBSVM: A Library for Support Vector Machines**. *Acm T Intel Syst Tec* 2011, **2**(3).

85. Murata K, Mitsuoka K, Hirai T, Walz T, Agre P, Heymann JB, Engel A, Fujiyoshi Y: **Structural determinants of water permeation through aquaporin-1**. *Nature* 2000, **407**(6804):599-605.

86. Platt J: **Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods**. In: *Advances in large margin classifiers.* vol. 10. Cambridge, MA: MIT Press; 1999: 61-74.

87. Lin H-T, Lin C-J, Weng RC: **A note on Platt's probabilistic outputs for support vector machines**. *Mach Learn* 2007, **68**(3):267-276.

88. Adiga SP, Jin CM, Curtiss LA, Monteiro-Riviere NA, Narayan RJ: **Nanoporous membranes for medical and biological applications**. *Wires Nanomed Nanobi* 2009, **1**(5):568-581.

89. Zydney AL: **Protein separations using membrane filtration: New opportunities for whey fractionation**. *Int Dairy J* 1998, **8**(3):243-250.

90. Firnkes M, Pedone D, Knezevic J, Doblinger M, Rant U: **Electrically Facilitated Translocations of Proteins through Silicon Nitride Nanopores: Conjoint and Competitive Action of Diffusion, Electrophoresis, and Electroosmosis**. *Nano Lett* 2010, **10**(6):2162-2167.

91. Wickner W, Schekman R: **Protein translocation across biological membranes**. *Science* 2005, **310**(5753):1452-1456.

92. Striemer CC, Gaborski TR, McGrath JL, Fauchet PM: **Charge- and size-based separation of macromolecules using ultrathin silicon membranes**. *Nature* 2007, **445**(7129):749-753.

93. Huopaniemi I, Luo K, Ala-Nissila T: **Langevin dynamics simulations of polymer translocation through nanopores**. *J Chem Phys* 2006, **125**(12).

94. Storm AJ, Storm C, Chen JH, Zandbergen H, Joanny JF, Dekker C: **Fast DNA translocation through a solid-state nanopore**. *Nano Lett* 2005, **5**(7):1193-1197.

95. Tian P, Smith GD: **Translocation of a polymer chain across a nanopore: A Brownian dynamics simulation study**. *J Chem Phys* 2003, **119**(21):11475-11483.

96. Nikoofard N, Fazli H: **Free-energy barrier for electric-field-driven polymer entry into nanoscale channels**. *Phys Rev E* 2011, **83**(5).

97. Yong HS, Wang YL, Yuan SC, Xu B, Luo KF: **Driven polymer translocation through a cylindrical nanochannel: interplay between the channel length and the chain length**. *Soft Matter* 2012, **8**(9):2769-2774.

98. Metzler R, Luo K: **Polymer translocation through nanopores: Parking lot problems, scaling laws and their breakdown**. *Eur Phys J-Spec Top* 2010, **189**(1):119-134.

99. Slonkina E, Kolomeisky AB: **Polymer translocation through a long nanopore**. *J Chem Phys* 2003, **118**(15):7112-7118.

100. Wong CTA, Muthukumar M: **Polymer translocation through a cylindrical channel**. *J Chem Phys* 2008, **128**(15).

101. Edmonds CM, Hudiono YC, Ahmadi AG, Hesketh PJ, Nair S: **Polymer translocation in solid-state nanopores: Dependence of scaling behavior on pore dimensions and applied voltage**. *J Chem Phys* 2012, **136**(6).

102. Zhang KH, Luo KF: **Dynamics of polymer translocation into a circular nanocontainer through a nanopore**. *J Chem Phys* 2012, **136**(18).

103. Javidpour L, Tabar MRR, Sahimi M: **Molecular simulation of protein dynamics in nanopores. I. Stability and folding**. *J Chem Phys* 2008, **128**(11).

104. Javidpour L, Tabar MRR, Sahimi M: **Molecular simulation of protein dynamics in nanopores. II. Diffusion**. *J Chem Phys* 2009, **130**(8).

105. Moussavi-Baygi R, Jamali Y, Karimi R, Mofrad MRK: **Brownian Dynamics Simulation of Nucleocytoplasmic Transport: A Coarse-Grained Model for the Functional State of the Nuclear Pore Complex**. *Plos Comput Biol* 2011, **7**(6).

106. Eltis LD, Herbert RG, Barker PD, Mauk AG, Northrup SH: **Reduction of Horse Heart Ferricytochrome-C by Bovine Liver Ferrocytochrome-B5 - Experimental and Theoretical-Analysis**. *Biochemistry-Us* 1991, **30**(15):3663-3674.

107. Frembgen-Kesner T, Elcock AH: **Striking Effects of Hydrodynamic Interactions on the Simulated Diffusion and Folding of Proteins**. *J Chem Theory Comput* 2009, **5**(2):242-256.

108. Geyer T: **Many-particle Brownian and Langevin Dynamics Simulations with the Brownmove package**. *Bmc Biophys* 2011, **4**.

