Characterizing Pockets on the Surface of Proteins by Computational Tools

Dissertation
Zur Erlangung des Grades des Doktors der Naturwissenshaften
der Naturwissenschaftlich-Technischen Fakultät der Universität
des Saarlandes

von
Rahmad Akbar

Saarbrücken
2018

Tag des Kolloquiums: 02.05.2018

Dekan: Prof. Dr. rer. nat. Guido Kickelbick

Berichterstatter: Prof. Dr. Volkhard Helms

Vorsitz: Prof. Dr. Katrin Philippar

Akad. Mitarbeiter: Dr. Karl Nordström

Acknowledgement

I would like to thank Prof. Volkhard Helms for his wisdom, guidance, patience, and generosity. Prof. Daniel Khananshvili for his contributions to our collaboration on Natrium Calcium exchangers. Dr. Siti Azma Jusoh and Prof. Rommie Amaro for their contributions to our collaboration on Nuclear Receptors. My friends, Thorsten, Jan, Mike, and Kerstin for their bountiful mensa sessions and lovely companies. A very special shout goes to Kerstin Gronow-Pudelek for her dedication and many helps during the past years. Without all of you, this dissertion might not exist.

Thank you.

To my family

Abstract

This dissertation compiles works that take advantage of computational tools such as machine learning, molecular docking, and molecular dynamics simulation to study proteins. In the first work, we used molecular dynamics simulation and a machine learning model in the form of naive Bayes classifier to help us discover good virtual screening targets. In the second work, we built naive Bayes and artificial neural network models to help us discriminate and prioritize allosteric targets. Finally, we used molecular dynamics simulation in combination with hierarchical clustering to study the surface dynamics of calcium exchangers. Specifically these methods were used to observe, examine, and characterize the formation and deformation of pockets on the surface of these proteins. We found that the combination of well established structural biology methods such as docking and molecular dynamics simulation with machine learning makes a particularly potent toolset. Such a combination allowed us to train predictive models, generate insights, and explore our data from a fresh perspective.

Kurzzusammenfassung

Diese Dissertation trägt Arbeiten zusammen die computergestützte Werkzeuge wie maschinelle Lernverfahren, Moleküldocking und Moleküldynamik-Simulation nutzen um Proteine zu studieren. Im ersten Projekt nutzten wir Moleküldynamik-Simulationen und ein maschinelles Lernmodell in der Form eines Bayes-Klassifikator um vielversprechende Kandidaten in einen virtuellen Screening zu finden. Im zweiten Projekt konstruierten wir Bayes- und neuronale Netzwerk-Modelle um die allosterischen Ziele von Liganden besser unterscheiden und priorisieren zu könne. Schlussendlich setzten wir Moleküldynamik-Simulationen zusammen mit hierarchischem Clustering ein um die Oberflächendynamik des Natrium Calcium Austauscher zu studieren. Die genannten Methoden wurden gezielt genutzt um die Bildung und Verformung von Taschen auf der Oberfläche dieser Proteine zu beobachten. Wir stellten fest, dass die Zusammenführung etablierter Methoden der Strukturbiologie, wie Docking und die Simulation von Moleküldynamik, mit maschinellen Lernverfahren ein mächtiger Werkzeugkasten darstellt. Die Kombination der verschiedenartigen Methoden erlaubte uns Vorhersagemodelle zu trainieren, dadurch Erkenntnisse zu generieren und unsere Daten aus einem neuen Blickpunkt zu erforschen.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Proteins

Proteins constitute a significant portion of the total mass of dry cells. For instance red blood cells contain about 55% of water at minimum and around 78% at maximum [1]. In the absence of water (when cells are dried), half of their mass is protein [2]. To put this into pespective, it was estimated that a bacterial cell contains 3-4 million proteins, a yeast cell was estimated to contain 100-150 million proteins, and mammalians can contain up to 10^{10} proteins per cell [2]. It is also interesting to point out that as the organisms become more complex (bacteria to mammals) the number of proteins their cells produce grows exponentially (Figure 1.1).

Proteins are large biomolecules (macromolecules) composed of one or several chains of amino acids. In nature, there exist twenty different amino acids with common structural features of a central alpha carbon, an amino group, a carboxyl group, and a variety of side chains. Polymers of these amino acids form the basic linear structure of proteins. Since the amino acids are connected with peptide bonds, this linear structure is also known as polypeptide [3]. In cells, proteins can function as catalysts for critical chemical reactions (enzymes), as messengers (cell signalling), and they can also provide structural stiffness and rigidity to various cell types.

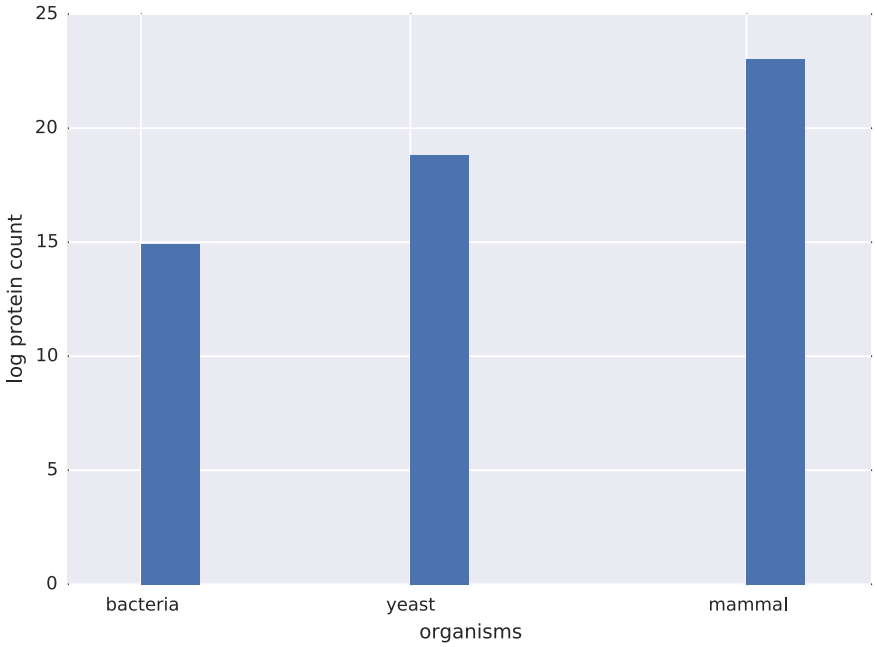


Figure 1.1: The number of proteins in various cells.

1.1.1 Proteins as Enzymes

The first enzyme, amylase, was discovered in Paris by Payen and Perzos [4]. In their paper, published in 1833, they described that from a solution of malt they sucessfully extracted *something* that has the ablility to convert starch into sugar. Furthermore, they also described that when this *something* was exposed to high temperatures (by heating) its ablity to convert starch into sugar ceased. Since it separates sugar from starch, they called it diastase (in greek diastase translates to separation). Over six decades later (in 1898), the last three letters from diastase (-ase) were established as a suffix to form the name of newly found enzymes, a practise that remains relevant until present day.

Today, according to the BRaunschweig ENzyme DAtabase (BRENDA), there are currently 4867 active enzyme classification (EC) classes reported by the International Union of Biochemistry and Molecular Biology (IUBMB), in addition to 871 deactivated or tranferred EC classes [5]. The database alone contains 2.7 million annotated data that include enzyme occurrence, function, kinetics, and molecular properties.

The most abundant enzyme on earth is Ribulose-1,5-bisphosphate car-

boxylase/oxygenase or in short Rubisco [6]. The enzyme is present in plants, cyanobacteria, and other photosynthetic organisms. The most amazing feature of this enzyme, however, is not its abundance or availability but rather its function. Rubisco is the central catalyst for carbon fixation. In orther words, the enzyme is responsible for the process of converting inorganic carbon (CO_2) to organic carbon. Without this enzyme life as we know it would probably not exist.

The substrates for Rubisco's catalytic reaction are ribulose-1,5-bisphosphate and carbon dioxide. The carboxylation step (carboxylase reaction) produces 3-keto-2-carboxyarabinitol-1,5-bisphosphate, a six carbon intermediate, which then decays into glycerate-3-phosphate [7]. The latter molecules can be used in subsequent steps to produce larger molecules such as glucose. While its importance in carbon fixation is unprecedented, Rubisco is pretty inefficient. The enzyme can only catalyse three to ten carbon dioxide molecules per second [8]. This inefficiency is likely due to the difference in atmospheric compositions when the enzyme evolved. When the enzyme first evolved, the earth's atmosphere was deprived of oxygen and the concentration of carbon dioxide was much higher. The need for an efficient catalyst simply did not exist back then. In the current atmosphere, however, the concentration of oxygen is much higher and carbon dioxide is lower. For this reason various methods were attempted to improve Rubisco such as via mutations and hybrid enzymes [9].

On the other end of the efficiency spectrum, an enzyme named 5'-phosphate decarboxylase (ODCase) is crowned as the most effective pure protein catalyst known in nature [10]. The substrate of this enzyme, orotic acid, decarboxylises in solution at room temperature with a half time of 78 million years. This half time dwarfs the life time of humans by a factor of over 10^6. ODCase, fortunately, catalyses the decarboxylation of this molecule by a factor of 10^{17} allowing the carboxylation reaction to happen much much faster. Even more fascinating, the enzyme does not require a metal or any other type of cofactor to function [10].

1.1.2 Proteins in Cell Signalling

Cells influence one another. For example, when a haploid *Saccharomyces cerevisiae* (yeast) is ready to mate, it secretes mating signals to the opposite mating type that trigger a fusion between the two cells creating a diploid cell [11]. These mating signals are commonly several types of small peptides. In higher animals, cells communicate using a more complex set of molecules in favor of several types of small peptides. These molecules include proteins, small peptides, amino acids, nucleotides and etc. Interestingly, no matter

what the signalling molecule is, the receiving cell (target cell) recognizes this signal using a specific protein called a receptor [11]. Receptors specifically recognize and bind the signal molecules. This binding event then triggers a cascade of events (response) in the target cell.

Among the prominent examples of receptors are nuclear receptors. These are receptors that respond to lipid-soluble signals and function as transcription factors responsible for controlling gene expression in numerous biological processes such as cell proliferation, metabolism, and reproduction [12]. Since nuclear receptors are heavily involved in cell propagation and metabolism, they become critical targets for drug discovery efforts to treat cancers and metabolic diseases [13].

Proteins in the nuclear receptor family share a common topology: a highly variable amino terminal domain, a highly conserved central DNA binding domain, a short nuclear localization region, and a conserved carboxy terminal ligand binding domain. The amino terminal domain contains transactivation regions referred to as activation function 1 (AF1), the DNA binding domain includes two zinc fingers (C domain), the localization domain is referred to as the D domain, and the ligand binding domain that hosts the binding pocket is referred to as the E or LBD domain [14].

When a ligand binds to its corresponding binding site, for example in type 1 receptors such as androgen, progesteron, and estrogen receptors, the binding event releases the receptors from their chaperone proteins. Once freed, these receptors form homodimers and localize to the nucleus of the cell. In the nucleus, the ligand-receptor complexes associate with their corresponding coactivators allowing them to bind and activate the target genes [15, 16]. In contrast to type 1 receptors, type 2 receptors readily recide and bind to the DNA even when the ligand is not present and they do not function as homodimers. Type 2 receptors commonly associate with retinoid X receptors to form heterodimers, in the absence of a ligand they carry out active repressive functions [17]. On the other hand, when a ligand is present, these receptors associate with coactivators and typically carry out enzymatic functions such as histone acetyltransferases which help open chromatin structures to activate the target genes [16].

Nuclear receptors also make use of allostery (remote regulation) in exerting their functions. Allosteric regulations are achieved via a short helical construct known as activation function 2 (AF2). In type 2 receptors, the absence of ligand in the binding site allows the AF2 helix to addopt an open conformation enabling the binding of corepressor proteins to the receptor. When a ligand is present, the AF2 helix interacts with coactivator proteins instead of corepressor, allowing activator proteins to interact directly with the ligand binding domain [18].

In chapter 2, we will use nuclear receptors and data from molecular dynamics simulations (section 1.2) in combination with machine learning algorithms (section 1.3) to find better virtual screening targets. Whereas in chapter 3 and chapter 4, we will use machine learning to study allosteric pockets in various proteins and examine the formation/deformation of pockets on the surface of three calcium exchanger proteins, respectively.

1.2 Docking and Molecular Dynamics

1.2.1 Molecular Docking

The term docking summarizes computational techniques aimed at finding the optimal match between two molecules: a receptor and a ligand. In a biomolecular context, the receptor is often a protein of biological significance; the ligand, most often in the form of a small molecule or protein, is the modulator of the target. In the early 1980s, this method was initially intended to reproduce the conformations of protein-ligand complexes determined in the (wet) lab. Ever since, docking has grown as a subfield of its own with important applications in drug discovery. It turns out that finding the optimal match between two molecules is useful for discovering new drug candidates as well. The advent of docking brought about the so called virtual screening (VS) approach in which a library of ligands is screened (docked) against a receptor to find new modulator candidates [19].

Finding the optimal match between two molecules translates into three problems: representation of the system, conformational space search, and scoring of the found solutions.

System Representation

The system is most often represented by geometric features of the molecular surface. The definition of the surface area originates from the solvent accessible surface concept introduced in the 1970s. The accessible surface is produced by a space-filling procedure in which van der Waal's radii are assigned to each atom or group of atoms and the surface is represented by a set of interlocking spheres of appropriate radii. This representation, despite highly dependent on the magnitude of the van der Waal radii, was able to provide a couple of interesting insights; first, it was found that large nonpolar amino acids tend to be buried in folded proteins; second, a reduction of accessibility by a factor of 3 was found when proteins fold from extended to the native folded conformations [20]. The accessible surface definition was further refined by rolling a probe ball over the van der Waal radii creating

the so called Connolly surface [21]. Connolly surfaces were found to produce satisfactory compliments at the interfaces of the molecule [22], a feature essential to molecular recognition problems such as docking. Finally, physicochemical information associated to the surface such as polarities, charges and electrostatics of residues were added to the surface description [23].

Interfaces can be dynamic. Interfaces such as binding sites have been associated with frequent structural instability and can be partly rigid and partly flexible [24]. Hence, the flexibility, to a sufficient extent, needs to be accounted for in the docking procedure. Accounting for the flexibility in a docking procedure is analogous to the induced fit principle whereas the rigid model represents the lock and key model [25]. The flexibility of a receptor has been described primarily by conformational samplings and rotamer libraries. The former represents the flexibility by exploring the conformational space through an ensemble of conformations; docking is carried out on this ensemble instead of single conformers. The ensemble can be obtained by curating relevant structures, from X-ray crystallography or nuclear magnetic resonance (NMR), deposited in the protein data bank (PDB) or from molecular dynamics simulations. The latter accounts for the flexibility by scanning various amino acid side chain orientations from a rotamer library; certain rotameric states have lower energy than others, the most favorable side chain configurations are then selected from this library [26].

Ligands, similar to receptors, can be flexible as well. The naive approach to account for the flexibility is through the ligand's rotatable bonds. However, this approach quickly transforms into an intractable problem since the number of possible conformations grows exponentially in size proportional to the number of rotatable bonds [27]. Monte Carlo simulation and simulated annealing are among the more popular solutions employed to circumvent this problem. For instance, a Monte Carlo approach in combination with a molecular affinity potential were found to be able to recover crystallographic binding modes of a test system comprising phosphocholine and immunoglobin [28]. Fragment based approaches, where the ligand is chopped into smaller pieces and docked separately to the binding site, have also been presented [29]. In addition, genetic algorithms, which incorporate operations like mutations and crosses, to generate flexible conformations, have seen marked successes. In fact, a variation of this approach termed Lamarckian genetic algorithm was reported to outperform both Monte Carlo and fragment based docking [30].

Conformational Search

Conformational space search attempts to locate the most stable docking pose: the global mimimum. This problem can be aproached systematically by scanning the entire solution space. However, much like flexibility, systematic scanning quickly becomes infeasible. For instance, the docking program DOT needed to evaluate 36 billion configurations in order to describe the cytochrome c oxidase system successfully [31]. This amount of computing is prohibitively expensive to most researchers. In particular for virtual screening, where thousands or even millions of compounds need to be docked. The search problem prevails even more prominently in protein-protein docking since the search space in protein-protein docking is often larger than that of protein-ligand docking. Monte Carlo, molecular dynamics and genetic algorithm are, again, relevant in circumventing this problem. The search, in general, proceeds in two steps: generation of populations of solutions and evaluation of the solutions by a certain energy function. The first step could employ Monte Carlo, molecular dynamics or any other approach capable of generating ensembles of conformations; in the second step, the complex energy, comprising the ligand docked to receptor, is evaluated by an energy function. A general form of such an energy function is given in equation 1.1.

$$E = \sum bonds + \sum angles + \sum dihedral \tag{1.1}$$

Scoring

Scoring discriminates between good and bad solutions. The complex with the lowest energy is, ideally, the solution. However this is not always the case since approximations and random elements are inherent in most parts, if not all, of the docking pipeline. For instance, the surface representation incorporates geometric approximations. Also, generating ensembles in the search procedure, in particular the one involving Monte Carlo, contains random elements. In addition, docking algorithms often produce a large number of solutions which are difficult to manage without some form of scoring and ranking [32]. Free energy calculations employing Monte Carlo or molecular dynamics simulation have been developed to score and rank docking poses [33], however, such calculations are often prohibitively expensive computationally. Hence, most docking programs incorporate some degree of aproximation; docked poses are often scored by binding energies estimated from a linear combination of pairwise receptor-ligand terms such as van der Waals (vdW), hydrogen bond (hbond), electrostatic, torsion and estimates of the solvent contribution [34]. A general form of a binding energy function is

given in equation 1.2. Once the score, namely the binding energy, for each pose has been computed, the values are then sorted to produce the ranking. Accurate scoring and computational efficiency is a trade off. For instance, free energy calculations are able to discriminate between good and bad poses quite successfully at the expense of large computational resources [35]. On the other hand, scoring functions implemented in many docking programs make various simplifications and assumptions while at the same time strive to provide a reasonable approximation of the binding energy at a much lower computational cost [19].

$$\Delta G_{bind} = \Delta G_{vdW} + \Delta G_{hbond} + \Delta_{electrostatic} + \Delta G_{torsion} + \Delta G_{solvent} \quad (1.2)$$

The docking landscape is actively evolving. A recent review on docking reported that over 10 new docking programs and over 20 new scoring functions have been made available from 2012 to 2013 [36]. The majority of efforts to improve docking results are focused on better considering flexibilities in both receptors and ligands. This is often achieved by generating ensembles that better represent the receptors and ligands. For instance, RosettaDock introduced a fragment based sampling method termed "shotgun" in which the ligand is chopped into fragments much like a bullet is dispersed by a shotgun. Later, a temperature-replica exchange Metropolis-Monte Carlo method was implemented to account for the flexibility. This approach was found to be able to generate a higher fraction of near-native conformations compared to the former [37]. Other recent approaches to account for the flexibility include: a combination of coarse grained and atomistic simulations termed "mixed-resolution modeling" [38], an induced fit docking involving quantum terms [39], a combinatorial arrangement and clustering of side chains [40], a graph theory based optimization [41] and a multi stage backbone reconstruction [42]. On the scoring front, quantum mechanics are becoming more and more prevalent. For instance, the application of a quantum mechanics based linear interaction energy to account for the flexibility improved the correlation between calculated and experimental binding affinities by almost 30%, from 0.66 to 0.91 [43]. Calculations involving quantum terms provide additional details compared to the general pairwise terms seen in equation 1.2 at a higher computational expense. Despite this, the advancement of modern computing facilities allows complicated computations, such as calculations involving quantum terms, to be completed in shorter and shorter time.

1.2.2 Molecular Dynamics

Richard Feynman, a recepient of the 1965 Nobel Prize in Physics, once wrote: "Everything that living things do can be understood in terms of the jigglings and wigglings of atoms". The simulation method termed molecular dynamics simulation (MD) models the dynamic phenomena of molecular systems. It is essentially the tool to capture this atomic jigglings and wigglings of biomolecular systems, namely proteins. In principle, atomic motions are best described by quantum mechanics in which motions are governed by probability functions and bonds are not formed mechanically, instead they are formed by shifting electron clouds that are simultaneously waves and particles. These quantum properties, similarly discussed in the molecular docking section, provide much details at a higher computational cost. The computational cost is particularly taxing when large systems like proteins are the primary interest. An alternative to the quantum aproach, introduced in the 1950s [44], uses Newton's equations of motions to describe the atomic motions instead. The Newton equation of motions serves as an aproximation to the quantum calculation and comes at a much cheaper computational cost. This type of MD is often referred to as the classical MD and is the method of choice in this thesis; here, it will simply be referred to as MD.

MD translates, primarily, to three parts: the molecular system representation (potential), the calculation of molecular forces acting on each atom of the system (force) and the calculation of the new atomic positions in response to the acting forces (trajectory). The latter two processes are iterated over the desired simulation length often in the order of hundreds of nanoseconds(ns). Simulations in the order of miliseconds performed on specialized computing platform have also been described recently [45].

System Representation

A molecular system is represented by a collection of atoms. The definition of an atom in a molecular mechanics forcefield is an aproximation of its quantum mechanics counterpart; instead of representing an atom as a nucleus surrounded by an electronic cloud, the nucleus and the electrons are conceptually separated. The nucleus motions are then treated as it is moving on an averaged electron density; consequently when viewed in this manner, the dynamics of the nucleus can be determined by a potential energy surface without accounting for the electrons explicitly. This approximation is known as the Born-Oppenheimer approximation [46]. The validity of this approximation is based on the large mass difference between electrons and protons; their dynamics, the electrons and protons, are practically decoupled. With

the potential energy surface available, Newton's equations of motions, known also as classical mechanics, can then be utilized to track the dynamics of each atom; the system dynamics is then the sum of the dynamics of all atoms.

Potential

Atoms are held together by interatomic terms: the potential. Since a molecular system comprises atoms, an individual atom continuously interacts with other atoms. The interactions can be broadly classified into bonded and non-bonded terms. Bonded interactions, most commonly, include bonds, angles and torsions (dihedral); while non-bonded ones are electrostatics and van der Waals interactions. These terms, collectively known as the potential $U(r_i, ..., r_N)$, represent the potential energy of N interacting atoms as a function of positions $r_i = (x_i, y_i, z_i)$. They are indeed reminiscent of terms seen earlier for molecular docking; equation 1.3 shows an example of a mathematical formalization of the potential. The first two terms, physically, describe the deformation energies of bond lengths and bond angles from their equilibrium values denoted as l_{i0} and θ_{i0}. The third term describes rotations around chemical bonds; the fourth captures the van der Waals repulsive and attractive interatomic terms; lastly, the fifth term describes the coulombic electrostatic forces between two atoms [47].

$$
\begin{aligned}
U(r_i, ..., r_n) = & \sum_{bonds} \frac{a_i}{2}(l_i - l_{i0})^2 \\
& + \sum_{angles} \frac{b_i}{2}(\theta_i - \theta_{i0})^2 \\
& + \sum_{torsions} \frac{c_i}{2}\Big(1 + cos(n\omega_i - \gamma_i)\Big) \\
& + \sum_{atompairs} 4\epsilon_{ij}\Big((\frac{\sigma_{ij}}{r_{ij}})^{12} - (\frac{\sigma_{ij}}{r_{ij}})^{6}\Big) \\
& + \sum_{atompairs} k\frac{q_i q_j}{r_{ij}}
\end{aligned}
\tag{1.3}
$$

Forcefield

The problem of finding a realistic potential is essentially a forcefield parameterization problem. Given the fomalization in equation 1.3, parameters such as a, b, c, l_{i0}, θ_{i0}, ϵ, q and etc., need to be estimated for each atom present in the system. Collectively, the mathematical formalization and the parameter

sets form the so-called forcefield. The parameters are usually obtained empirically from small organic molecules since these molecules are more amenable to experimental studies and complex quantum calculations. In addition, parameters could come from different types of experimental data and/or different emphasis giving rise to different forcefields. For instance, among the more popular forcefields such as Amber, CHARMM, GROMOS, and OPLS-AA, similarities and and differences exist. Similarities are most prominently seen in the parameterization of bond lengths and angles; parameters for bond lengths and angles for OPLS-AA are optimized based on the same infrared spectra data used to optimize Amber, identical data are used to optimize CHARMM as well [48]. Moreover, parameterizations of the van der Waals term is done in similar fashion across all four forcefields, this term is developed without the explicit consideration of interaction sites for nonpolar hydrogen [49]. On the other hand, improper dihedrals are accounted for differently; while Amber and OPLS-AA account for this quantity by including it in the dihedral term, CHARMM and GROMOS add a separate quadratic quantity to the potential function [50].

Variations in the parameter optimization of the forcefields are reflected in subsequent MD simulations. In a comparative study involving 10 different forcefields, 8 out of 10 forcefields were able to succesfully recover the test protein's native conformation; despite so, marked differences among the recovered conformations from different forcefields were apparent [51]. In practice, most forcefields would provide a good approximation of the potential. However, some systems are better represented by a certain forcefield. Hence, choosing the most relevant forcefield for a given task remains of crucial importance.

Force

The force is the negative gradient of the potential. The second component of MD is the force that acts on the atoms. Force, in the context of potential energy surfaces, is the slope of the surface, hence, forces can be computed by taking the partial derivative of the potential function U with respect to positions r. This quantity is formalized in equation 1.4. Force, from Newton's second law, is also equal to the product of mass and acceleration and is formalized in equation 1.5. Combining the two expressions brings us equation 1.6. This final expression relates, nicely, the force F, position r, and time t; the ingredients for the third component of MD: the trajectory.

$$\begin{aligned} F_i &= -\nabla_{r_i} U(r_i, ..., r_N) \\ &= -\Big(\frac{dU}{dx_i}, \frac{dU}{dy_i}, \frac{dU}{dz_i}\Big) \end{aligned} \tag{1.4}$$

$$\begin{aligned} F_i &= m_i a_i \\ &= m_i \frac{d^2 r_i}{dt^2} \end{aligned} \tag{1.5}$$

$$F_i = -\nabla_{r_i} U(r_i, ..., r_N) = m_i \frac{d^2 r_i}{dt^2} \tag{1.6}$$

Trajectory

Solving Newton's equation of motion yields the trajectory. The third and final component of MD is obtained by solving equation 1.6. However, an analytical solution is not available, since the problem includes a position term in the order of $3N$. Fortunately the solution can be obtained numerically. One approach to this is the Verlet algorithm; equation 1.7 shows the basic formulation of this algorithm. In this expression, the position of an atom as a function of time $r_i(t + \Delta t)$ relates to force F_i and mass m_i, Δt is the time step. Forces and mass can be obtained from the potential and forcefield. The largest possible value for Δt is, in practice, determined by the fastest motions in the system. For instance bonds involving light atoms such as O–H vibrate in the order of several femtoseconds, hence, Δt should be in the order of sub O–H vibration. Due to this, the time step for a typical MD simulation is often in the order of 2 femtoseconds if bond lengths are constrained to their equilibrium values. Otherwise, it is in the order of 0.5 femtoseconds. The Verlet algorithm is simple, stable, and commonly used in MD simulations. That said, alternatives to integrate the Newton's equation of motion such as leap-frog algorithm, velocity Verlet, and Beeman's algorithm have also been devised [46].

$$r_i(t + \Delta t) \approx 2r_i(t) - r_i(t - \Delta t) + \frac{F_i(t)}{m_i} \Delta t^2 \tag{1.7}$$

MD Programs

Forcefields are integrated in various MD simulation software packages. Initially, forcefields were developed for their specific molecular mechanics and

dynamics software packages. For instance, the Amber, CHARMM, and GROMOS forcefields were primarily developed in the context of the respective Amber, CHARMM and GROMOS MD packages. OPLS-AA was intended for BOSS and MCPro [50]. Integration of various forcefields into a single program, fortunately, has been succesfully carried out and made available for academic use at no cost. One MD package named GROMACS illustrates this nicely. The software handles a wide variety of molecular systems such as proteins, nucleic acids, and lipids. It includes the most commonly used forcefields, implicit, and explicit water models. On the computing front, efficient parallelization and algorithms have been implemented. Moreover, implementations of the virtual site algorithm permit the removal of hydrogen atoms degree of freedom enabling integration time steps to up to 5 femtoseconds [52]. These qualities allow GROMACS to perform robust and fast calculations on a wide variety of systems making it a very attractive toolkit for carrying out MD simulations of biomolecular systems.

1.3 Machine Learning

Machine learning imparts the ability to learn to computers without explicitly being programmed. As A.L. Samuel described in an article published by the IBM journal in 1959, computers can be programmed to learn to play the game checkers better than the person who created the program. Not only the computer learned to play the game in a short period of time (10 hours), it learned the rules (parameters) of the game eventhough the relative signs and weights of these parameters were not specified or unknown [53].

Learning to recognize patterns and rules in games, while often used to illustrate advancements in the machine learning field, is just one way to demonstrate the capacity of machine learning algorithms. Machine learning algorithms have been used to learn and regocnize patterns on problems involving satellite images, text, drug design and many others. For instance, Wieland and Pittore [54] used four different machine learning algorithms, Normal Bayes, K Nearest Neighbors, Random Trees, and Support vector machines to recognize built-up areas (urban patterns) from medium resolution (MR) and very high resolution (VHR) satellite images. On the other hand, Pang and coworkers [55], used Naive Bayes, Maximum Entropy Classification, and Support Vector Machines to gain sentiments from a population of texts. As a final example, Burbidge and coworkers [56] used Neural Networks, Decision Tree, and Support Vector Machine to predict the inhibition of dihydrofolate reductase, an enzyme that is critical to cell proliferation and growth.

Broadly, machine learning algorithms can be categorized into two distinct categories: supervised and unsupervised learning. The former requires the data to have some sort of label (annotation/supervision). The algorithm learns to recognize patterns that would enable it to minimize mistakes (error) between its predictions and the original labels. The latter group of algorithms do not require annotation (no supervision) in the data, making such algorithms very useful for exploring and recognizing the underlying classes (clussters) in the data. Examples of supervised learning algorithms include Artificial Neural Networks, Naive Bayes Classifier, Support Vector machines. Whereas unsupervised learning algorithms include Hierarchical Clustering, Principle Component Analysis and Density Based Clustering, among others. Naive bayes, Artificial Neural Networks, and Hierarchical Clustering are among the algorithms utilized in the comming chapters (chapter 2 to chapter 4), hence we will discuss them a bit deeper in the comming sections.

1.3.1 Naive Bayes Classifier

The Naive Bayes Classifier discriminates between two classes by comparing the probabilities of observing these classes. More specifically, it uses Bayes theorem to obtain the probability of observing a class conditioned by the given data [57]. Equation 1.8 states the theorem mathematically.

$$P(C_k|D) = \frac{P(C_k)P(D|C_k)}{P(D)} \tag{1.8}$$

Here, C_k is the class k, D is the data, $P(C_k|D)$ is the probability of observing a class k given the data, and $P(D|C_k)$ is the probability of observing the data in the class k.

The terms $P(C_k|D)$, $P(C_k)$, $P(D|C_k)$, and $P(D)$ are also known as *posterior, prior, likelihood,* and *evidence*, respectively. For multidimensional data where the data contain more than one random variable, the numerator (*prior* and *likelihood*) is equal to the joint probability model defined in equation 1.9. Following the chain rule for conditional probability, the joint probability model is summarized in equation 1.10. By naively assuming the variables to be independent, the joint probility model becomes much simpler as shown in equation 1.11.

$$P(C_k, d_1, ..., d_n) \tag{1.9}$$

$$\begin{aligned} P(C_k, d_1, ..., d_n) &= P(d_1, ..., d_n, C_k) \\ &= P(d_1|d_2, ..., d_n, C_k)P(d_2, ..., d_n, C_k) \\ &= P(d_1|d_2, ..., d_n, C_k)P(d_2|d_3, ..., d_n, C_k)P(d_3, ..., d_n, C_k) \\ &= ... \\ &= P(d_1|d_2, ..., d_n, C_k)...P(d_n|C_k)P(C_k) \end{aligned} \tag{1.10}$$

$$\begin{aligned} P(C_k, d_1, ..., d_n) &= P(d_1, ..., d_n, C_k) \\ &= P(C_k)\prod_{i=1}^{i=n} P(d_i|C_k) \end{aligned} \tag{1.11}$$

Eventhough the Naive Bayes Classifier is a rather simple classifier, it often performs respectably in many scenarios including difficult ones. For instance, Rosen and coworkers [58] successfully used a Naive Bayes Classifier to classify metagenomic reads to their optimal taxonomic match. Their results not only demonstrate that the classifier managed to assign reads from next generation sequencing technologies (a very high-dimensional data) to their taxonomic classes correctly, they also found that the classifier was able to identify significant number of genera that were missed by other classifiers.

1.3.2 Artificial Neural Networks

In 1943 McCulloh and Pitts [59] introduced the mathematics and algorithms to computationally model neural networks. They described in their paper that neural events and the relations among these events can be treated by propositional logics. This is due to the "all-or-non" (activation or deactivation) characteristic of the nervous activities. Almost two decades latter, Frank Rosenblatt put forward a hypothetical nervous system called perceptron to capture how information is sensed, remembered, and retained. The model allowed him to predict learning curves from neurological variables and helped him understand the organization of cognitive systems [60]. Works laid out by McCulloh, Pitts, and Rosenblatt over a half century ago are the foundation of modern artificial neural networks.

A perceptron is a network model that takes several binary inputs and returns a single binary output. The model uses weights to express the importance of the input variables, the weighted sum of these inputs (z) is returned at the output neuron of the network. A threshold value is then used to determine whether the output is 0 or 1. Both the weights and the threshold

are real numbers, together they form the parameters of the network. Figure 1.2 illustrates a perceptron containing one neuron with three input variables and equation 1.13 summarizes the algebraic form of the output. Varying the weight of each input variable allows one to control the contribution of individual variables in the network. Similarly varying the threshold value allows one to fine tune the outputs.

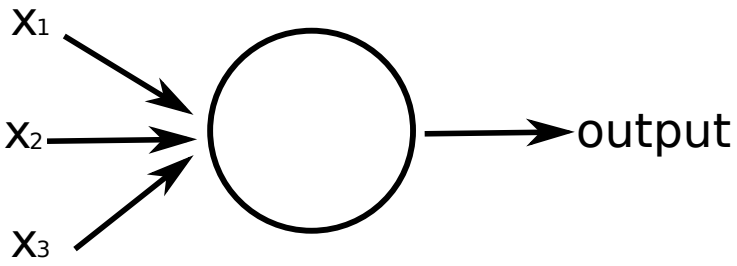


Figure 1.2: A perceptron with a single neuron.

$$z = \sum_i w_i x_i \tag{1.12}$$

$$\text{output} = \begin{cases} 0, & \text{if } z \leq \text{threshold} \\ 1, & \text{if } z > \text{threshold} \end{cases} \tag{1.13}$$

Modern artificial neural networks use sigmoid neurons in their implementation. A sigmoid neuron uses a sigmoidal function as its activation function. More specifically this type of neuron takes the weighted sum z as an input and feeds it to a sigmoid funtion (equation 1.14). While simply using z in combination with a threshold value to decide an output (as in perceptron) is a simple and elegant solution, this approach forces the network to make a binary decision at every node. Hence a slight change in one of the neurons could cause the network to output large differences at the final output neuron. A sigmoid function, on the other hand, is a smooth function. Thus, a network equiped with sigmoid neurons can capture and reflect subtle changes in the neurons in a much better fashion.

$$\sigma(z) = \frac{1}{1 + \exp(-z)} \tag{1.14}$$

One other aspect crucial to artificial neural networks is the weight optimization procedure. To optimize the weights in a network, a procedure called gradient descent has become the de facto standard. Gradient descent, as the name suggest, uses the gradient of the cost function with respect to

each weight in the network to find the optimal set of weights. The sum of squred error (SSE) function (equation 1.15) is often used as the cost function whereas the gradients are obtained using a computationally efficient procedure called backpropagation. Equation 1.16 illustrates a typical weights optimization in an artificial neural networks model.

$$\text{SSE} = \sum_{i}^{n} (y_i - \hat{y}_i)^2 \tag{1.15}$$

Here, y_i is the true value and $\hat{y}_i$ is the predicted value.

$$\text{W} = \text{W}_{prev} - \eta \nabla C \tag{1.16}$$

Here, W are the current weights, W_{prev} are the weights from the previous iteration, and ∇C is the gradient of the cost function.

Today neural network models are being used to solve diverse problems involving drug discovery, natural languange processing, self driving cars and many others. For example Guerra and coworkers [61] used artificial neural networks to find novel antiproliferative drugs for *Trypanosoma cruzi* infection, a parasite that infects seven million people in twenty one countries of the American continent. In finding these novel antiproliferatives, they used an artificial neural network model of three layers with a 4-4-1 architecture. The model was able to predict anti *T. cruzi* activities with an accuracy of 78%. Wehrmann et al. [62], on the other hand, used deep convolutional neural networks to figure out the sentiment of a sentence written in distinct (mulitple) languages. Their model is capable of learning latent features of these languages while at the same time requires substantially smaller set of parameters. In addition, the model allowed them to skip a prerequisite machine translation step, a rather computationally expensive process on its own. Finally, a neural network based self driving system called PilotNet [63] was demonstrated to be able to perform lane keeping in a wide range of driving conditions. The system learned on road images and steering angles generated by humans. It was able to derive the necessary domain knowledge from this data, eliminating the need for human engineers to foresee the complete set of rules for safe driving.

1.3.3 Hierarchical Clustering

Hierarchical clustering, as the name suggests, clusters objects (samples) by building a hierarchy of cluster out of them [64]. More specifically, the clus-

tering procedure starts by regarding each sample as distinct cluster then iteratively merges these clusters into one giant cluster. Since this type of clustering works from the bottom (single clusters) and moves up to build the hierarchy, it is also known as agglomerative clustering. On the other hand, a method that starts at the top where all samples are treated as one giant cluster then splits this cluster into smaller ones are termed divisive clustering.

Regardless of the clustering type, agglomerative or divisive, there are two important metrics that dictate the outcome of a clustering procedure: the distance measure (similarity) and the linkage criterion. The distance measure provides a way to quantitatively evaluate a sample in comparison to other samples. When two samples have small distance, they are thought to be similar to each other, hence, they can be grouped as a single cluster. The Euclidian distance (equation 1.17) is one example of a distance metric. Here, the distance between two samples is simply defined as the squared difference between them. Another example of a commonly used distance metric is the Minkowski distance defined in equation 1.18. This distance metric is regarded as the generalization of both the Euclidian and Manhattan distance since one can switch between the two metrics by adjusting the order p.

$$d(a, b) = \sqrt{\sum_i (a_i - b_i)^2} \tag{1.17}$$

$$d(a, b) = \sum_i \left(|a_i - b_i|^p\right)^{1/p} \tag{1.18}$$

Here a and b are the samples and p is the order of the function.

The linkage criteria provide a systematic way to merge two clusters into a larger cluster. In complete-linkage (maximum-linkage), the criterion to merge two clusters is the farthest distance between them, hence this method is also known as the farthest distance clustering. The single-linkage (minimum-linkage) clustering, on the other hand, merges two clusters by the shortest distance between them. Finally, the average-linkage (mean-linkage) uses the average distance between two clusters to merge them. The clustering results are then visualized using a dendrogram. Depending on the choice of distance and linkage criteria, different clusters can be obtained from the same dataset. Figure 1.3 illustrates a dendrogram on the Iris dataset, a commonly used dataset to evaluate clustering algorithms [65].

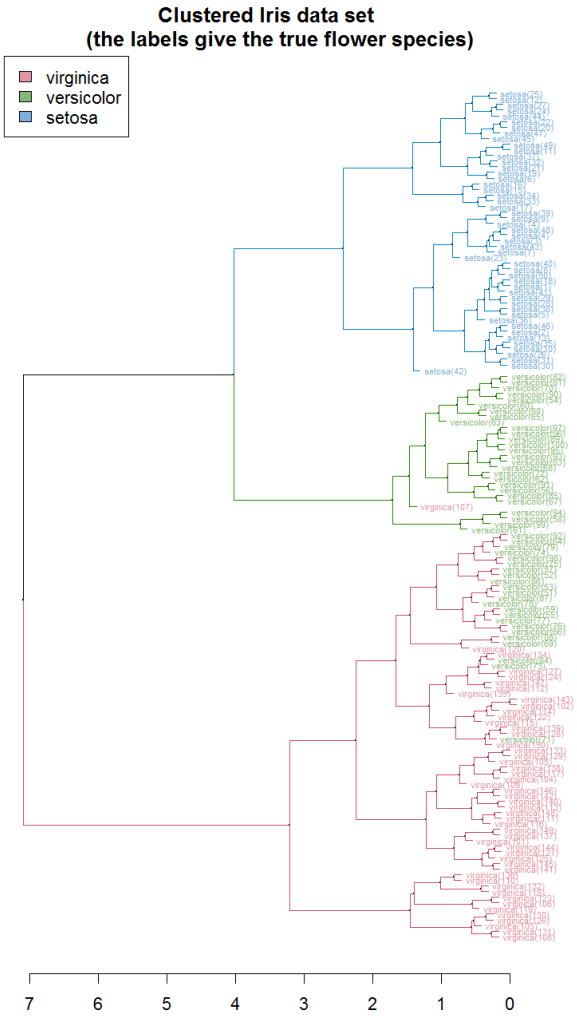


Figure 1.3: A dendogram on the Iris dataset [65].

In addition to grouping samples into clusters, hierarchical clustering is also useful for selecting features in complex datasets. For instance, Kasai et al. [66] used hierarchical clustering in combination with principle component analysis to merge clusters based on similarity. Using such a method, they were able to reduce the amount of wavelenght bands for mature cancer cell analysis from 256 to 8 bands. Gupta et al., on the other hand, used hierarchical clustering to identify B cell clones. Their single-linkage clustering model was able to recognize these clones with over 99% specificity. Furthermore, the model also recorded over 99% sensitivity and positive predictive value.

Chapter 2

Finding Good Virtual Screening Targets

This work was published in 2017 under the title "ENRI: A tool for selecting structure-based virtual screening target conformations" (*Akbar, R., Jusoh, S. A., Amaro, R. E. and Helms, V. (2017), ENRI: A tool for selecting structure-based virtual screening target conformations. Chem Biol Drug Des, 89: 762-771. doi:10.1111 cbdd.12900*). The developed programs and dataset were made available to public via github at `https://github.com/fibonaccirabbits/enri`. Volkhard Helms supervised the work. Siti Azma Jusoh and Rommie Amaro provided the MD snapshots and performed the virtual screening experiments. I developed the approach, implemented the machine learning model, and wrote the manuscript.