109. Segel DJ, Fink AL, Hodgson KO, Doniach S: **Protein denaturation: A small-angle X-ray scattering study of the ensemble of unfolded states of cytochrome c**. *Biochemistry-Us* 1998, **37**(36):12443-12451.

110. Medinanoyola M, McQuarrie DA: **On the Interaction of Spherical Double-Layers**. *J Chem Phys* 1980, **73**(12):6279-6283.

111. Winter U, Geyer T: **Coarse grained simulations of a small peptide: Effects of finite damping and hydrodynamic interactions**. *J Chem Phys* 2009, **131**(10).

112. Einstein A: **Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen**. *Annalen der physik* 1905, **322**(8):549-560.

113. Ermak DL, Mccammon JA: **Brownian Dynamics with Hydrodynamic Interactions**. *J Chem Phys* 1978, **69**(4):1352-1360.

114. Geyer T, Gorba C, Helms V: **Interfacing Brownian dynamics simulations**. *J Chem Phys* 2004, **120**(10):4573-4580.

115. Brunn PO, Fabrikant VI, Sankar TS: **Diffusion through Membranes - Effect of a Nonzero Membrane Thickness**. *Q J Mech Appl Math* 1984, **37**(May):311-324.

116. Cichocki B, Jones RB: **Image representation of a spherical particle near a hard wall**. *Physica A* 1998, **258**(3-4):273-302.

117. Choi WT, An J: **Biology and clinical relevance of chemokines and chemokine receptors CXCR4 and CCR5 in human diseases**. *Exp Biol Med* 2011, **236**(6):637-647.

118. Wu BL, Chien EYT, Mol CD, Fenalti G, Liu W, Katritch V, Abagyan R, Brooun A, Wells P, Bi FC *et al*: **Structures of the CXCR4 Chemokine GPCR with Small-Molecule and Cyclic Peptide Antagonists**. *Science* 2010, **330**(6007):1066-1071.

119. Tan Q, Zhu Y, Li J, Chen Z, Han GW, Kufareva I, Li T, Ma L, Fenalti G, Li J *et al*: **Structure of the CCR5 Chemokine Receptor–HIV Entry Inhibitor Maraviroc Complex**. *Science* 2013, **341**(6152):1387-1390.

120. Paila YD, Chattopadhyay A: **The function of G-protein coupled receptors and membrane cholesterol: specific or general interaction?** *Glycoconjugate J* 2009, **26**(6):711-720.

121. Ablan S, Rawat SS, Viard M, Wang JM, Puri A, Blumenthal R: **The role of cholesterol and sphingolipids in chemokine receptor function and HIV-I envelope glycoprotein-mediated fusion**. *Virol J* 2006, **3**.

122. Nguyen DH, Taub D: **CXCR4 function requires membrane cholesterol: Implications for HIV infection**. *J Immunol* 2002, **168**(8):4121-4126.

123. Nguyen DH, Taub D: **Cholesterol is essential for macrophage inflammatory protein 1 beta binding and conformational integrity of CC chemokine receptor 5**. *Blood* 2002, **99**(12):4298-4306.

124. Nguyen DH, Taub DD: **Inhibition of chemokine receptor function by membrane cholesterol oxidation**. *Exp Cell Res* 2003, **291**(1):36-45.

125. Benson DA, Karsch-Mizrachi I, Clark K, Lipman DJ, Ostell J, Sayers EW: **GenBank**. *Nucleic Acids Res* 2012, **40**(D1):D48-D53.

126. Flicek P, Amode MR, Barrell D, Beal K, Brent S, Carvalho-Silva D, Clapham P, Coates G, Fairley S, Fitzgerald S *et al*: **Ensembl 2012**. *Nucleic Acids Res* 2012, **40**(D1):D84-D90.

127. Sievers F, Wilm A, Dineen D, Gibson TJ, Karplus K, Li WZ, Lopez R, McWilliam H, Remmert M, Soding J *et al*: **Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega**. *Mol Syst Biol* 2011, **7**.

128. Li H, Papadopoulos V: **Peripheral-type benzodiazepine receptor function in cholesterol transport. Identification of a putative cholesterol recognition/interaction amino acid sequence and consensus patterns**. *Endocrinology* 1998, **139**(12):4991-4997.

129. Schuttelkopf AW, van Aalten DMF: **PRODRG: a tool for high-throughput crystallography of protein-ligand complexes**. *Acta Crystallogr D* 2004, **60**:1355-1363.

130. Frisch M, Trucks G, Schlegel H, Scuseria G, Robb M, Cheeseman J, Montgomery Jr J, Vreven T, Kudin K, Burant J: **Gaussian 03, Rev. D. 01**. *Gaussian, Inc, Wallingford CT* 2004.

131. Case DA, Cheatham TE, Darden T, Gohlke H, Luo R, Merz KM, Onufriev A, Simmerling C, Wang B, Woods RJ: **The Amber biomolecular simulation programs**. *J Comput Chem* 2005, **26**(16):1668-1688.