2.1 Introduction

In structure-based virtual screening (SBVS), the capacity of an arbitrary protein conformation to enrich a library of ligands with active compounds is not known *a priori*. As the number of known receptor conformations grows, the challenge becomes even larger as an increased conformational space poses selection problems. Indeed the protein data bank (RCSB PDB) has seen an exponential growth since early 90s [67]; the data bank holds over 100 thousand conformations as of this writing. In addition, taking into account the dynamics of a potential target is often useful for drug discovery since such dynamics relate to the induced-fit paradigm of protein-ligand interactions and/or reflect the intrinsic plasticity of a protein's binding site [68–71]. Methods such as molecular dynamics (MD) simulations can be used to generate an ensemble of conformations from a single crystallographic protein

conformation to sample its dynamics. Despite the growth in the number of experimentally determined protein conformations and the ability to generate ensembles of conformations through simulations, it remains a large challenge to infer, *a priori*, the performance of these conformations in an SBVS campaign.

Machine learning algorithms are useful in dealing with such selection problems by assigning classes to objects (instances). The process of assigning distinct classes such as active-inactive or high-low to objects is termed *classification* while numerical assignments are referred to as *regression* [72]. One group of applications of machine learning in drug discovery is the so-called quantitative structure-activity relationship (QSAR) models. Here, algorithms such as support vector machines (SVM) [73], decision trees [74], artificial neural networks (ANN) [59] and naive bayes classifier (NBC) [75] capture relationships between structural and physico-chemical descriptors of ligands and activity measures. Similarly for SBVS, one can relate descriptors of targets and their activity measures using a set of machine learning algorithms. The resulting descriptors-activity relationship can then be used to guide the selection of relevant receptor conformations.

Imbalance is a common property of data sets and presents a common challenge in both VS (ligand-based) and SBVS (target-based) data sets. As an example, PubChem [76], an open repository for small molecules, currently hosts over 61 million unique compounds. Of those, only around one million compounds are annotated as active. In other words, inactive compounds outnumber active compounds by a factor of 60. On the receptor side, intrinsic plasticities of a protein seem to influence the number of conformations capable of improving enrichment in a corresponding SBVS. An MD simulation study [70] recently addressed this point for two proteins of different plasticities. In the flexible HIV-1 reverse transcriptase system, around 50% of the conformations sampled from the simulations were able to outperform the reference X-ray structure. Imbalance, however, was prominently observed in the rigid cytochrome *c* peroxidase system where only 12% of the conformations outperformed the corresponding reference structure.

Imbalance in the data can be dealt with independently from the class assignment task. This is often done through sampling procedures such as under-sampling or over-sampling [77]. Under-sampling reduces the size of the majority class to balance the classes while over-sampling inflates the minority class. A simple random resampling procedure creates new objects by simply adding (over-sampling) or removing (under-sampling) objects randomly from a data set. More sophisticated approaches "literally" generate new objects. For instance, the synthetic minority over sampling-technique (SMOTE) takes a random set of neighbors of an object and synthesizes a new object by using

information derived from the neighbor set [78].

In this work, we present ENRI, a tool that combines a SMOTE and an NBC procedure to infer the potential performance of a receptor/protein conformation to enrich an SBVS campaign. The program uses binding pockets on protein surfaces as its primary data. Descriptors of these pockets were obtained using DoGSiteScorer [79] and enrichment measures were computed using Maestro [80]. SMOTE is employed to appropriately resolve imbalance in the data. Relationships between pocket descriptors and enrichment measures are then captured by the NBC. To demonstrate the usefulness of the program, we trained ENRI on conformational ensembles of eleven nuclear receptors. The best performing NBC model was then used to assign classes (high-low) to conformations from MD simulations. The conformations were classified as either *high* or *low* for enriching and non-enriching, respectively. ENRI enabled us to infer, *a priori*, the performance of these conformations for the corresponding SBVS campaigns satisfactorily well. In future, ENRI can be trained on other data sets (e.g., other receptors), providing means to find enriching conformations beyond nuclear receptors.

2.2 Methods

2.2.1 Data

The training data set comprised a mixture of 421 conformations from MD simulations and crystal structures of 11 nuclear receptors. Table 2.1 lists the protein names and the number of conformations considered for each system. We docked each conformation with the corresponding set of active and decoy ligands and labeled the conformation as *high* or *low* based on its enrichment factor (EF) which were calculated by only considering the top one percent of the corresponding SBVS results (EF1%). A target conformation would be labeled as *high* if its EF1% was larger or equal to the reference X-ray structure or was labeled as *low* if its EF1% was lower than the reference structure. Equation 2.1 defines the formula for an EF calculation. Docking experiments, scoring and EF1% calculations were performed using Maestro [80]. Active and decoy ligands were obtained from the DUD-E database [81].

$$EF = \frac{actives\_in\_sampledset/sampledset}{total\_actives/total\_ligands} \tag{2.1}$$

Table 2.1: PDB ID and the number of *high-low* conformations and active-decoy ligands.

protein name: pdbid	high	low	active	decoy
Glucocorticoid receptor: 3bqd	2	39	258	15185
Retinoid X receptor-alpha: 1fm9	0	40	131	7707
Peroxisome proliferator activated receptor-alpha: 2p54	11	30	373	19831
Estrogen receptor-alpha: 1sj0	33	8	383	20818
Peroxisome proliferator activated receptor-beta: 3sp9	21	10	240	13232
Mineralocorticoid receptor: 2a3i	10	35	94	5240
Estrogen receptor-beta: 2fsz, 3omq, 2yjd, 2jj3, 1zaf, 2nv7, 3oll, 4j24, 1qkm	9	30	367	20313
Androgen receptor: 3l3x	18	15	269	14503
Progesterone receptor: 3kba	0	30	293	15814
Peroxisome proliferator activated receptor-gamma: 2gtk	17	23	484	25867
Thyroid hormone receptor-beta: 1q4x	0	40	103	7653

2.2.2 Pockets and descriptors

Pockets on the surface of the nuclear receptors were identified using DogSiteScorer [79] along with the corresponding descriptors. Default parameters for the pocket detection algorithm, grid spacing and contour level cut-off were employed. The tool developed in this work, ENRI, was then used to filter only pockets that overlap with the bound ligand in the reference X-ray structure. ENRI interfaces with DoGSiteScorer to automatically generate pockets and descriptors for conformations in the training data set.

2.2.3 Handling imbalance in the data

To handle imbalanced data we employed a variation of SMOTE termed adaptive synthetic sampling (ADASYN). ADASYN is a systematic procedure that generates synthetic data by taking into account the minority class distribution [82]. The algorithm proceeds in four steps:

1. G, the total number of synthetic samples needed to balance the classes is computed. Our data consisted of only two classes: *high* and *low*; the minority class was the former. The total number of needed samples was obtained by taking the difference between the majority class and the minority class:

$$G = (|S_{maj}| - |S_{min}|) \times \beta \tag{2.2}$$

   $\beta \in \mathbb{R}$ is a tuning parameter used to determine the desired amount of synthetic samples. S_{maj} and S_{min} correspond to the *high* and *low* classes, respectively.

2. Γ_i is computed by taking the fraction of samples belonging to the major class (Δ_i) in K-nearest neighbors (K) of a minority sample normalized by a constant Z. Each minority sample x_i was associated with one Γ_i.

$$\Gamma_i = \frac{\Delta_i / K}{Z} \tag{2.3}$$

3. The number of synthetic samples g_i that are required for each minority sample x_i is computed.

$$g_i = \Gamma_i \times G \tag{2.4}$$

4. g_i synthetic samples $x_{new,i}$ are then generated for the corresponding minority sample x_i by the equation:

$$x_{new,i} = x_i + (\hat{x_i} - x_i) \times \delta \tag{2.5}$$

Where $\delta \in [0, 1]$ is a random real number and $\hat{x_i}$ is a random neighbor of the sample.

2.2.4 Class assignments

Once the data had been balanced or over-sampled, NBC parameters were estimated on this data. We assumed here that the descriptors are normally distributed. Hence, for each descriptor, an NBC requires mean (μ) and standard deviation (σ) for both the *high* and *low* populations. Class labels, *high* or *low*, were assigned by computing the ratio of joint probabilities of observing a sample in the *high* and *low* populations (P_{ratio}).

$$P_{ratio} = \frac{P_{high}(x_{d1}, ..., x_{dp})}{P_{low}(x_{d1}, ..., x_{dp})} \tag{2.6}$$

A sample was labeled as *high* if its P_{ratio} was larger than 1 and low otherwise. Probabilities were computed using a cumulative distribution function (CDF) of a normal distribution.

2.2.5 Prioritizing conformations (ranking)

Two simple schemes were employed to rank the labeled conformations. Both schemes utilize P_{ratio} values which were computed during the class assignment step. The first ranking scheme simply sorts the conformations based on the P_{ratio} values from largest to smallest. The second scheme weights the P_{ratio}

values by the probability of observing the sample in the *high* population, i.e., the weighted probability ratio (WP_{ratio}).

$$WP_{ratio} = P_{high} \times P_{ratio} \tag{2.7}$$

The top 10 conformations from the ranked list along with their predicted label, P_{ratio}, and WP_{ratio} were subsequently written to an output file.

2.2.6 TPR and FPR

False positive rate (FPR) and true positive rate (TPR) are established criteria to measure classifiers performance in an objective manner. FPR is the fraction of wrongly classified negative samples (false positive) over all the negative samples. This metric is also known as *1-specificity*; the smaller the number the more specific the classifier. TPR, on the other hand, is the fraction of correctly classified positive samples over all positive samples, also known as *sensitivity*; the higher the better. A desirable classifier provides a good compromise between sensitivity and 1-specificity (or specificity for that matter).

2.3 Results and discussion

2.3.1 Pocket Descriptors

DoGSiteScorer returned a set of 65 descriptors for each protein-ligand conformation. 14 of these such as pocket coverage (poc_cov), volume, depth, etc. are real numbers whereas 49 descriptors like amino acid counts, hydrogen bond donor (donor), etc. are integers. Amongst the integer descriptors, 16 were zero descriptors i.e., contained only zero values. Comprehensive definitions of these descriptors are given in the original publications [79, 83, 84].

At the moment, ENRI uses only continuous descriptors. This is due to the way that ADASYN generates synthetic samples. The final step of the ADASYN algorithm, defined in equation 2.5, involves a multiplication of an Euclidian distance between an object and its neighbor with a variable δ. This variable is a random real number between zero to one, hence this step will transform any values into continuous numbers. Due to this we decided to focus exclusively on continuous descriptors for the remaining part of this work. Distributions of these descriptors are shown in Figure 2.1.

Apparently the distributions of samples from the *high* class overlap strongly with those from the *low* class (e.g., the green histograms are mostly shaded

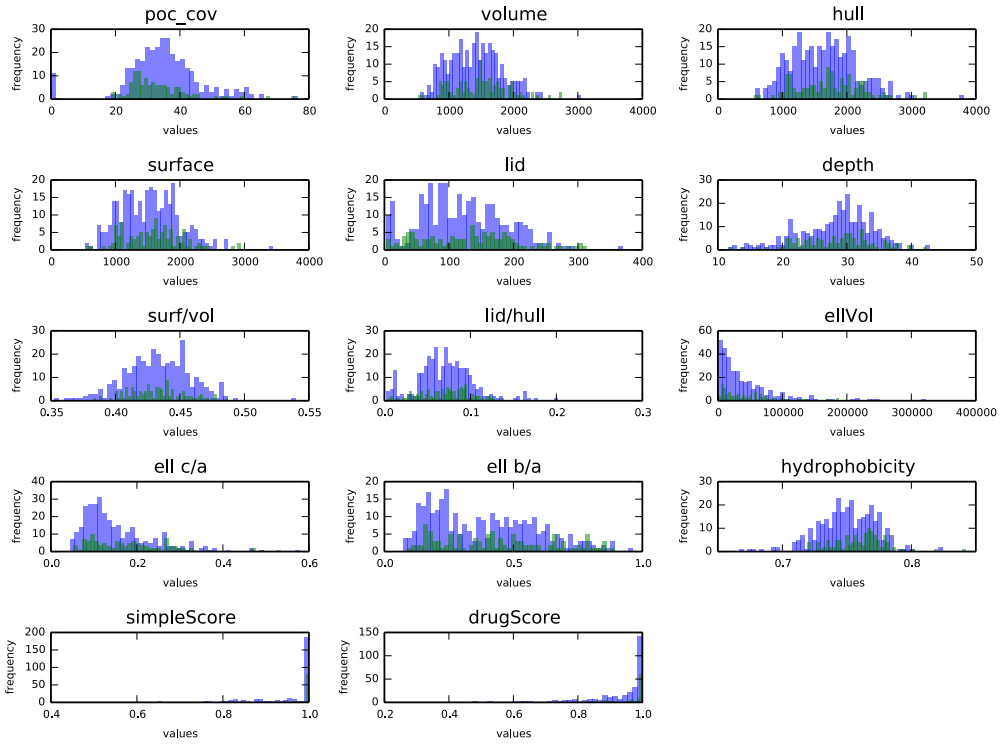


Figure 2.1: Distributions of continuous descriptors. Green and purple represent samples from *high* and *low* classes, respectively. Dark-green areas are the overlapping portions of the distributions.

in dark-green in Figure 2.1). This indicates that samples from the two classes are very similar to each other, at least based on the current set of descriptors. The ratio between *high* and *low* classes was approximately 1:3.

2.3.2 Discriminating classes using the data

To assess the capacity of the data in discriminating the underlying *high* and *low* classes, we trained a NBC on this data and evaluated the performance of the resulting classifier using cross-validations. The two parameters, μ and σ, required by the NBC were estimated for both classes. These parameters were used to compute the probability of observing a sample for each descriptor for each class. Doing so provided us with a set of probabilities, one for each class. The joint probability of observing a sample in each class was computed along with the corresponding P_{ratio} and WP_{ratio}. A sample was labeled as *high* if its P_{ratio} value was larger than one. This process was then iterated over each

sample in the data. Cross-validation was carried out by dividing the samples of each class into 10 bins. NBC models were systematically trained on 9 bins and evaluated on one bin (test bin). Then we computed a confusion matrix to evaluate the performance of the classifier. A single evaluation, corresponding to one test bin, yields one confusion matrix; hence, a 10-fold cross-validation procedure returns 10 matrices. An average matrix over these 10 matrices was then computed and returned.

The original imbalanced data was not able to provide enough "signal" to discriminate the classes well. The performance of the classifier built using the original data is summarized in Table 2.2. Out of 12 conformations from the *high* class only two conformations were correctly identified. On the other hand, 28 conformations were correctly classified and only 4 conformations were misclassified for the *low* class. Thus, the original data produced a highly specific classifier (small FPR) but not a sensitive one (small TPR). Since we were interested in finding members of the *high* class, a sensitive classifier, for this purpose, is more valuable than a specific one.

Table 2.2: Confusion matrix of the original data (left) and FPR, TPR rates (right).

	Predicted high	Predicted low	FPR	TPR
True high	2	9.8	0.127	0.169
True low	4.1	28.1		

2.3.3 Discriminating classes with over-sampling

In an attempt to assess the usefulness of a balanced data set, we performed an over-sampling procedure on the data using an ADASYN algorithm. The algorithm first computed a distribution of required synthetic samples for each member of the minority class. An appropriate number of synthetic samples were subsequently generated using the distribution. The β parameter in equation 2.2 was used to tune the amount of synthetic samples; setting β to 0.65 produced a balanced distribution between the two classes (Figure 2.2). As expected by way of constructions, the distributions of the classes remained overlapped indicated by the dark green shaded areas (Figure 2.2).

Interestingly, balancing the number of samples between the two classes improved the discrimination capability of the classifier, inspite of the overlapping distributions. When the data was balanced by generating synthetic samples for the minority class using the ADASYN algorithm, the performance of the resulting classifier trained on this data improved noticeably.

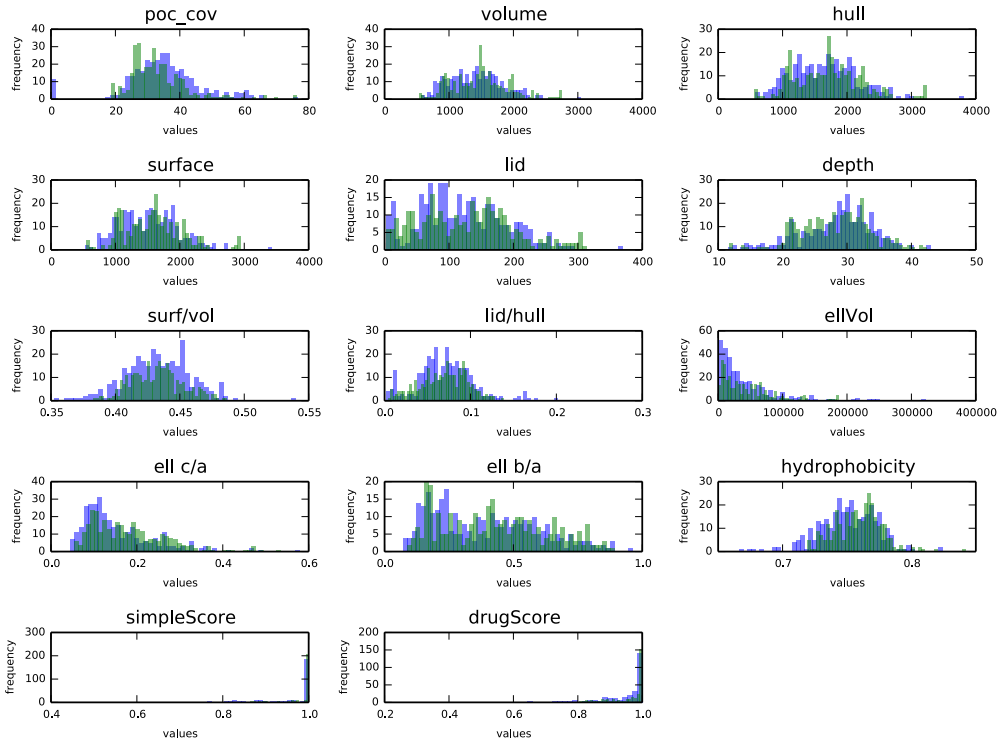


Figure 2.2: Distributions of balanced continuous descriptors. Purple and green represent samples from *high* and *low* classes, respectively. Dark-green shaded areas indicate the overlapping samples from the two classes.

The observed FPR and TPR values for this classifier were 0.217 and 0.413, respectively. While FPR climbed from 0.127 to 0.217 (170% increase); a larger gain was observed for TPR, from 0.169 to 0.413 (240% increase). In other words, the classifier trained on balanced data was more sensitive (Table 2.3). More importantly, the gain in sensitivity came at a lower reduction of specificity.

2.3.4 Scanning the over-sampling space

Having observed that balancing the data by over-sampling the minority class had a positive impact on the classifier sensitivity, we asked how far the balance between the classes can be tilted while remaining beneficial (in terms of sensitivity and specificity)? To answer this question, we generated a number of NBC models (classifiers) for varying β values by scanning values ranging from 0 to 50 in increments of 0.2. As each β is associated with one classifier,

Table 2.3: Confusion matrix of the balanced data (left) and FPR, TPR rates (right).

	Predicted high	Predicted low	FPR	TPR
True high	13.3	18.9	0.217	0.413
True low	7	25.2		

we ended up with 250 classifiers. TPR values for all the classifiers were then computed and plotted as a function of FPR; such a curve is known as a receiver operating characteristic (ROC) curve. We evaluated three different measures to compute a probability value for a sample: σ and $\sigma/2$ as the upper and lower boundaries, and a canonical cumulative probability from a CDF. The original models summarized in Tables 2.2 and 2.3 used σ as the upper and lower boundaries. In doing so, we evaluated a total of 750 NBC models (ROC curves shown in Figure 2.3).

Over-sampling the minority class improved the sensitivity of the classifiers for all 750 models (Figure 2.3). The balanced model (Table 2.3 and Figure 2.3) illustrates a gain in TPR with a steep increase on the *y-axis* for this model. Improvements in TPR can still be obtained by additional over-sampling of the minority class, however, this came at the expense of larger reductions in specificity as indicated by larger shifts along the *x-axis* beyond the balanced model. The figure also suggests that slightly better models were obtained when probabilities were computed using σ or $\sigma/2$ (models ranging from 0.2 to 0.5 on the *x-axis*) when compared to models computed using plain CDF.

2.3.5 Prioritizing predicted conformations

Due to the trade-off between sensitivity and specificity, misclassifications (wrongly labeled conformations) must exist in any classifier barring a perfect one. In order to prioritize the true positives over the false positives, we used P_{ratio} values since they reflect an inverse relationship between the classes in a sample. A large P_{ratio} suggests that the probability of finding a sample in the *high* distribution is much larger than the inverse scenario. In addition to this metric, we also considered the density of the *high* distribution by weighting P_{ratio} values with joint probabilites of observing samples in the *high* distribution, this metric is termed WP_{ratio}.

WP_{ratio} ranked the conformations better than P_{ratio}. To assess the efficacy of the ranking metrics, we trained two models for each metric with two different βs reflecting two conditions: mildly over-sampled where β was set

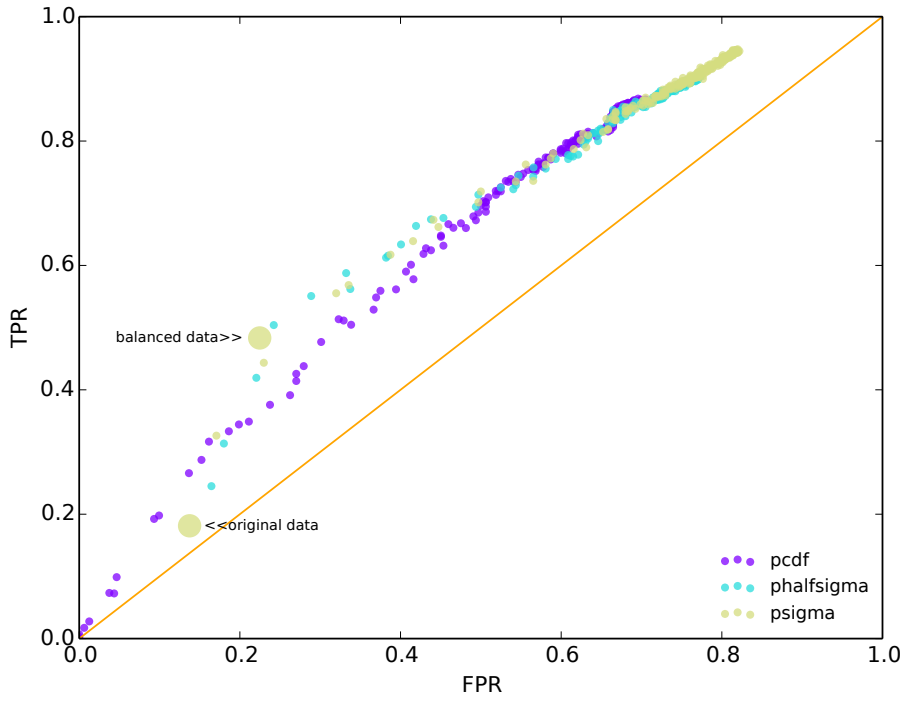


Figure 2.3: ROC plots of the models. Purple, cyan and green are models computed with σ, $\sigma/2$ and plain CDF, respectively. The orange line represent a random classifier.

to 0.65 (balanced data) and a heavily over-sampled counterpart where β was set to 6.5. The top ten conformations ranked by each metric were returned. Subsequently the correctly labeled conformations were identified and the percentage of correctly labeled conformations from this set was computed. Due to the presence of stochastic elements in the ADASYN algorithm, a single test would not be sufficient. Hence we iterated the process 100 times to obtain a distribution for each case. This enabled us to assess the metrics more objectively. When β was set to 0.65 and WP_{ratio} was used as the ranker we were, on average, able to prioritize around 72% of true positives (Figure 2.4 shows the distributions of these percentage values). Increased over-sampling by setting β to 6.5 further improved the percentage to 73% (Figure 2.4, left panel). On the other hand, with P_{ratio} as the ranker we were only able to prioritize around 52% and 44% of the true positives with $\beta = 0.65$ and 6.5, respectively.

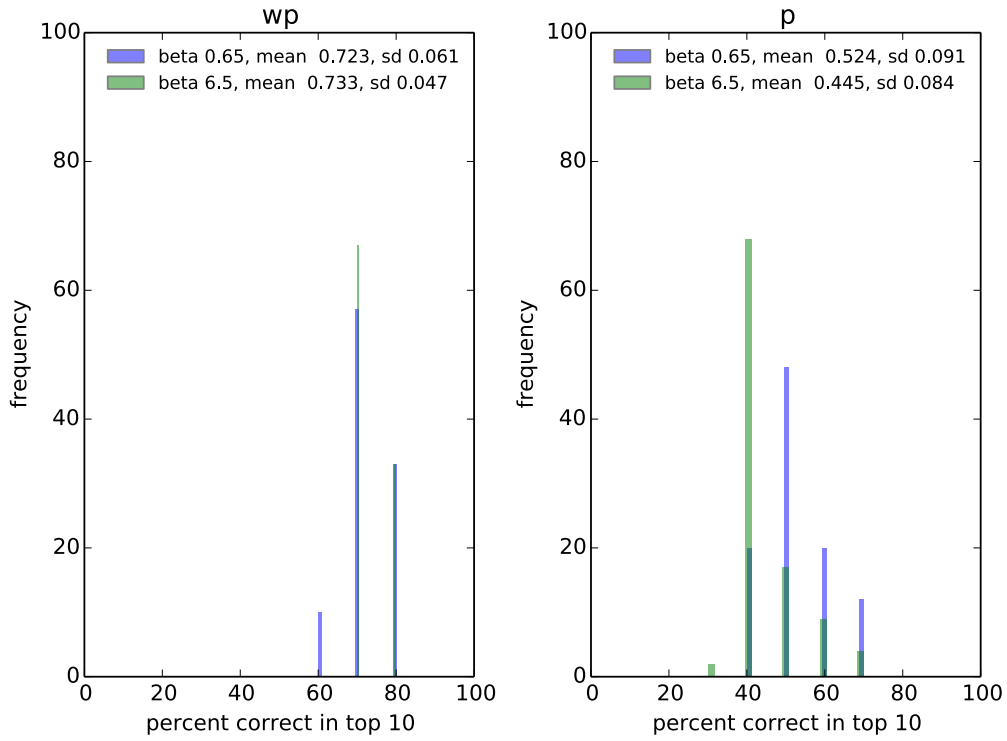


Figure 2.4: Distributions of percent correct in top 10. Left and right panels are WP_{ratio} and P_{ratio}, respectively.

2.3.6 Contributions of descriptors

To evaluate the contributions of the descriptors in a classifier, we performed a two sample Kolmogorov-Smirnov (KS) test on each descriptor. The test returned the largest distance (D) between classes in a given descriptor. A descriptor exhibiting a large distance between the *high* and *low* classes is assumed to contribute positively to the discriminating power of a classifier. Table 2.4 summarizes the distances and their corresponding p-values for the original and balanced data sets. Five descriptors exhibited more than 10% difference between *high* and *low* distributions in the original data set as well as in the balanced data set: *hydrophobicity, ell c/a, simpleScore, poc_cov* and *drugScore*.

The highest D values were designated to *hydrophobicity* in both data sets. Hydrophobicity has, traditionally, been used to characterize druggable pockets. For instance, Map_{pod} [85], SiteMap [86], DLID [87] and DrugScore [88] incorporate hydrophobicity when prediciting the druggability score of a pocket. In agreement with this, hydrophobicity exhibited the largest distance

Table 2.4: D statistics and p-values of *high* and *low* classes for each feature in the original data set (left panel) and balanced data set (right panel).

Feature	D	Pval
hydrophobicity	0.197	0.00223
ell c/a	0.19	0.0135
simpleScore	0.158	0.00941
poc_cov	0.148	0.0318
drugScore	0.138	0.00885
ell b/a	0.095	0.000807
lid	0.087	0.0623
volume	0.072	0.0141
hull	0.068	0.0146
surface	0.064	0.00396
depth	0.054	0.0787
ellVol	0.053	0.0585
surf/vol	0.051	0.0483
lid/hull	0.032	0.0281

Feature	D	Pval
hydrophobicity	0.205	0.0194
simpleScore	0.168	0.0544
ell c/a	0.161	0.0185
poc_cov	0.144	0.0326
drugScore	0.129	0.0665
ell b/a	0.087	0.00351
surf/vol	0.067	0.0104
lid	0.061	0.061
surface	0.059	0.0155
hull	0.056	0.00586
volume	0.055	0.00251
depth	0.049	0.0268
ellVol	0.045	0.159
lid/hull	0.044	0.0458

in the data sets used in this study affirming the central role of hydrophobic environments in protein-drug interactions.

To evaluate the discriminating capability of descriptors with high D values, data sets containing only these descriptors (HD and HDB) were compiled from the original and balanced data sets, respectively. HD and HDB data sets contain only *hydrophobicity, simpleScore, ell c/a, poc_cov* and *drugScore* in each sample. Similar to the previous section, the distributions of percent correct in top 10 predicted conformations were plotted after iterating the ranking procedures 100 times (Figure 2.5). HD and HDB data sets exhibited a substantially lower performance in comparison to the original and balanced data sets. When WP_{ratio} was used, HD and HDB data sets returned around five correct predictions averaging at 0.55 and 0.50, respectively. In contrast, classifiers trained on the full set of descriptors were able to return around seven correct predictions averaging at 0.72 and 0.73 (Figure 2.4). Similar reductions of performance were observed when P_{ratio} was used to rank the predictions. These observations highlight the faint nature of the discriminating signals in the data sets (weak signal to noise ratio). Considering only descriptors with high D values did not provide sufficient amount of information to discriminate the classes. Hence, the following section uses classifiers trained on the full set of descriptors exclusively.

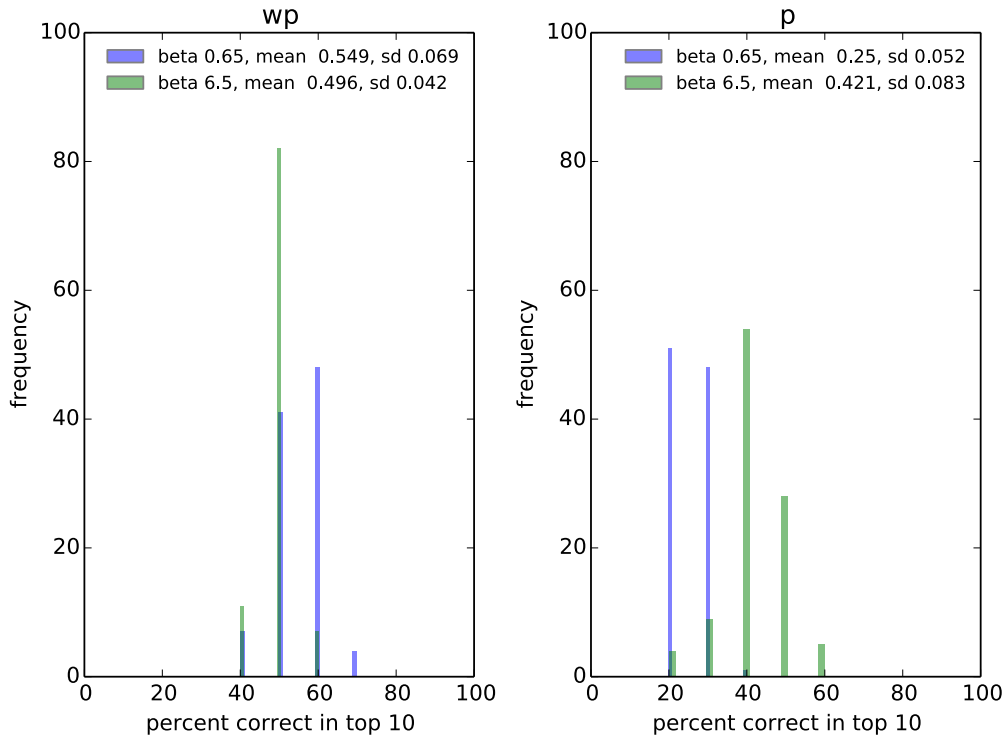


Figure 2.5: HD (purple) and HDB(green). Distributions of percent correct in top 10. Left and right panels are WP_{ratio} and P_{ratio}, respectively.

2.3.7 ENRI

ENRI is a Python [89] program that encapsulates all the aforementioned procedures. It offers a simple interface and access to these procedures. The program was written in Python version 2.7. The program currently uses training data derived from MD simulations of nuclear receptors. Hence at the moment it is only relevant for processing conformations from this nuclear receptor family. However, extending its usage to other proteins only requires the input of a new training data set to the program. ENRI, in short, works as follows: first, the program takes inputs in PDB format (from PDB itself or MD simulations); second, it computes pockets and the corresponding descriptors using DoGSiteScorer; third, it predicts/labels the inputs using an NBC, parameters for the NBC are estimated from the training data with minority over-sampling by ADASYN; fourth, it ranks conformations labeled as *high* using either WP_{ratio} or P_{ratio}; lastly, it writes the top ten ranked conformations to a text file. The program (source code) is available for download from GitHub at `https://github.com/fibonaccirabbits/enri`.

2.3.8 Use case: nuclear receptors

In assessing the usefulness of ENRI, we used a subset of seven nuclear receptors from Table 2.1 as a test case. Unrestrained MD simulations were performed on the nuclear receptors using GROMACS [90] so that the resulting pockets were generated independently from the training data set. We used a total of 2500 frames (conformations) extracted from the resulting MD trajectories of each nuclear receptor. Binding pockets and the corresponding descriptors from these conformations were computed and fed to ENRI for predictions. We then identified SBVS-enriching conformations in the top 10 predicted conformations for each nuclear receptor. We also compared ENRI's predictions with conformations from clustering-based selection procedures described in the work of Jusoh et.al. [manuscript in preparation]: POVME [91] (pocket shape), RMSD [92] (binding pocket atoms) and VOM [93] (pocket-volume).

Table 2.5: SBVS-enriching conformations selected by ENRI, POVME, RMSD and VOM. Count is the total number of SBVS-enriching conformations found (maximum is 10) and EF max is the largest EF1% value amongst the selected conformations.

PDB ID	EF X-ray	ENRI count	EF max	POVME count	EF max	RMSD count	EF max	VOM count	EF max
1SJ0	46	0	44	1	47	0	45	0	45
3BQD	20	1	20	0	16	0	18	1	20
3L3X	19	6	24	6	22	5	22	5	24
2P54	26	1	27	2	34	2	30	6	32
2A3I	15	7	28	1	21	2	17	1	22
3SP9	24	6	33	5	39	7	32	8	38
2GTK	11	4	15	6	20	7	17	5	17

ENRI performed well when compared to the clustering-based selection procedures. Table 2.5 summarizes the total number of SBVS-enriching conformations (count) and maximum EF1% values (EF max) for ENRI, POVME, RMSD, and VOM. ENRI was able to find SBVS-enriching conformations (ENRI count) in six out of seven nuclear receptors along with POVME and VOM. In terms of EF1%, ENRI was able to select conformations with the highest EF1% (EF max) in three out of seven nuclear receptors. On the other hand, POVME, RMSD and VOM were able to find four, zero, and two out of seven nuclear receptors, respectively. ENRI performed particularly well for mineralocorticoid receptor (PDB ID 2A3I). Out of 10 predicted conformations, seven of them were SBVS-enriching conformations; POVME, RMSD, VOM were only able to identify one, two, and one conformation(s), respectively. For 2A3I, ENRI was also able to select conformations with the highest EF1% improvement over the protein's reference X-ray structure av-

eraging 17.6 whereas the average of other methods were only in the range of 8.4 to 8.7 (Table S1, supplementary materials). The best predicted conformation derived from 2A3I yielded an EF1% value of 28, reaching almost 100% improvement over the reference X-ray structure (EF1% 15).

While ENRI's performance was comparable or better than the other methods in six of the seven nuclear receptors, the program notably underperformed for peroxisome proliferated activated receptor alpha (PDB ID 2P54). For this nuclear receptor, only VOM was able to select a substantial number of SBVS-enriching conformations. This suggests that the current set of selection procedures, at least the ones discussed in this work, are complementary to each other. ENRI, with its current training data, remains useful. However, the program was not able to outperform the other selection procedures exhaustively. This also suggests that proteins classified under the same family can produce disparities in terms of enrichment measures in an SBVS.

2.4 Concluding remarks

Selecting "good" or "optimal " SBVS targets is often a challenging exercise, particularly when a large number of target conformations are available (e.g., conformations generated through computational techniques such as MD simulations). Here, we presented a new method to extract optimal target structures for SBVS experiments. We found that distinct features discriminating between a good or a bad structure often are only faintly represented. In other words, the descriptors themselves are ambiguous. Nevertheless, class-discerning signals can be amplified in ambiguous data by a minority oversampling procedure. Amplified data from nuclear receptors, when utilized to train a binary classifier, allowed us to identify SBVS-enriching conformations with an EF1% value reaching two-fold that of the reference X-ray structure. Ultimately, a trade-off between sensitivity and specificity exists and is an intrinsic characteristic of any classifier including the one described in this work. Even so, the trade-off can be circumvented by feeding *better* data to the classifier i.e. the performance of the classifier can be further improved by providing improvements in the data itself, both in terms of quality and quantity. Finally, we developed a Python program (workflow) that encapsulates all the procedures described in this work. The program is freely available at `https://github.com/fibonaccirabbits/enri`.

Chapter 3

Finding Allosteric Targets

This work was published in 2017 under the title "ALLO: a tool to discriminate and prioritize allosteric pockets" (*Akbar, R. and Helms, V. ALLO: a tool to discriminate and prioritize allosteric pockets. Chem Biol Drug Des. Accepted Author Manuscript. doi:10.1111/cbdd.13161*). The developed programs and dataset were made available to public via github at `https://github.com/fibonaccirabbits/allo`. Volkhard Helms supervised the work. I developed the approach, implemented the machine learning model, and wrote the manuscript.

3.1 Introduction

Binding of ligands to allosteric sites of proteins may, by definition, either activate or de-activate the corresponding active (orthosteric) sites [94]. By mapping proteins with known allosteric sites [95] to gene ontology (GO) annotations [96], one notices that allostery is found in a wide range of biological processes (Figure S1). The three most frequent GO annotations (from Figure S1) are nucleotide binding, metal ion binding, and membrane. These terms relate to important protein families such as tyrosine kinases, ion channels, and G-protein coupled receptor (GPCRs). These families represent 5%, 17%, and 19% of the current human drug targets, respectively [97]. In other words, a substantial portion of all human drug targets are allosteric proteins. In addition to its significance as drug targets, allostery has also been named *the second secret of life* [98].

Protein surfaces are not smooth. They are decorated by an array of knobs (38%) and clefts (62%) [99]. Concave areas created by these structural elements are termed pockets. Residues constituting such pockets dictate the properties of the pockets. For instance, residues found in knobs were shown

to be more charged, less hydrophobic and less aromatic than those found in clefts [99]. Such differing compositions may create highly specialized microenvironments that favour interactions with certain classes of molecules such as drugs or ligands [100]. Protein surfaces can be dynamic as well. Using molecular dynamics (MD) simulations, Eyrisch and Helms [68] demonstrated for several protein systems (BCL-XL, IL-2, and MDM2) that transient pockets can open multiple times on a time scale of 100 picosecond. Furthermore, using molecular docking, they sucessfully placed inhibitor molecules in these transient pockets.

As allostery couples allosteric sites to orthosteric sites, allosteric pockets afford regulation at a distance. That is, modulations of protein functions are not only possible through targeting their canonical active sites but also can be achieved by exerting pharmaceutically desirable conformational changes from their allosteric sites. For instance, the drug Cinacalcet, a calcimimetics molecule, interacts with a calcium receptor (a GPCR) of thyroid cells at its allosteric site and causes increased sensitivity to calcium ions [101].

A prerequisite to allosteric modulation is the identification and characterization of allosteric pockets. Databases such as the allosteric database (ASD) [95] and ASbench [102] provide compilations of known allosteric sites, orthosteric sites, and allosteric protein-ligand complexes to the scientific community. Machine learning algorithms such as Support Vector Machine (SVM) [103], Naive Bayes Classifier (NBC) [75], and Artificial Neural Network (ANN) [104] can be trained on these datasets to recognise biologically and pharmaceutically interesting patterns or to discriminate allosteric sites from orthosteric sites. Tools such as AlloPred [105] and AlloSite [106] took advantage of data from ASD and ASbench to train machine learning models capable of discriminating allosteric pockets from other pockets. Besides, Panjkovich and Daura [107] used structural dynamics and evolutionary conservation to identify allosteric sites, whereas Su et al. [108] employed thermodynamic coupling between allosteric sites and orthosteric sites to achieve the same objective.

This work, similar to AlloPred and AlloSite, takes advantage of machine learning algorithms to recognise patterns and to train predictive models on allosteric-orthosteric and allosteric protein-ligand complexes datasets. Pockets identified in the datasets were characterized by a set of physicochemical descriptors. Using these descriptors, we trained NBC models that can discriminate allosteric pockets from orthosteric pockets and ANN models that can prioritize allosteric pockets in a set of pockets found on a protein surface. Such models might be useful for discovering potentially novel allosteric pockets on various pharmaceutically relevant proteins of interest. Datasets along with a Python program encapsulating the predictive models (termed

ALLO) are available at `github.com/fibonaccirabbits/allo`.

3.2 Methods

3.2.1 Datasets: AO and APLC datasets

The first dataset was obtained from ASD [95] and comprises allosteric-orthosteric sites from proteins such as phophatases, GPCRs, nuclear hormone receptors, transcription factors, channels, peptidases, and kinases. From this dataset, we considered only those sites that were formed by a single protein chain. Sites formed by two or more chains were filtered out in order to create a less complex and balanced dataset. From these sites, pockets along with their corresponding descriptors were identified and characterized using the tool DoGSiteScorer [79]. Hence, each pocket is accompanied by a set of 65 physicochemical descriptors such as hydrophobicity, residue counts, residue types, ligand coverage, pocket coverage, etc. The complete list of descriptors is described in the original publications [79, 83, 84]. Descriptors that contained only zero values were omitted. The final dataset contains 143 unique PDB IDs (Supplementary Table S2); with 145 and 121 allosteric and orthosteric pockets, respectively, and a total of 48 descriptors. This dataset is refered to as allosteric-orthosteric (AO) dataset, hereafter.