132. Bayly CI, Cieplak P, Cornell WD, Kollman PA: **A Well-Behaved Electrostatic Potential Based Method Using Charge Restraints for Deriving Atomic Charges - the Resp Model**. *J Phys Chem-Us* 1993, **97**(40):10269-10280.

133. Dolinsky TJ, Czodrowski P, Li H, Nielsen JE, Jensen JH, Klebe G, Baker NA: **PDB2PQR: expanding and upgrading automated preparation of biomolecular structures for molecular simulations**. *Nucleic Acids Res* 2007, **35**:W522-W525.

134. Cornell WD, Cieplak P, Bayly CI, Gould IR, Merz KM, Ferguson DM, Spellmeyer DC, Fox T, Caldwell JW, Kollman PA: **A 2nd Generation Force-Field for the Simulation of Proteins, Nucleic-Acids, and Organic-Molecules**. *J Am Chem Soc* 1995, **117**(19):5179-5197.

135. Jafurulla M, Tiwari S, Chattopadhyay A: **Identification of cholesterol recognition amino acid consensus (CRAC) motif in G-protein coupled receptors**. *Biochem Bioph Res Co* 2011, **404**(1):569-573.

136. Epand RM: **Proteins and cholesterol-rich domains**. *BBA-Biomembrane* 2008, **1778**(7-8):1576-1582.

137. Schroeder C: **Cholesterol-binding viral proteins in virus entry and morphogenesis**. In: *Cholesterol Binding and Cholesterol Transport Proteins:.* Springer; 2010: 77-108.

138. Wang J-C, Lin J-H, Chen C-M, Perryman AL, Olson AJ: **Robust scoring functions for protein–ligand interactions with quantum chemical charge models**. *J Chem Inf Model* 2011, **51**(10):2528-2537.

139. Cherezov V, Rosenbaum DM, Hanson MA, Rasmussen SGF, Thian FS, Kobilka TS, Choi HJ, Kuhn P, Weis WI, Kobilka BK *et al*: **High-resolution crystal structure of an engineered human beta(2)-adrenergic G protein-coupled receptor**. *Science* 2007, **318**(5854):1258-1265.

140. Rosenbaum DM, Cherezov V, Hanson MA, Rasmussen SGF, Thian FS, Kobilka TS, Choi HJ, Yao XJ, Weis WI, Stevens RC *et al*: **GPCR engineering yields high-resolution structural insights into beta(2)-adrenergic receptor function**. *Science* 2007, **318**(5854):1266-1273.

141. Morth JP, Pedersen BP, Toustrup-Jensen MS, Sorensen TLM, Petersen J, Andersen JP, Vilsen B, Nissen P: **Crystal structure of the sodium-potassium pump**. *Nature* 2007, **450**(7172):1043-1049.

142. Wada T, Shimono K, Kikukawa T, Hato M, Shinya N, Kim SY, Kimura-Someya T, Shirouzu M, Tamogami J, Miyauchi S *et al*: **Crystal Structure of the Eukaryotic Light-Driven Proton-Pumping Rhodopsin, Acetabularia Rhodopsin II, from Marine Alga**. *J Mol Biol* 2011, **411**(5):986-998.

143. Manglik A, Kruse AC, Kobilka TS, Thian FS, Mathiesen JM, Sunahara RK, Pardo L, Weis WI, Kobilka BK, Granier S: **Crystal structure of the mu-opioid receptor bound to a morphinan antagonist**. *Nature* 2012, **485**(7398):321-326.

144. Rasmussen SGF, Choi HJ, Fung JJ, Pardon E, Casarosa P, Chae PS, DeVree BT, Rosenbaum DM, Thian FS, Kobilka TS *et al*: **Structure of a nanobody-stabilized active state of the beta(2) adrenoceptor**. *Nature* 2011, **469**(7329):175-180.

145. Rasmussen SGF, DeVree BT, Zou YZ, Kruse AC, Chung KY, Kobilka TS, Thian FS, Chae PS, Pardon E, Calinski D *et al*: **Crystal structure of the beta(2) adrenergic receptor-Gs protein complex**. *Nature* 2011, **477**(7366):549-555.

146. Wallace AC, Laskowski RA, Thornton JM: **LIGPLOT: a Program to Generate Schematic Diagrams of Protein Ligand Interactions**. *Protein Eng* 1995, **8**(2):127-134.

147. Eyrisch S, Helms V: **What induces pocket openings on protein surface patches involved in protein-protein interactions?** *J Comput Aid Mol Des* 2009, **23**(2):73-86.

148. Krishnamurthy H, Gouaux E: **X-ray structures of LeuT in substrate-free outward-open and apo inward-open states**. *Nature* 2012, **481**(7382):469-U480.

149. Ma D, Lu PL, Yan CY, Fan C, Yin P, Wang JW, Shi YG: **Structure and mechanism of a glutamate-GABA antiporter**. *Nature* 2012, **483**(7391):632-636.