The second dataset was the *Core-Diversity set* obtained from ASbench [102]. It comprises structurally diverse allosteric protein-ligand complexes commonly found in human and bacteria. Similar to the AO dataset, we considered only protein-ligand complexes with a single chain and used DoGsiteScorer to identify and characterize pockets on the protein surfaces. After removing descriptors that contained only zero values, the final dataset contains 95 unique proteins (Supplementary Table S3); with 118 and 1757 allosteric and non-allosteric pockets, respectively, and a total of 48 descriptors. This dataset is termed allosteric protein-ligand complexes (APLC) dataset.

3.2.2 Machine learning models

Naive Bayes Classifier (NBC)

NBC is a simple binary classifier where the input data (i.e., physico-chemical descriptors) are assumed to be independent from each other. The classifier discriminates between classes (here, allosteric or orthosteric pockets in the AO dataset) by calculating the probability of observing a pocket as an allosteric pocket and the probability of observing the same pocket as an orthosteric pocket and subsequently compares the two quantities.

The former probability was computed by taking the joint probability of observing the pocket in each descriptor (Equation 3.1) [109]. The latter probability, $P(orth|x)$, was defined analogously. Predictions were made based on the probability ratio between the allosteric and orthosteric pockets (Equation 3.2) [109]. A pocket would be predicted as allosteric if the ratio was larger than a certain threshold and vice versa. Probabilities for each descriptor on each class were obtained from density functions that were non-parametrically learned on the AO dataset by using kernel density estimators in SciPy [110].

$$P(allo|x) = \prod_{i=1}^{m} P(x_i|allo) \tag{3.1}$$

$$P_{ratio} = \frac{P(allo|x)}{P(ortho|x)} \tag{3.2}$$

Here, x is the descriptor vector of a pocket and m is the total number of descriptors.

Artificial Neural Networks (ANN)

ANN, much like NBC, can function as a classifier. Additionally, the model can also solve regression problems. The simplest ANN comprises only one layer of input and output nodes with a single weight matrix connecting them. More elaborate network topologies can contain any number of layers between the input and output nodes. Classification ANNs as well as regression in combination with classification ANNs were used to learn on the APLC dataset. A sigmoid function was used as the activation function (Equation 3.3) [111] and weights were learned using the stochastic gradient descent method to minimize an objective function (sum squared error, SSE, Equation 3.4) using a canonical update rule (Equation 3.5) [111].

$$s(z) = \frac{1}{1 + e^{-z}} \tag{3.3}$$

$$SSE = \sum_{i}^{n} (\hat{y}_i - y_i)^2 \tag{3.4}$$

$$W = W_{prev} - \lambda \nabla_W SSE \tag{3.5}$$

Here, n, $\hat{y}$, W_{prev}, λ, and $\nabla_W SSE$ are the total number of pockets, predicted outcomes, a weight matrix from the previous iteration, a learning rate, and the gradient of SSE with respect to weight matrix W, respectively. To reduce the parameter search space, we set the number of nodes in a hidden

layer, λ, and the number of iterations in gradient descent to 10, 3, and 30, respectively. The considered ANN topologies were implemented in Python.

3.2.3 Quality control

Accuracy (Equation 3.6), true positive rate (TPR, Equation 3.7), and false positive rate (FPR, Equation 3.8) [111] were used to assess the quality of the models.

$$accuracy = \frac{TP + FP}{P + N} \tag{3.6}$$

$$TPR = \frac{TP}{TP + FN} \tag{3.7}$$

$$FPR = \frac{FP}{FP + TN} \tag{3.8}$$

True positive (TP) is the number of pockets that are allosteric and are predicted as allosteric. False positive (FP) is the number of pocket that are non-allosteric but predicted as allosteric. True negative is the number of pockets that are non-allosteric and are predicted as non-allosteric. Lastly false negative (FN) is the number of pockets that are allosteric but predicted as non-allosteric.

3.2.4 ALLO

ALLO is a Python program for discriminating and/or ranking allosteric pockets against orthosteric pockets and non-allosteric pockets. As it turned out (see Results) that the NBC models can satisfactorily discriminate between allosteric and orthosteric pockets and, in addition, the ANN models can prioritize allosteric pockets over non-allosteric pockets, we implemented the most optimal models based on the two algorithms in a Python program and added a few simple helper scripts to interface with the program. ALLO, in brief, takes an output file from DoGSiteScorer (containing pockets and the corresponding desriptors) as its input. Users can then choose to either label (predict) the pockets as allosteric or orthosteric using an NBC model or rank the pockets using an ANN model. Finally, an output file containing the predictions or ranked pockets is written to the local directory. Datasets along with the (source code) program are freely accessible at `github.com/fibonaccirabbits/allo`.

3.3 Results

3.3.1 Discriminating allosteric pockets from orthosteric pockets in the AO dataset

Residue counts

To compare the approximate size of allosteric and orthosteric pockets, we counted the number of residues that constitute a pocket (pocket lining residues) and plotted the counts along with the corresponding frequencies (Figure 3.1). Allosteric and orthosteric pockets had 5 to 37 and 2 to 38 residues, respectively. On average, allosteric pockets comprised 18.8 residues whereas orthosteric pockets comprised 19.2 residues. The difference between the two distributions was, however, not statistically significant (P-value 0.61, t-test).

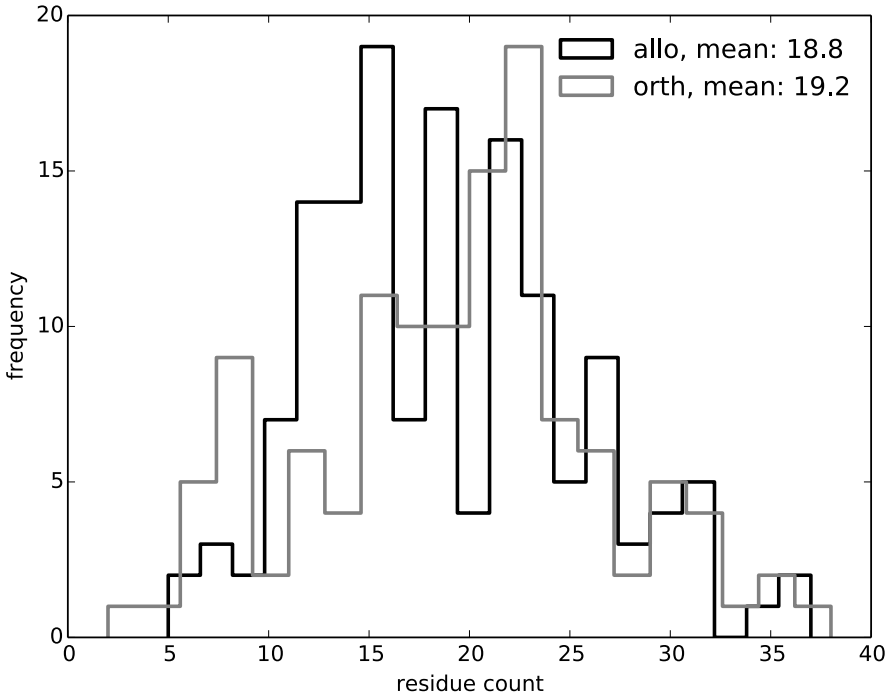


Figure 3.1: Frequency distribution of the number of residues in allosteric (black) and orthosteric pockets in the AO dataset.

Pocket descriptors

Next, we analyzed the distribution of descriptors for allosteric and orthosteric pockets using cumulative density plots (Figure S2). Despite most descriptors had similar distributions, some descriptors such as ligand coverage (*lig_cov*), pocket coverage (*poc_cov*), and *hydrophocity* showed distinctive differences between the classes.

To get a quantitative measure for the signal strength in a descriptor, we calculated the maximum difference between the cumulative densities of allosteric and orthosteric pockets. This procedure corresponds to the Kolmogorov-Smirnov (KS) test. Only 22 descriptors (out of 48) returned false discovery rate (FDR) adjusted P-values below 0.05 (see Table 3.1). Indeed, *poc_cov* (0.811), *lig_cov* (0.551), and *hydrophobicity* (0.363) were the three descriptors that held the strongest discriminating signals. We also noticed that the druggability score (*drugScore*) was ranked 8 in Table 3.1. Apparently allosteric pockets were predicted as more druggable than orthosteric pockets by DoGSiteScorer.

3.3.2 Classifying pockets with NBC

Then, we assessed the efficacy of NBC models for classifying pockets into allosteric and orthosteric pockets by randomly partitioning the AO dataset into three equal parts. An NBC model was trained on two parts of the dataset and evaluated on the remaining part. Since the AO dataset was rather balanced (allosteric to orthosteric pockets ratio is 1.2:1), we used *accuracy* as the quality metric to evaluate the models. The partitioning, training, testing and evaluation procedures were repeated 100 times to account for variability during the partitioning steps. We also examined the model performance with respect to signals in descriptors by training models on subsets of all descriptors (top three, top ten, and all descriptors, according to the ranking in Table 3.1).

NBC models trained using the top three descriptors, top ten descriptors, and all descriptors correctly classified 90%, 90%, and 87% of the test dataset, respectively (Figure 3.2). Despite the similarly high accuracy across these models, the ranges where these models yielded good *accuracies* differed. The simplest models (trained on only three descriptors) were most robust since they operated over the widest P_{ratio} threshold range (see Figure 3.2). On the other hand, the most complex models (trained using all descriptors) yielded the tightest threshold range (narrowest plot on Figure 3.2). Models trained on ten descriptors (top10) were in between these extremes. Maximum accuracies (max_acc) were obtained at around the canonical P_{ratio} threshold

Table 3.1: Maximum distances (D) between the cumulative densities of descriptors in allosteric and orthosteric pockets sorted from high to low. P-values were corrected for false discovery rate (FDR) with alpha 0.05.

D	P-value	Descriptor
0.811	< 10e-3	poc_cov
0.551	< 10e-3	lig_cov
0.363	< 10e-3	hydrophobicity
0.323	< 10e-3	GLU
0.294	< 10e-3	PHE
0.26	< 10e-3	ILE
0.258	< 10e-3	ell c/a
0.254	< 10e-3	drugScore
0.242	0.005	LEU
0.238	0.005	aromat
0.228	0.009	negAA
0.221	0.011	ALA
0.219	0.011	lid/hull
0.214	0.014	GLY
0.209	0.016	polarAA
0.207	0.018	simpleScore
0.205	0.019	accept
0.205	0.019	Os
0.196	0.028	apolarAA
0.19	0.036	surf/vol
0.188	0.037	depth
0.187	0.037	ellVol

value of one.

Descriptors pertaining ligand and pocket information (*poc_cov* and *lig_cov*) bear large proportions of discriminating signals and were the highest ranked descriptors in the dataset. These two descriptors incorporate information from both pocket residues and ligands whereas the other descriptors are computed solely based on the residues that constitute a pocket. One should notice, though, that information on the bound ligand is typically not available in a drug design effort. Thus, we tested the efficacy of the NBC models after excluding the two descriptors from the dataset. As in earlier sections, we trained the models on subsets of three, ten, and all descriptors. Upon eliminating these pocket-ligand descriptors, we noticed a clear decrease of the *accuracy* in all models. Further, we observed an inversed scenario whereby the simplest models (top3) yielded inferior performance than the more complex models. The maximum *accuracy* of the best performing model (all descriptors) was now around 75% (Figure 3.3). It appears that in the ab-

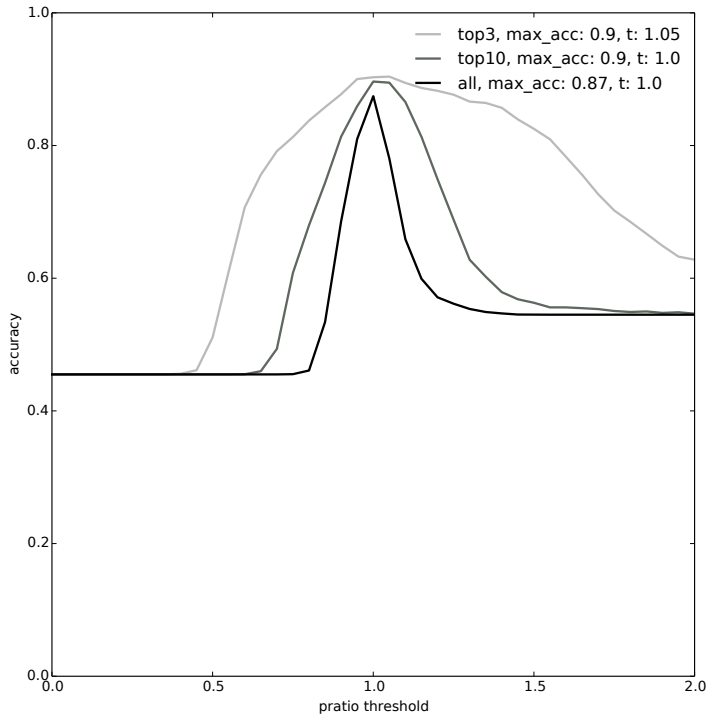


Figure 3.2: Accuracies of NBC models trained on top three descriptors (light grey), top ten descriptors (medium grey), and all descriptors (black). Maximum accuracies (max_acc) and the corresponding P_{ratio} thresholds (t) are given in the plot's legend.

sence of strong discriminating signals, aggregating weaker signals from all descriptors (building more complex models) partially offsets the loss and is a reasonable strategy.

3.3.3 Prioritizing allosteric pockets in a set of pockets on APLC dataset

Unlike the AO dataset (which contains only allosteric sites and orthosteric sites), the APLC dataset includes any pocket from each protein-ligand complex in the dataset. Only a small fraction of the total pockets (1875) were allosteric pockets (118); the ratio of allosteric to non-allosteric pockets was 1:15. This substantial imbalance in the dataset altered the (machine) learning objective. Instead of using machine learning to solve a binary classification

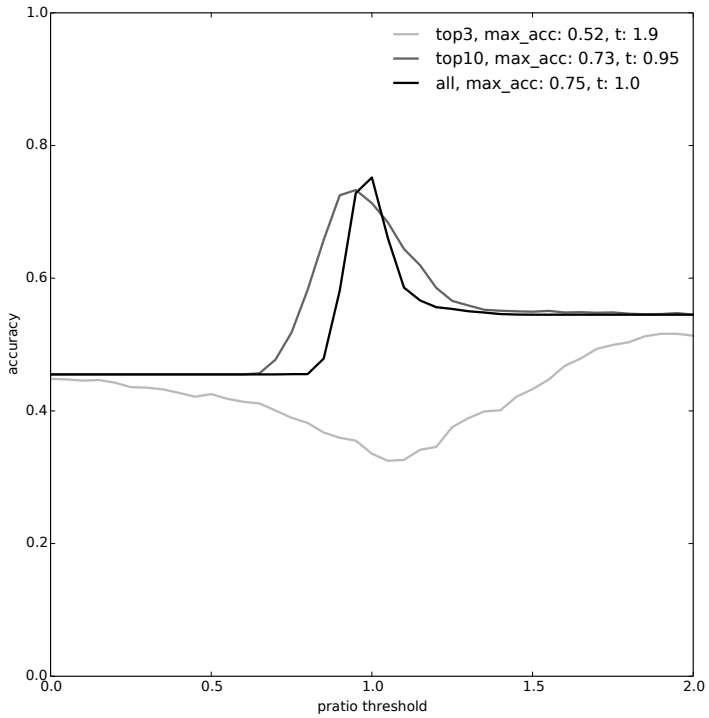


Figure 3.3: Accuracies of NBC models trained on the top three descriptors (light grey), top ten descriptors (medium grey), and all descriptors (black) after eliminating pocket-ligand descriptors. Maximum accuracies (max_acc) and the corresponding P_{ratio} thresholds (t) are given in the plot's legend.

problem (as in discriminating allosteric from orthosteric), we now used it to solve a ranking problem. Precisely, the goal was to rank a set of pockets found in a protein such that pockets at the top of a ranked list were enriched with allosteric pockets.

All ligands in the APLC dataset are allosteric ligands, hence, non-allosteric pockets in the dataset are empty pockets. Thus, we needed to exclude pocket-ligand features (*poc_cov* and *lig_cov*) when training a machine learning algorithm. Since aggregating weaker signals in the absence of strong signals proved to be a sound strategy (as seen in previous sections), ANN models in the following sections use the full set of descriptors.

Pocket residues

First, we analyzed again the size of the pockets. As the dataset was heavily imbalanced we visualized the residue count distribution as cumulative densities (Figure 3.4). Allosteric pockets, on average, contained 44.8 residues while non-allosteric pockets contained 27.8 residues.

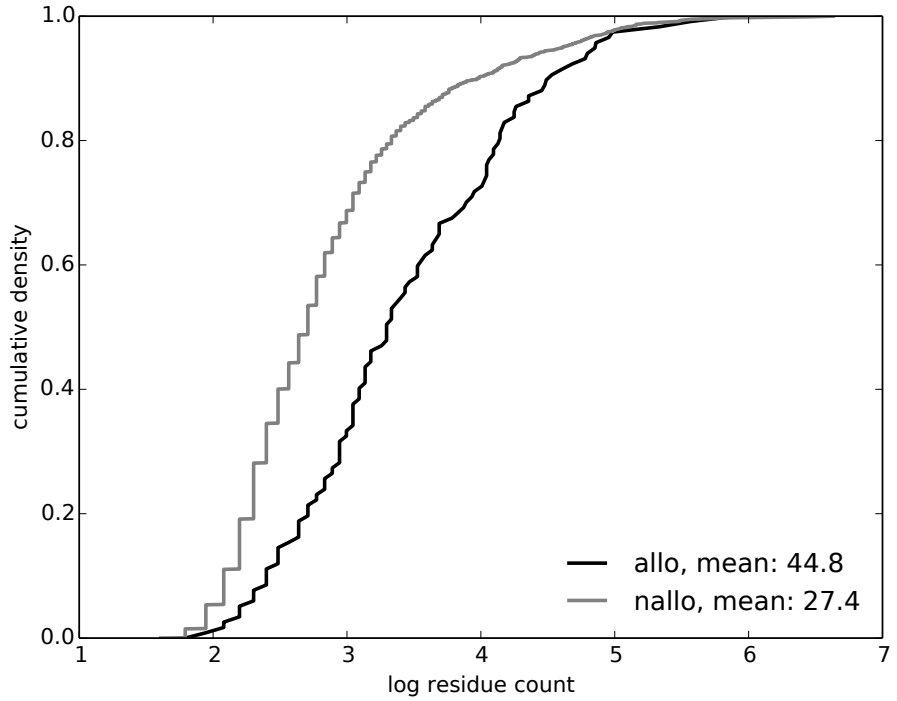


Figure 3.4: Cumulative density plots of residue counts in allosteric pockets (allo) and non-allosteric pockets (nallo) in the APLC dataset.

Pocket descriptors

Figure S3 illustrates the cumulative distribution of the decriptors and Table 3.2 lists the descriptors with FDR adjusted P-values below 0.05. For most descriptors, allosteric pockets have larger values than non-allsoteric pockets. Similar to the AO dataset, allosteric pockets were predicted as more druggable than non-allosteric pockets. On the other hand, *hydrophobicity* showed no discernable difference between allosteric and non-allosteric distributions. Larger distances (indicated by larger maximum distance D in Table

3.2) compared to distances found in the AO dataset (Table 3.1) were also recorded.

Prioritizing allosteric pockets using NBC

Due to the large imbalance between classes in the dataset, we evaluated the classes separately by plotting true positive rate (TPR) as a function of false positive rate (FPR) and computing the area under the curve (AUC) for the resulting plot. Such a plot is known as the reciever operating characteristic (ROC) plot. As the ROC plots were computed for all pockets in the dataset without accounting for which set of pockets belong to which protein, these quality metrics signify the global performance of the models. All models, regardless of their descriptors sets, returned similar plots and AUCs (Figure 3.5).

Finally, we used the P_{ratio} values to rank pockets in each protein and calculated the percentage of finding allosteric pockets in the top three of the ranked pockets on all proteins in the test dataset (*percent in top3*). This measure served as an indicator for local performance as it takes into account which set of pockets belong to which protein. Despite the decent AUC values, on average, our NBC models were only able to prioritize allosteric pockets over the non-allosteric ones for 30% of the total proteins in our test dataset (*percent in top3*, Figure 3.5). In other words, while the global performance of these models was satifactory, their local performance remained insufficient.

Prioritizing allosteric pockets using ANN

Next, we tested a more elaborate model in the form of ANN. Contrary to NBC models that require a minimal quantity of parameters (here, we needed to optimize only the thresholds of P_{ratio}), ANN models require at least two topology parameters (number of nodes in hidden layers and number of hidden layers) and two gradient descent parameters (a learning rate, and number of iterations). We reduced the parameter search space by only optimizing the number of hidden layers (size) of the ANN models. The other parameters were set to fixed values (see Methods). Similar to earlier sections, we randomly partitioned the dataset into three equal parts, trained ANN models on two parts, tested the models on one part and iterated the procedures 100 times to account for variabilities during the partitioning steps.

We examined two strategies to build ANN models. The first strategy used a "vanilla" setup wherein the APLC dataset (containing 46 descriptors) was passed directly to a classification ANN. The second strategy used two ANNs, the APLC dataset (of 46 descriptors) was first passed to a regression

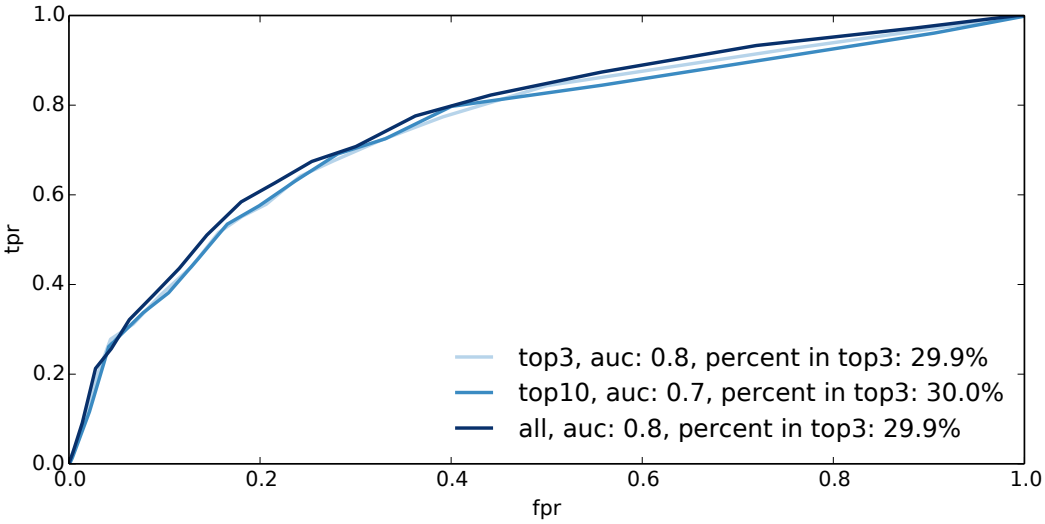


Figure 3.5: ROC plots of NBC models trained on top three descriptors (light blue), top ten descriptors (medium blue), and all descriptors (dark blue); area under the curve (AUC). The maximum percentage of finding allosteric pockets in the top three of the ranked pockets on all proteins in the test dataset (*percent in top3*) are given in the plot's legend.

ANN to predict the pocket-ligand descriptors *poc_cov* and *lig_cov*. Recall that these descriptors had to be removed because they are absent in non-allosteric pockets. The predictions of the first stage were then added to the original dataset resulting in a new dataset with 48 descriptors. Finally the new dataset was then passed to a classification ANN.

Both strategies prioritized allosteric pockets over non-allosteric pockets satisfactorily well. ANN models built using the vanilla strategy yielded *percent in top3* ranging from 29.0% to 80.6% with AUCs ranging from 0.5 to 0.8 (Figure 3.6). The best model in the vanilla strategy was an ANN model comprised of one input layer, five hidden layers, and an output layer (size 7). Models flanking this model yielded lower *percent in top3* particularly the most complex model (size 11, AUC 0.5 and, *percent in top3* 29.0%). Models built using the second strategy, regression-classification ANN, yielded *percent in top3* ranging from 22.6% to 83.9% with AUCs ranging from 0.4 to 0.8 (Figure S4). The best model on this strategy was an ANN of one input layer, seven hidden layers, and an output layer (size 9). In contrast to the vanilla strategy, we noticed a more erratic distribution of *percent in top3* values in regression-classification ANN models. Even though the global performance measures, indicated by AUC values between NBC models and ANN models, were comparable (maximum AUC in both NBC and ANN are 0.8), ANN

models performed much better locally (indicated by the higher *percent in top3* values found on ANN models). The most optimal ANN models were able to prioritize 80.6% (vanilla ANN) and 83.9% (regression-classification ANN) of the test dataset, respectively. This was a clear improvement over the NBC results where only around 30% of the test dataset could be correctly prioritized. Thus, the best model in either ANN strategy produced an almost three fold improvement over the best NBC models described in earlier sections.

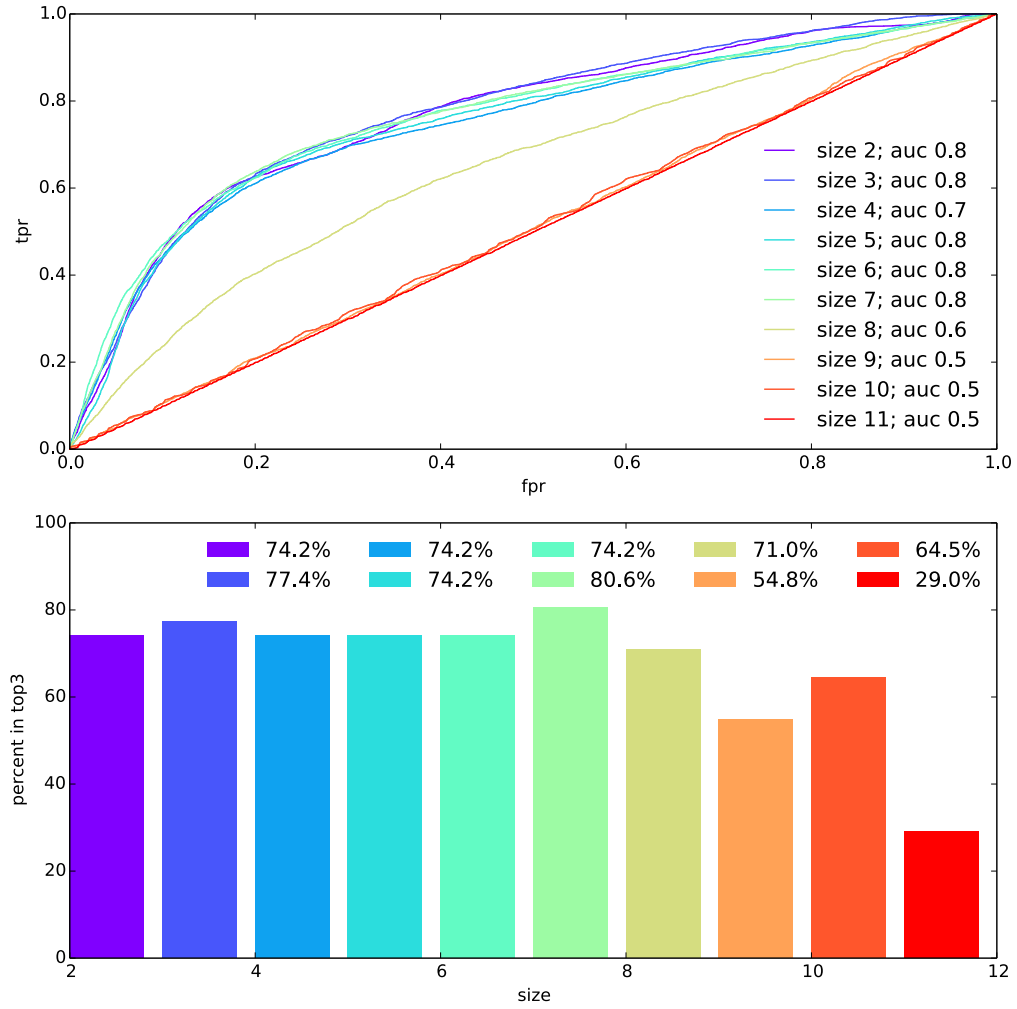


Figure 3.6: Vanilla ANN: ROC plots (top panel) and the percentages of finding allosteric pockets among the top three of the ranked pockets on all proteins in the test dataset (bottom panel).

3.4 Discussion

Descriptors containing ligand information turned out to contain the largest amount of information for distinguishing allosteric and orthosteric pockets. Both allosteric pockets (18.8) and orthosteric pockets (19.2) found in the AO dataset contained a similar quantity of residues (Figure 3.1). In addition, other geometric descriptors such as *surface, volume,* and *hull* had similar distributions as well (Figure S2). However, the coverage of a ligand in allosteric pockets (*lig_cov*) is typically much smaller than for the orthosteric pockets. Similarly, the coverage of a pocket (poc_cov) is much smaller in allosteric pockets than in the orthosteric ones. This indicates that allosteric ligands are more exposed to the hydrophilic environments compared to orthosteric ligands and that allosteric ligands interact less tightly with the pocket compared to orthosteric ligands. Furthermore, even though allosteric and orthosteric pockets share similar geometric templates, their residue (and thus their chemical) compositions differ.

When comparing allosteric pockets against other non allosteric pockets found in the APLC dataset the amount of residues constituting allosteric pockets was larger, and geometric descriptors such as *surface, volume* and *hull* showed clear differences between the classes (Figure S3 and Table 3.2). In addition, allosteric pockets, in both datasets, were predicted as more druggable than orthosteric pockets or non-allosteric pockets. This reiterates the significance of allosteric pockets as noteworthy pharmaceutical targets [112].

ALLO complements the current set of allosteric computational tools. AlloPred and AlloSite are two other tools that take advantage of the data from ASD and ASbench to build predictive models. AlloPred uses Fpocket [113] to identify pockets and generate the corresponding pocket descriptors. In addition to descriptors from Fpocket, it also integrates descriptors from a normal mode analysis (NMA) [114]. This set of descriptors is then used to train an optimal SVM model. Similarly, AlloSite uses Fpocket to identify pockets and to generate descriptors and subsequently trains an optimal SVM model. ALLO, on the other hand, uses DoGSiteScorer to identify pockets and to generate descriptors and then trains NBC and ANN models. Despite differences in the way these programs generate data, recognize patterns in the data, and evaluate the corresponding optimal models, their results are remarkably similar. AlloSite reported an accuracy of 95% when discriminating allosteric sites from other sites and our optimal NBC models yielded an accuracy of 94% when used to discriminate allosteric pockets from non-allosteric pockets. AlloPred reported a 70% success rate in its top two predictions and our optimal ANN models yielded a 67.7% and 83.9% success rate for its top two and three predictions, respectively. We also complemented AlloSite and

AlloPred by emphasising on how the usage of different sets of descriptors influence the performance of our models as this aspect was only briefly touched in these previous works.

One other noteworthy point is the inclusion of receptor flexibility in a drug discovery study. To achieve this objective computational tools such as molecular docking and MD simulation are often combined. For instance, a study on kinetoplastid RNA editing ligase 1 (KREL1) used an MD simulation to generate an ensemble of conformations, re-docked, and finally re-ranked the ligand using these conformations [115]. Such a procedure outputs a binding spectrum and is known as the relaxed complex scheme (RCS). Similarly here, albeit comming from the opposite direction, the ANN model could be used to recover the missing ligand information from an ensemble of apo conformations of an MD simulation. This information can then be used to select or re-rank a set conformations for further docking studies. Thus ALLO may also be used to select or prioritize a set of conformations from MD simulations (similar to the tool ENRI [116]) in addition to other usage discussed in earlier sections.

3.5 Concluding remarks

While allostery might not be the panacea for drug discovery, it could be an area where new pharmaceutics for remote functional modulations are discovered. Prerequisite to this, however, is the identification and the characterization of suitable allosteric pockets. This work describes, compares, and contrasts desriptors that constitute allosteric pockets, orthosteric pockets and other non-allosteric pockets. We found that these desriptors are useful for discriminating between allosteric pockets from other pockets and that some descriptors–particularly descriptors that take into account both ligand and pocket residue information– appear to contain more discriminating signals than other descriptors. In the absence of such strong discriminating signals, we noticed that aggregating signals from weaker descriptors can be a good strategy. Using these descriptors we trained NBC models capable of discriminating allosteric pockets from orthosteric pockets with satisfactory accuracies. In addition we also trained ANN models capable of prioritizing allosteric pockets over non-allosteric pockets on a set of pockets found in a protein surface. This model sucessfully identified allosteric pockets in the top two-ranked pockets for around 68% of the considered proteins and among the three top-ranked pockets for around 84%. Finally we assembled these models and provide them to the scientific community in the form of a Python program. Such predictive models might be relevant for

finding new allosteric pockets and potentially novel allosteric drugs targets. Datasets along with the program (source code) are freely accessible from `github.com/fibonaccirabbits/allo`.

Table 3.2: Maximum distances (D) between descriptors for allosteric and non-allosteric pockets sorted from high to low. P-values were corrected for false discover rate (FDR) with alpha 0.05.

D	P-value	Descriptor
0.476	< 10e-3	hull
0.471	< 10e-3	volume
0.452	< 10e-3	surface
0.427	< 10e-3	ellVol
0.421	< 10e-3	siteAtms
0.417	< 10e-3	Cs
0.402	< 10e-3	donor
0.4	< 10e-3	simpleScore
0.393	< 10e-3	aromat
0.382	< 10e-3	Os
0.381	< 10e-3	accept
0.38	< 10e-3	Ns
0.379	< 10e-3	sumAA
0.363	< 10e-3	depth
0.343	< 10e-3	polarAA
0.334	< 10e-3	apolarAA
0.322	< 10e-3	lid
0.317	< 10e-3	SER
0.27	< 10e-3	drugScore
0.262	< 10e-3	posAA
0.261	< 10e-3	ILE
0.256	< 10e-3	TRP
0.255	< 10e-3	TYR
0.249	< 10e-3	GLY
0.247	< 10e-3	VAL
0.234	< 10e-3	lid/hull
0.231	< 10e-3	PHE
0.231	< 10e-3	LEU
0.218	< 10e-3	THR
0.216	< 10e-3	surf/vol
0.211	< 10e-3	ALA
0.195	< 10e-3	ARG
0.188	0.001	PRO
0.181	0.001	negAA
0.181	0.001	LYS
0.148	0.018	HIS
0.147	0.019	ell c/a
0.146	0.019	Ss
0.146	0.019	MET
0.138	0.031	ASP
0.137	0.031	GLN

Chapter 4

Examining the Surface Dynamics of Calcium Exchangers

A manuscript based on this work is currently in preparation for submission to a peer reviewed journal. Volkhard Helms supervised the work. Daniel Khananshvili provided the domain expertise on sodium-calcium exchangers. I developed the apporach, implemented the machine learning model, and wrote the manuscript.

4.1 Introduction

Studies on sodium-calcium exchangers (NCX) date back more than a hundred years when Ringer, Daly and Clark [117, 118] showed that contractions of frog's cardiac muscles relate directly to the concentration of intracellular calcium. Furthermore, they also noticed that such contractions relate inversely to the concentration of extracellular sodium. Since then, these exchangers have also been shown to reduce cell tension in vascular smooth muscle cells [119], to involve in nerve terminal depolarization and transmitter release in brain and neuron cells [120, 121], to modulate intracellular signalling in astrocytes (the most abundant glial cells in the brain) [122], to maintain calcium ions homeostasis in kidney cells [123], and to be involved in the exchange of calcium ions in other cells such as skeletal, hepatocytes, osteoblasts and blood cells [124].

Structurally, these proteins contain two distinct regions; a channel that spans the cell membrane and a regulatory region situated in the cytosol. The channel contains ten alpha helices [125], whereas the regulatory region

contains two highly conserved beta sandwich domains of seven strands each [126]. The regulatory region, as the name suggests, contains the calcium binding sites and can be further categorized into two domains (Figure 4.1). The first calcium binding domain (CBD1) hosts the high affinity binding site and serves as the primary calcium sensor and the second calcium binding domain (CBD2) binds calcium with less affinity, a cytocolic loop connects the two domains [127]. In addition, CBD2 is expressed in a tissue-specific manner and experiences alternative splicing resulting in different responses to calcium ions in various tissues [128].

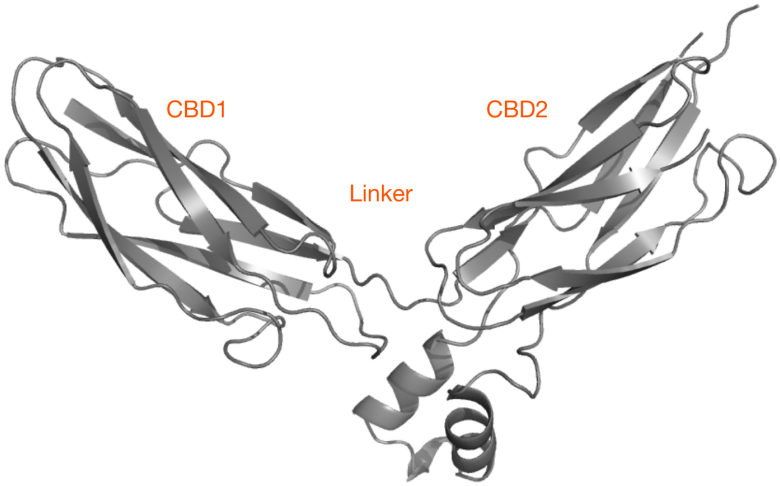


Figure 4.1: An illustration of regulatory domains of an NCX protein.

While their functions in various physiological processes are fairly well understood, the molecular mechanisms underlying the different phenotypic responses to calcium ions remain elusive. The interdomain angle between CBD1 and CBD2 was thought to be responsible for such differences [129], however, angles obtained from high resolution X-ray structures of two NCX splice variants associated with contrasting phenotypic responses were found to be nearly identical [130]. Despite the absence of large conformational changes (such as changes in the interdomain angle), more subtle changes have been observed. For instance, calcium binding to CBD1 has been observed to rigidify the main-chain of CBD2 and CBD2 is thought to contribute to the

stability of CBD1 in the absence of calcium ions [126]. Furthermore, structural analyses in combination with molecular dynamics (MD) simulations of NCX from *Methanococcus jannaschii* revealed the mechanisms for extracellular ion recognition and the outward-to-inward transition of the channel [125].

In this work, we examined dynamic events on the surface of CBD1 and CBD2 and tested whether and how such dynamics can be used to profile different phenotypic responses of NCX variants. We used three bound (holo) X-ray structures of CBD12 tandem associated with activation, no response, and inhibition as the phenotypic responses to calcium. We then generated a corresponding unbound (apo) state from each structure. MD simulations in combination with the pocket identification tool DoGSiteScorer were used to characterize pockets on the surface of the proteins during the 500 nanosecond (ns) long simulations. These data were then used to investigate the pocketwise diversity and density on the surface of the domains. We found meaningful differences in the pocket diversity of the apo and holo states in each splice variant in addition to an increase in the pocket density in CBD2 when calcium ions are present. Furthermore, residues with the highest density changes localized around the linker region and the tip of CBD2. Taken together, these dynamic conformational features may offer a novel perspective in examining and investigating the molecular mechanisms that underlie various phenotypic responses of these NCX variants.

4.2 Methods

4.2.1 Molecular dynamics simulation

Three crystal structures of CBD12 tandem, PDBID 3US9 (CBD12_1_4) [130], 3RB7 (CBD12_1_2) [129], and 3RB5 (CBD12_1_1) [129], associated with activation, no response and inhibition were retrieved from the protein data bank (PDB). In each case two states were generated for the simulations: a holo state wherein regulatory ions were kept (regulatory ions present) and an apo state where regulatory ions were removed (regulatory ions absent). The structures were then centered in a simulation box and solvated with the Tip3p [131] water model. The CHARMM27 [132] forcefield was used to describe the systems. A ten nanosecond (ns) simulation under NVT condition followed by another ten nanosecond simulation under NPT condition were performed to equilibrate the system. Production MD simulations were carried out for 500 ns. All simulations were carried out using GROMACS (version 4.5.5) [133].

4.2.2 Pockets, features, and clustering

Pockets on the surface of the NCX proteins were identified using the tool DoGSiteScorer [79]. Default parameters for the pocket detection algorithm, grid spacing, and contour level cutoff were used. A binary vector with length equal to the number of protein residues was generated for each pocket, a residue would have a value of one in the vector if the residues was present in a pocket and a value of zero otherwise. These vectors were merged to generate a matrix termed residue position matrix. Table 4.1 shows a snippet from such a residue position matrix. Hierarchical clustering was then performed on the matrices, the average linkage criterion was used to merge the clusters, and the Minkowski distance was used as the similarity metric. Clustering was carried out using SciPy [134]. In addition, we computed the inconsistency coefficient (Equation 4.1) for each clustering procedure and used this coefficient as a measure of diversity in the clusters.

$$IC_f = \frac{D_f - mean(D_{others})}{std(D_{others})} \tag{4.1}$$

Here, IC_f is the inconsistency coefficient of joint f and D_{others} are the other joints below the joint f.

Table 4.1: A snippet of a residue position matrix. The dimension of the full matrix is N (total pockets) by M (total residues in a protein).

name	ILE442	ARG443	MET444	TYR445	PHE446	GLU447	PRO448	GLY449	HIS450
cbd12_1_1_md_ca_model100_res_P_1	0	0	0	0	0	0	0	0	0
cbd12_1_1_md_ca_model100_res_P_10	0	0	0	0	0	0	0	0	0
cbd12_1_1_md_ca_model100_res_P_2	0	0	0	0	0	0	0	0	0
cbd12_1_1_md_ca_model100_res_P_3	0	0	0	0	0	0	0	0	0
cbd12_1_1_md_ca_model100_res_P_4	0	0	0	0	0	0	0	0	0
cbd12_1_1_md_ca_model100_res_P_5	0	0	0	0	0	0	0	0	0
cbd12_1_1_md_ca_model100_res_P_6	0	0	0	0	0	0	0	0	0
cbd12_1_1_md_ca_model100_res_P_7	0	0	0	0	0	1	1	0	1
cbd12_1_1_md_ca_model100_res_P_8	0	0	0	0	0	0	0	0	0
cbd12_1_1_md_ca_model100_res_P_9	1	0	0	0	0	0	0	0	0

4.2.3 Statistics

Hypergeometric and Wilcoxon signed-rank tests were used to examine the statistical significance of differences observed in the simulations. The former uses the hypergeometric distribution to estimate the probability of observing k number of successes out of a total of n trials (Equation 4.2). The latter is a non-parametric test to examine the difference of two related samples (Equation 4.3).

$$P(X = k) = \frac{\binom{K}{k}\binom{N-K}{n-k}}{\binom{N}{n}} \tag{4.2}$$

Here, N is the total number of samples in the population, K is the total number of successes, k is the number of observed successes, and n is the total number of trials. A success is observed when the IC value of the holo state is higher than the apo state.

$$W = \sum_{i}^{Nr} sign(x_{2,i} - x_{1,i})R_i \tag{4.3}$$

Here, W is the Wilcoxon statistic, N_r is the number of samples in which the difference between the pair is larger than zero, and R_i is the sample's rank.

4.3 Results

4.3.1 Pocket formations and surface dynamics

To assess how pockets are formed during the simulations and how different these pockets are to each other, we extracted snapshots (one per nanosecond) from the trajectories and used DoGSiteScorer to identify pockets on the surface of each protein conformation. We then counted the number of pockets found in these snapshots and plotted the counts per snapshot, see Figure 4.2. When these pockets are simply viewed as count per nanosecond, the plots for apo and holo states overlap almost completely indicating that there is no discernable difference between the states. On average, seven to nine pockets were formed per nanosecond with the minimum and maximum of three and 15 pockets, respectively.

Next, for each simulation we created a residue position matrix wherein the residues (total residues in a protein) were used as columns. If a residue was in a pocket, the residue would be marked as one (see example matrix in Table 4.1). We extracted patterns from such a matrix by using unsupervised machine learning algorithms. Specifically we used a hierarchical clustering procedure to cluster these pockets into a set of clusters. As the maximum pocket count observed in all simulations is 15, we clustered the pockets into 15 clusters and mapped the clusters back to the protein (essentially creating a heatmap, Figure 4.3).

We observed distinct patterns (fingerprints) on the heatmaps not only between apo (first column of Figure 4.3) and holo (second column of Figure

4.3) states of a splice variant, but also between splice variants. When comparing heatmaps between apo and holo states in a splice variant, we observed an increase in clusters on the holo simulations. This is particularly vivid for cbd12_1_1 and cbd12_1_2 simulations in which more clusters can be seen on the second column of the corresponding heatmaps compared to the first column. Splice variant wise (row wise), we observed reductions of observable clusters from the top row to the bottom row in Figure 4.3 for both apo and holo states.

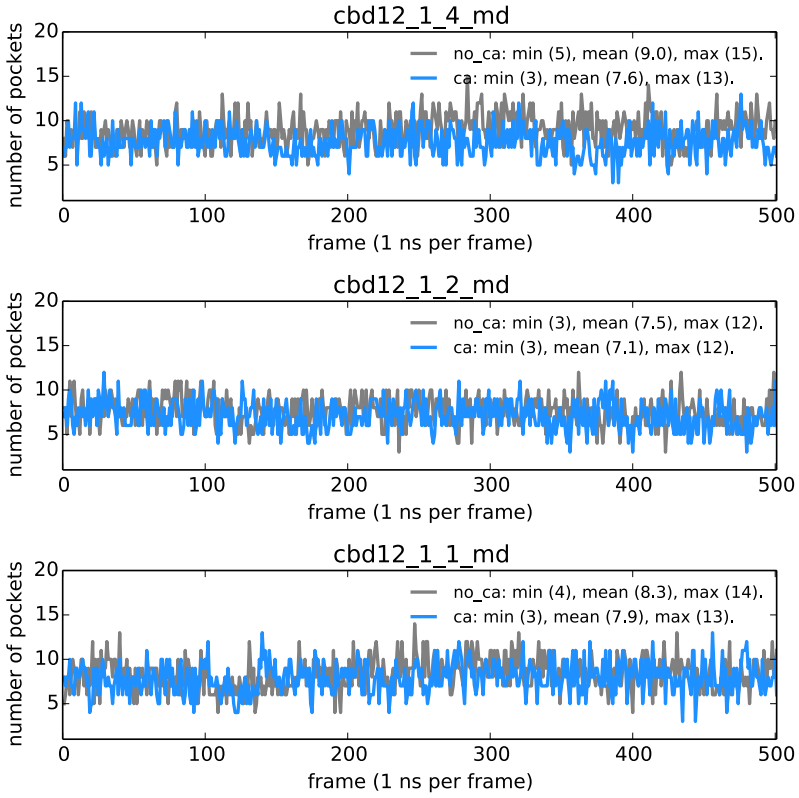


Figure 4.2: Pocket counts per MD snapshot.

4.3.2 Pocket diversity

To quantify the diversity observed in the heatmaps, we computed the mean inconsistency coefficient for each clustering result. The coefficient characterizes a merge (node) in a clustering procedure by comparing the node's height to the heights of other nodes in the same hierarchy (level). Higher values

indicate that objects within the cluster are less similar. Hence, a higher collection of inconsistency coefficient values in a clustering procedure indicates more diversity in the population. As we were only in possesion of a 500 ns long simulation for each state and splice variant, we segmented the trajectories into smaller chunks (25 ns) and treated these chunks as replicates in order to get statistical estimates. We calculated the mean inconsistency coefficient for each replicate and plotted the values in Figure 4.4. Indeed the mean coefficients for holo states (blue lines in Figure 4.4) were almost always larger than the corresponding apo states (gray lines in Figure 4.4), indicating that pockets in the holo states are more diverse when compared to the apo states.

We then performed hypergeometric and Wilcoxon signed-rank tests using these sets of mean values as samples to see if differences between the states and splice variants are statistically meaningful. Tables 4.2 and 4.3 summarize the pairwise comparisons of the simulations. We found that differences between apo and holo states within a splice variant were statistically meaningful according to both tests (p-values smaller than 0.05), apart from cbd12_1_1 (the p-value of the Wilcoxon signed-rank test for this variant is 0.057). Furthermore, differences between the apo states of the splice variants were also statistically meaningful in both tests. These observations indicate that transitioning between apo and holo states affects the population of the pockets on the protein surface suggesting a shift in the population of these pockets in response to calcium. In addition, pocket populations in apo states of these splice variants differ suggesting that the "resting states" of these splice variants differ.

Table 4.2: P-values of pairwise hypergeometric tests.

name	cbd12_1_4_ca	cbd12_1_4_no_ca	cbd12_1_2_ca	cbd12_1_2_no_ca	cbd12_1_1_ca	cbd12_1_1_no_ca
cbd12_1_4_ca	-	0.002	0.205	0.0	0.115	0.0
cbd12_1_4_no_ca	-	-	0.115	0.0	0.248	0.011
cbd12_1_2_ca	-	-	-	0.0	0.044	0.011
cbd12_1_2_no_ca	-	-	-	-	0.0	0.0
cbd12_1_1_ca	-	-	-	-	-	0.044
cbd12_1_1_no_ca	-	-	-	-	-	-

4.3.3 Pocket Density

Upon observing meaningful differences in the diversity of pockets found on the surface of these proteins, we were interested to see how such differences manifest on their structures. To tackle this we aggregated the data from the residue position matrix of a simulation by summing all columns in the

Table 4.3: P-values of pairwise Wilcoxon signed rank tests.

	cbd12_1_4_ca	cbd12_1_4_no_ca	cbd12_1_2_ca	cbd12_1_2_no_ca	cbd12_1_1_ca	cbd12_1_1_no_ca
cbd12_1_4_ca	-	0.018	0.317	0.0	0.019	0.0
cbd12_1_4_no_ca	-	-	0.232	0.001	0.536	0.018
cbd12_1_2_ca	-	-	-	0.0	0.032	0.003
cbd12_1_2_no_ca	-	-	-	-	0.002	0.023
cbd12_1_1_ca	-	-	-	-	-	0.057
cbd12_1_1_no_ca	-	-	-	-	-	-

matrix. This provided us with a measure of "pocket density" for each residue in a protein. We then mapped this density information to the three dimensional structure of the corresponding protein and visualized these structures in supplementary Figures S5 to S7. When the structures were examined from this perspective, we found notable shifts of pocket densities between the apo and holo states. In the absence of calcium, pockets primarily populated the linker region (this is the region where calcium ions bind, between CBD1 and CBD2) and to a lesser extent the CBD1 and CBD2 regions. In the presence of calcium, we noticed that pocket densities on CBD2 increased (more red segments on the holo state of CBD2 compared to the apo state). Binding to calciums appears to directly influence pocket formations on the protein surface particularly on CBD2.

While the shifts in pocket density on CBD2 across the splice variants were similar, such shifts were not uniformly found on the other two regions. To visualize this, we calculated the difference between the density of the holo and apo states and mappped the difference to the structures (Figure 4.5 to 4.7). In cbd12_1_4 (Figure 4.5), the linker region was similarly dense in apo and holo states (rendered mostly in white) whereas pockets were more dense for the apo state in its CBD1 domain (rendered in blue and white). In cbd12_1_2 (Figure 4.6), pockets were more dense in the holo states for both its linker region and CBD1 domain (rendered mostly in red). Finally, in cbd12_1_1 (Figure 4.7), pockets on the linker region were denser in its apo state compared to the holo state (rendered in blue) whereas on its CBD1 domain pockets were more dense in its holo state (rendered mostly in red). In other words, the holo state of cbd12_1_4 showed the smallest density changes compared to its apo state, cbd12_1_2 recorded the largest density changes, and changes in cbd12_1_1 were in between these two extremes. To a certain extent, these observations highlight the different phenotypic responses of the splice variants.

4.4 Discussion

Observing protein dynamics in real life remains to be a large challenge. Thus computational tools are often used to achieve this objective. For instance, a protein dynamics study using a combination of simulations and essential dynamics [135] on protease proteinase K revealed that the protein has a highly flexible binding site and that the removal of potassium influences the global conformational flexibility of the protein but decreases the local flexibility of the binding site. Furthermore, the study demonstrated that the removal of substrate relates to concerted motions that can be connected to binding, orientation or to product release. Similarly here, we used MD simulations to observe the dynamics of pocket formation and deformation on the surface of splice variants of NCX regulatory domains. We found that different regions in each splice variant uniquely respond to the absence and presence of calciums. In splice variants that respond to calciums (either by activation: cbd12_1_4 or by inhibition: cbd12_1_1), the linker region was sparsely populated by pockets when calciums were present. This indicates that the binding site is less flexible and that calciums bind tightly. In addition, the CBD1 domain of these two variants exhibited contrasting behaviours. The former was sparsely populated with pockets when calcium ions were present whereas the latter was more densely populated. Their CBD2, however, behaved rather similarly. The linker region of the variant that does not respond to calciums (cbd12_1_2), on the other hand, was densely populated by pockets indicating that the binding site is more flexible and that calciums bind less tightly to the binding site. Its CBD1 and CBD2 domains were densely populated by pockets as well indicating a rather flexible surface in general.

Another interesting aspect to focus on is how signals (from binding to calciums) may be propagated across these domains. To examine this, we ranked the residues based on the difference of density between the holo and apo states and selected ten residues with the largest density difference. These residues were then mapped to the corresponding structures (supplementary Figures S8 to S10). Interestingly, these residues were localized around the linker region and at the tip of the CBD2 domain in all splice variants. These observations suggest that binding to calciums increases the dynamics on surfaces around the linker region and at the very tip of the CBD2 domain. It has been hypothesized that upon binding to regulatory calciums, signals may travel from the binding site towards the CBD2 domain. A structure-dynamics study of the same NCX splice variants revealed that binding to calciums causes the backbone on CBD2 to become more rigid and that large conformational changes are unlikely to happen within such constraints [126]. More recently, backbone dynamics were demonstrated to play critical roles

on other NCX isoforms as well [136]. It appears that in the absence of large conformational changes and the presence of constraints on the backbone, the protein may include the dynamics on its surface as part of the signal propagation mechanisms. In other words, the dynamics on the surface of the protein might contribute to the overall structure-dynamics and signal propagation on these proteins.

Naturally, observations from this study can benefit from additional samplings. In order to obtain statistics, we opted to split up our 500ns long simulations into smaller chunks and treated this chunks as samples. This allowed us to calculate the necessasary statitistics to support our findings. However, the independence (of the samples) could be improved by running multiple simulations instead of segmenting a trajectory from an MD simulation. An aspect we aim to resolve in the next round of studies.

4.5 Concluding remarks

Protein conformational dynamics remain integral to the mechanistic characterization of proteins. Such dynamics are often examined by computational tools such as MD simulations. In this work, we investigated how regulatory calcium ions influence the dynamics on the surface of NCX variants by simulating their apo and holo states and examining the formation/deformation of pockets on the surface. We found that when calcium ions were present, pockets were more diverse and that the diversity profiles varied across the splice variants. Furthermore, we found that the density of the pockets on CBD2 increased when calcium ions were present in all variants. In contrast, the pocket density on the other two regions (linker and CBD1) responded in a unique manner to calciums. Finally, we observed that residues displaying the largest density difference between the holo and apo states were localized around the linker region and the tip of CBD2 indicating that the surface dynamics might contribute to the signal propagation on these proteins.

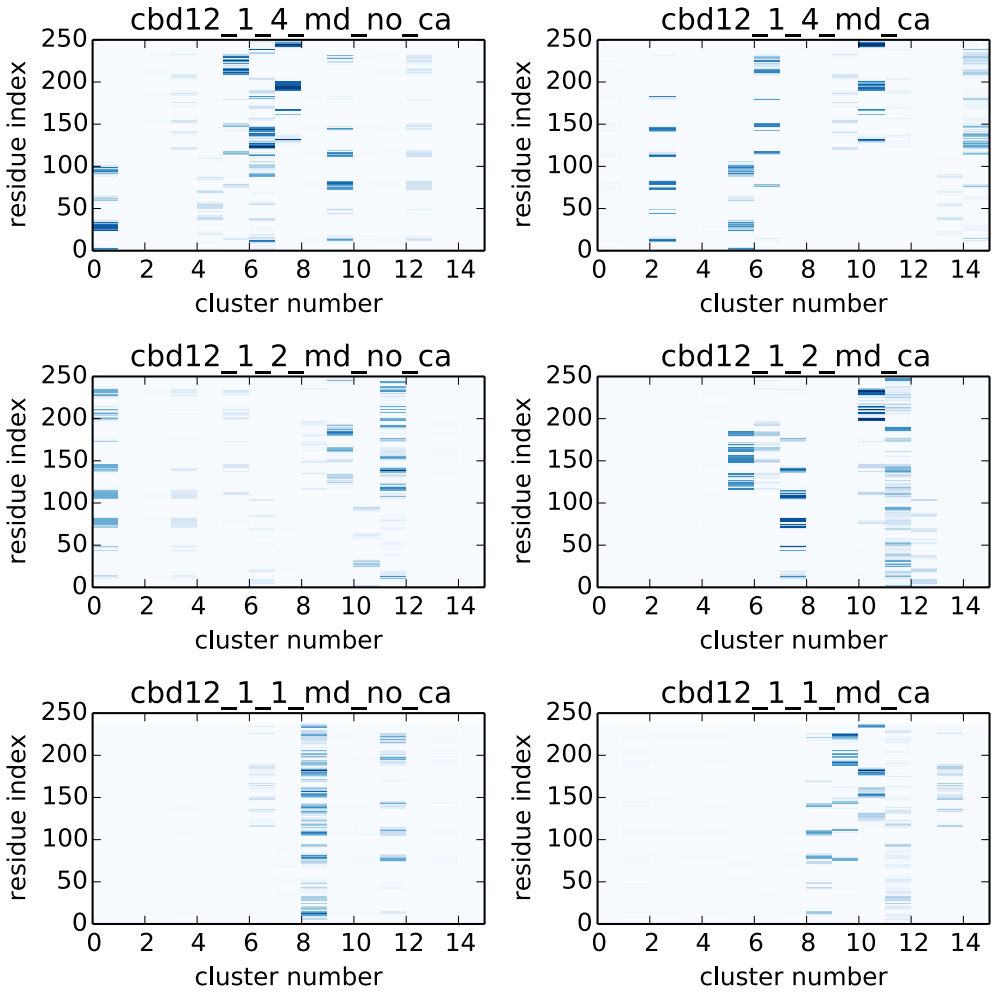


Figure 4.3: Heatmaps for clusters derived from the MD simulations.

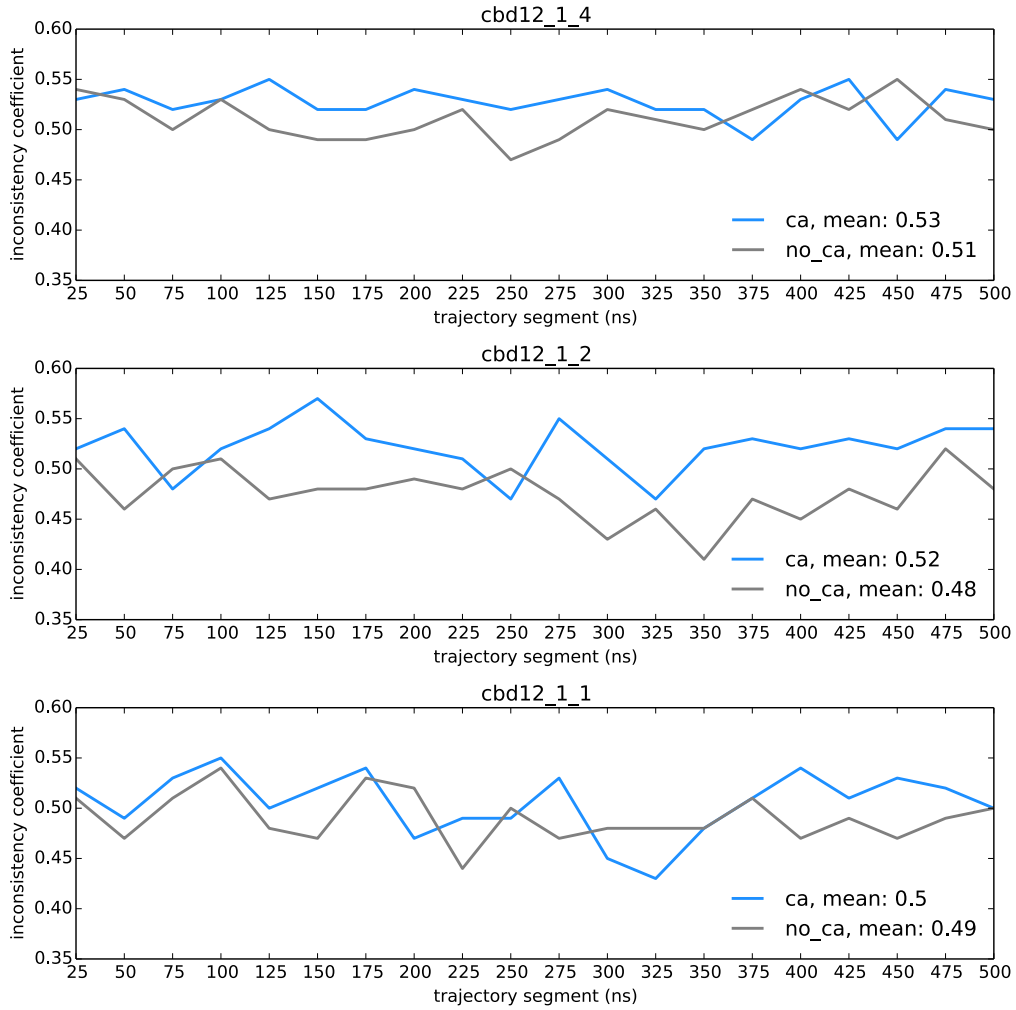


Figure 4.4: Mean of inconsistency coefficient.

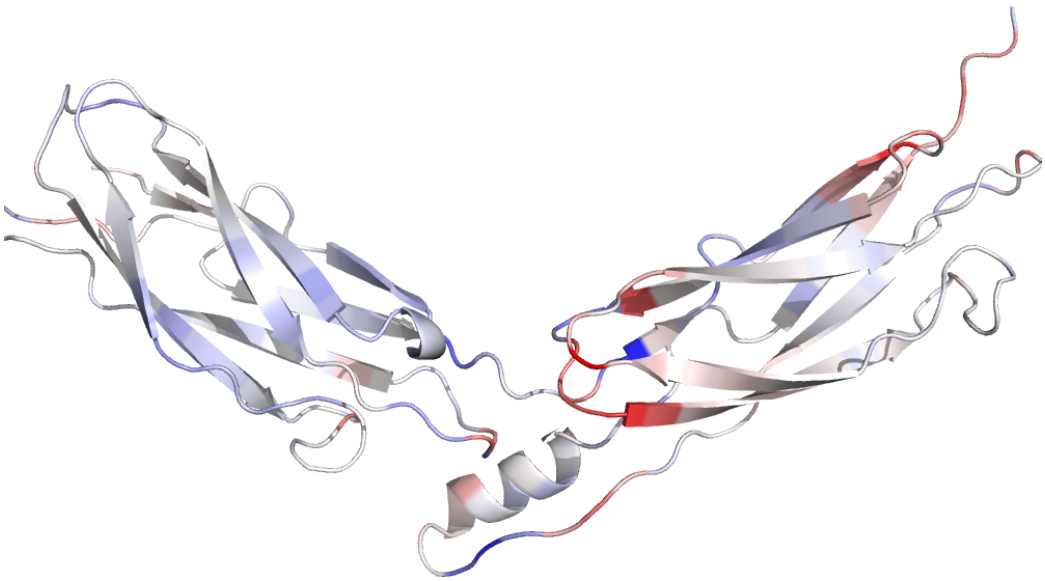


Figure 4.5: The difference of the pocket density in holo and apo structures of cbd12_1_4 (activation). Blue: the pocket density in apo is higher than holo; white: the density in apo and holo is similar; red: the density in holo is higher than apo.

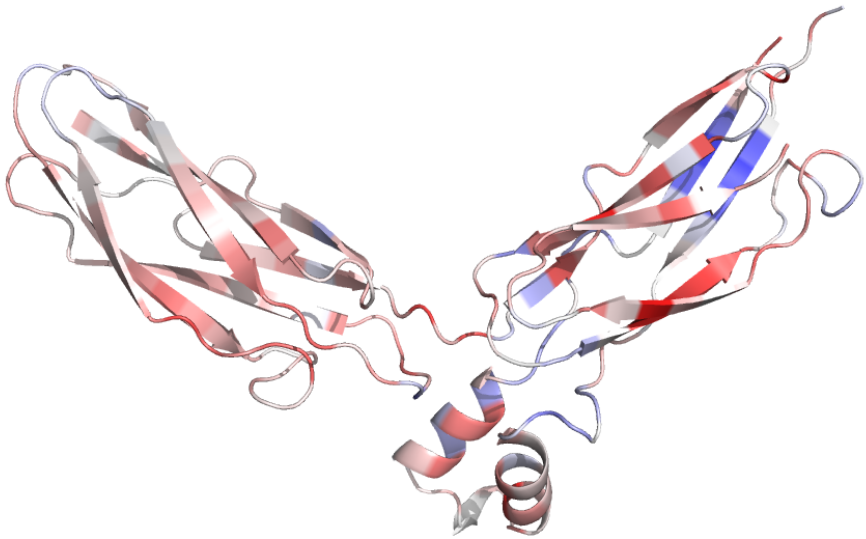


Figure 4.6: The difference of the pocket density in holo and apo structures of cbd12_1_2 (no response). Blue: the pocket density in apo is higher than holo; white: the density in apo and holo is similar; red: the density in holo is higher than apo.

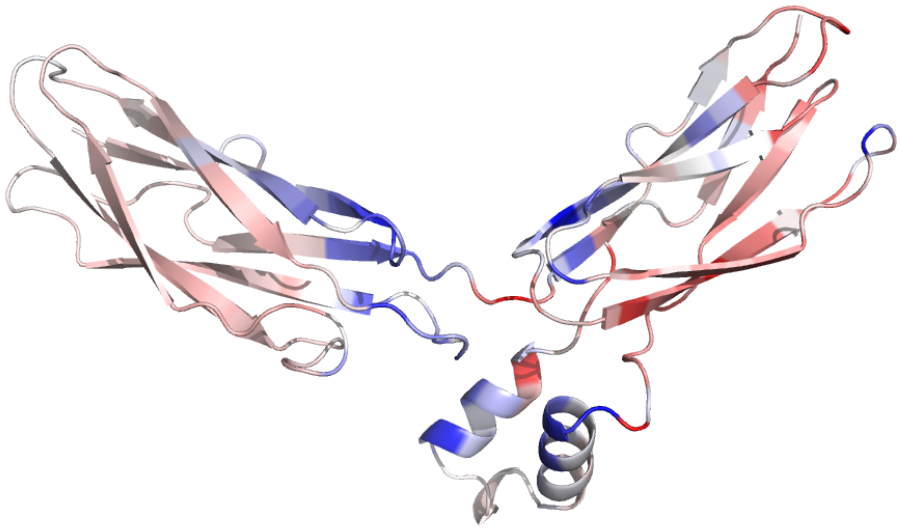


Figure 4.7: The difference of pocket density in holo and apo structures of cbd12_1_1 (inhibition). Blue: the pocket density in apo is higher than holo; white: the density in apo and holo is similar; red: the density in holo is higher than apo.

Chapter 5

Concluding Remarks and Outlook

A German socialist Friedrich Engels once said: "life is the mode of action of proteins" [137]. This statement not only captures the roles proteins play in life, it also highlights the significance of proteins as machineries of life. When cells are dried, one half of their mass is protein, a humble bacterial cell contains up to four million proteins, whereas mammalian cells may contain a staggering 10^{10} millions of proteins. Proteins can function as catalysts in the form of enzyme, they provide structural support in cells, and they allow cells to communicate with each other by functioning as a signalling molecule or as a receptor.

This dissertation compiles works that take advantage of computational tools such as molecular docking, virtual screening, and molecular dynamics in combination with machine learning to learn about proteins. Molecular docking (and virtual screening) allows one to find an optimal match between proteins and ligands. Molecular dynamics provides a way to look beyond the static limitation of molecular docking by allowing the protein of interest to move around in space. Finally, machine learning algorithms allow one to learn and generate insights on data generated from the former two approaches.

Chapter 2 describes a Naive Bayes Classifier capable of finding virtual screening targets with high enrichment factor. The classifier was trained on over four hundred conformations from molecular dynamics simulations and crystal structures of eleven nuclear receptors. An adaptive synthetic minority over sampling technique was employed to handle the imbalance in the dataset prior to training. The classifier trained on the balanced dataset was more sensitive compared to the one that was trained on the vanilla (standard) dataset. It was able to find targets with enrichment factor value reaching two-fold of the reference crystal structure. Such findings highlight

the impact of data preprocessing on the performance of a machine learning method.

Chapter 3 describes a Naive Bayes Classifier and an Artificial Neural Networks model capable of discriminating and prioritizing allosteric targets. The former was trained on allosteric and orthosteric sites from over 250 proteins. The optimal Naive Bayes Classifier was able to discriminate allosteric and orthosteric sites with an accuracy reaching over 90%. The latter was trained on almost 2000 allosteric and orthosteric sites. The optimal Artificial Neural Network model used a two stage procedure; the first network was used to recover missing information from the dataset and the second network was used to rank (prioritize) allosteric sites in the dataset. Such a model was able to prioritize allosteric sites correctly for over 80% of proteins in our test dataset.

Chapter 4 examines the formation and deformation of pockets on the surface of calcium exhanger proteins. Here, molecular dynamics simulations were used to sample the conformational dynamics of three calcium exchangers with three distinct phenotypes (activation, no response, inhibition) for 500 nanoseconds. Hierarchical clustering was then used to profile the pockets harvested from these simulations. Our results revealed meaningful differences in the pocket diversity and density of the apo and holo proteins. Furthermore, these changes were found to localize around areas that are critical to the function of the proteins.

In addition to works compiled in this dissertation, I am also a contributor to ongoing works on evaluating various pocket identification tools with Zhao Yuan et al. and on cytokine carrier transportation with Bin Qu et al. (manusripts are in preparation).

In conclusion, combining well established methods in structural biology such as molecular docking and molecular dynamics with machine learning can be very potent. Not only such an approach allows one to built predictive models, unearth trends and insights from datasets using supervised machine learning techniques (as in chapter 2 and 3), it also affords exploratory studies (chapter 4). In the future, it will be beneficial and of interest to generalize some of the machine learning models described in this dissertation. More specifically, the Naive Bayes Classifier described in chapter 2 was trained on eleven nuclear receptors, however, in total there are around 20 members in the nuclear receptor family. More exhaustive datasets can be compiled and used to improve the current models. Similarly, models in the latter chapters can also benefit from more exhaustive datasets. Fortunately, we now live in the age of data. I firmly believe that someday soon, we will get there!

Chapter 6

Appendix

6.1 Supplementary Materials

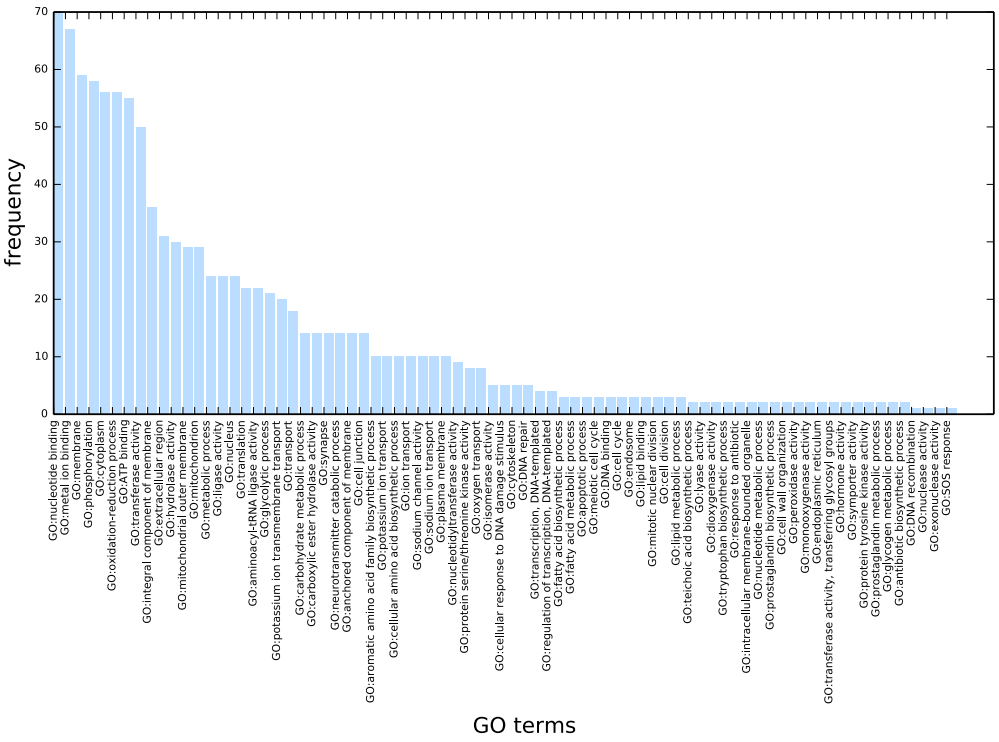


Figure S1: Allosteric sites mapped to gene ontology anotations.

Table S1: Average(mean), Minimum (min) and Maximum (max) EF1% values for ENRI, POVME, RMSD and VOM.

PDB ID	ENRI mean, min, max	POVME mean, min, max	RMSD mean, min, max	VOM mean, min, max
1SJ0	38.4, 25, 44	39.3, 32, 47	40.4 33, 45	36.3, 31, 45
3BQD	12.02, 5.8, 20	12.79, 8.9, 16	13.63, 9.3, 18	16.1, 11, 20
3L3X	19.1, 13, 24	17.7, 12, 22	17.46, 9.6, 22	19.1, 14, 24
2P54	20.3, 15, 27	23.3, 14, 34	22.6, 16, 30	25.5, 15, 32
2A3I	17.62, 3.2, 28	8.69, 0, 21	8.7, 1.1, 17	8.44, 2.1, 22
3SP9	23.7, 15, 33	25.6, 16, 39	23.6, 10, 32	26.9, 18, 38
2GTK	9.12, 3.7, 15	12.63, 4.9, 20	11.79, 6.2, 17	10.61, 6.2, 17

Table S2: PDB IDs of allosteric and orthosteric sites in the AO dataset.

PDB IDs											
3ZQH	1U33	1XD1	1XD0	4DLT	4DLU	4DLW	2XJC	4MIB	3NQS	1N5M	1DD7
3KGF	4AN9	1WBV	4AN3	4AN2	3BAJ	2V5Z	2ZB2	2HA4	3FI0	3LU7	3N1V
3N1W	2Q72	4FKZ	2V60	2V61	1E7C	1B2Y	2BYB	2C65	3ZYX	4LMN	2D41
2QB4	3EQC	3EQB	3EQG	1EGY	3CMU	4LEG	4A79	2G50	2OV4	1QHA	1VM1
3CEP	3O8P	3N45	1MAU	3HL8	2P55	3KH5	3ZCW	2BK3	3N25	2D5X	3F3T
3F3U	4A7A	2XFQ	2XCW	3N3L	3OLE	4C7B	1CZA	1SFQ	1XH2	1UA3	3O6I
4A50	4Q9M	3PXZ	3PXQ	2ORT	3NUE	4G0N	4CLZ	3NUD	4CRT	3PY1	3P4W
3BEO	3LBI	3LBH	3RZI	3QH0	1PPI	1IWH	1THC	3PP1	3DY7	3N5H	4EAG
3N5J	2HA5	3N46	3V01	2HA6	2HA0	3V04	4OYO	3N49	4HO2	4OYP	3OC1
3ORN	3OLD	3P50	4NES	1Q5O	3N6K	3OS3	1XCX	3ZG1	1I6L	1I6M	1S9J
1S9I	2DTI	3OIU	3OIW	3ZLL	2ORO	1J07	1Q43	1HKB	4ANB	1EUP	4MK8
4MK7	2ORS	2ORR	2ORQ	2ORP	1OJ9	4B3U	2HA7	3L2L	3K8Y	3K8S	

Table S3: PDB IDs of allosteric protein-ligand complexes in the APLC dataset.

PDB IDs											
1FX2	4HSG	4B1F	3O2M	1PFK	2XJC	2JHR	4MBS	4BBG	3LU6	3N1V	4NBN
3M6F	3MW9	1I7S	1FIY	1OF6	1GZ3	4EBW	4P3H	1UXV	1B86	3G86	3QEL
2YLO	2RD5	3PJG	4JA8	3H30	2VD3	4ETZ	3H6O	4JAF	3ZCW	2D5X	3EPS
3KCG	1T49	4IG3	4KKO	4IO8	4BQH	3MKS	4C7B	4EJ8	3O96	2BU8	2BU7
2BU6	3PXF	1NSG	3QOP	3BZ7	4G8O	3VQ8	3L3V	1H5S	2JFN	2Y0P	3QH0
1QP0	1ZDS	1PCQ	4EAJ	2BE9	3LSW	1EM6	1DKU	3KF0	2HIM	2I7N	4I1R
3F6G	2PUV	1KZ8	4KFB	3H1V	1NJJ	1CE8	2A69	3F9N	2BXA	2Q5O	2YHD
3UO9	4EO6	1HKB	4AHS	3KGF	2VPR	3PG9	1W96	4NLD	11BG	3LAJ	

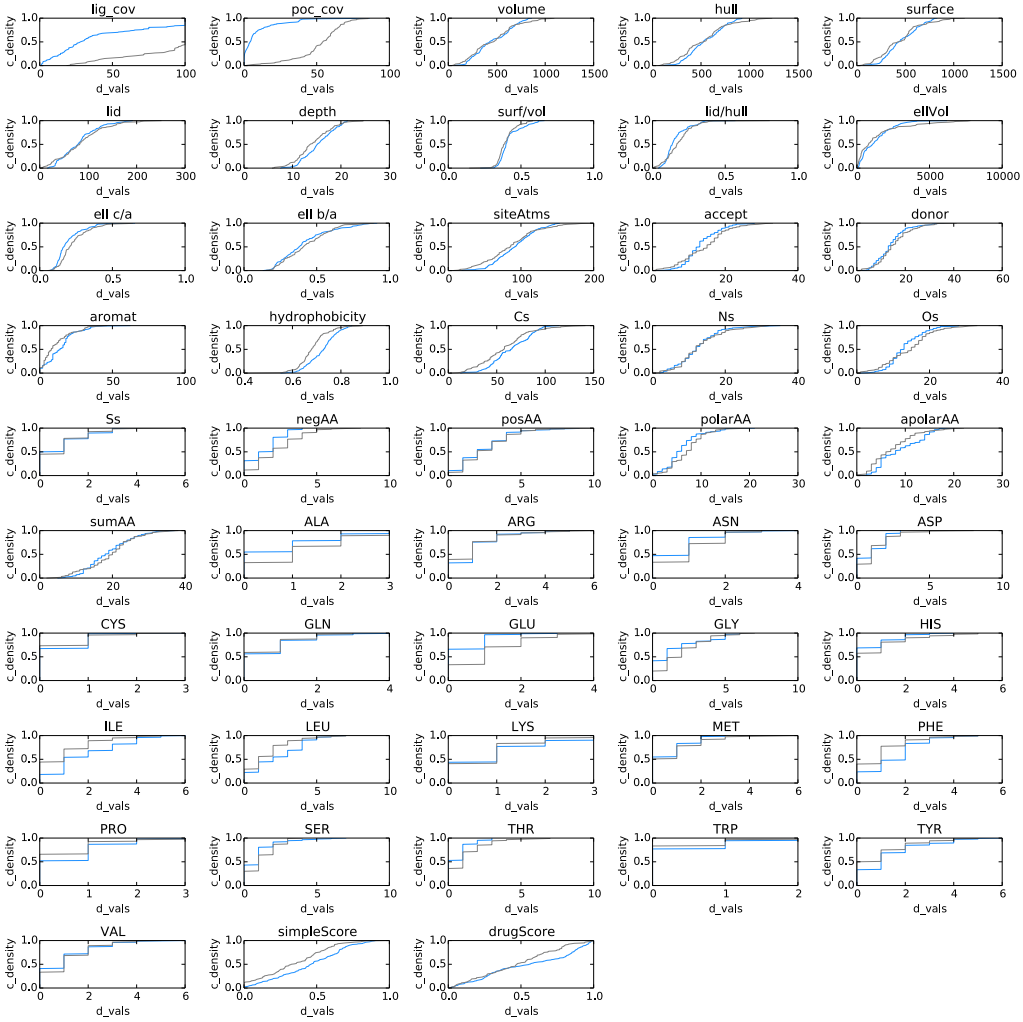


Figure S2: Cumulative density plots of the descriptors for allosteric pockets (blue) and orthosteric pockets (grey). Shown on the x and y axes are descriptor values (d_vals) and cumulative densities (c_density).

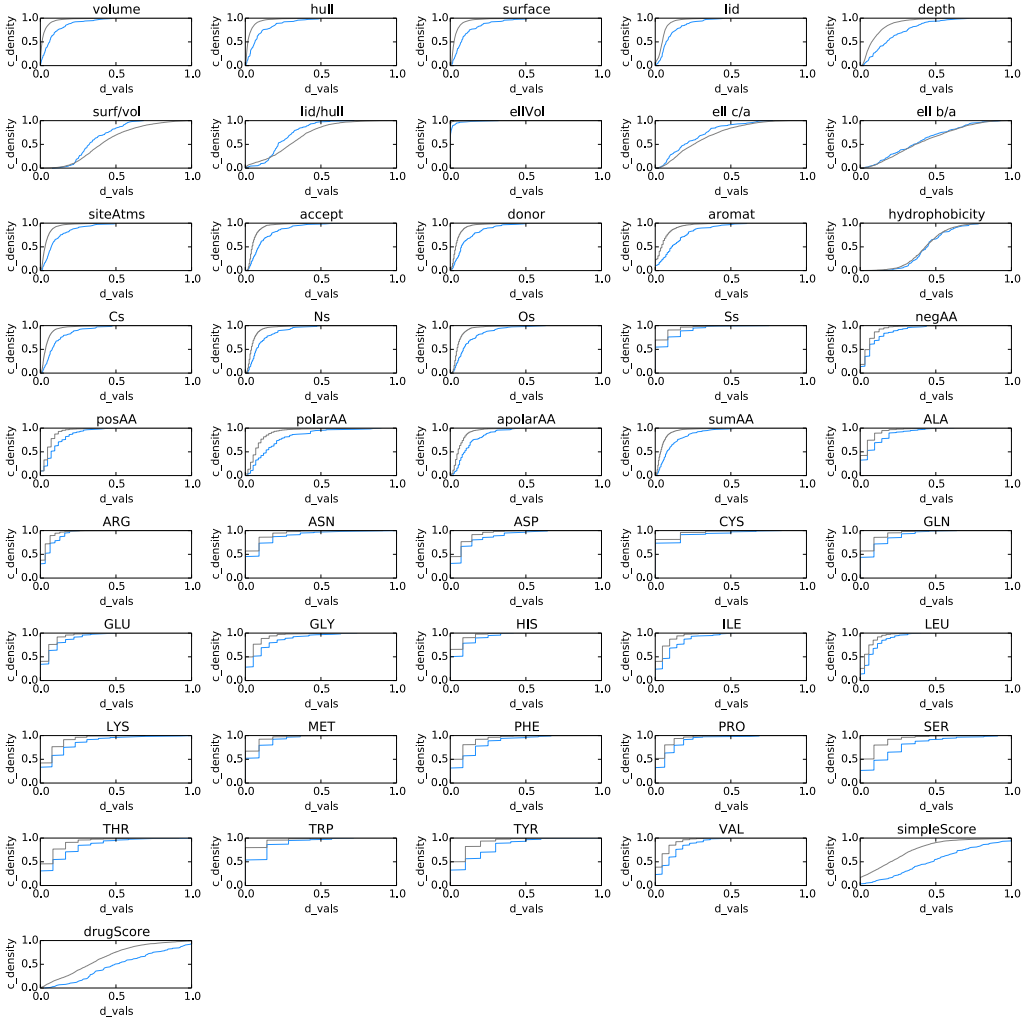


Figure S3: Cumulative density plots of descriptors for allosteric pockets (blue) and non-allosteric pockets (grey). Shown on the x and y axes are descriptors values (d_vals) and cumulative densities (c_density), respectively.

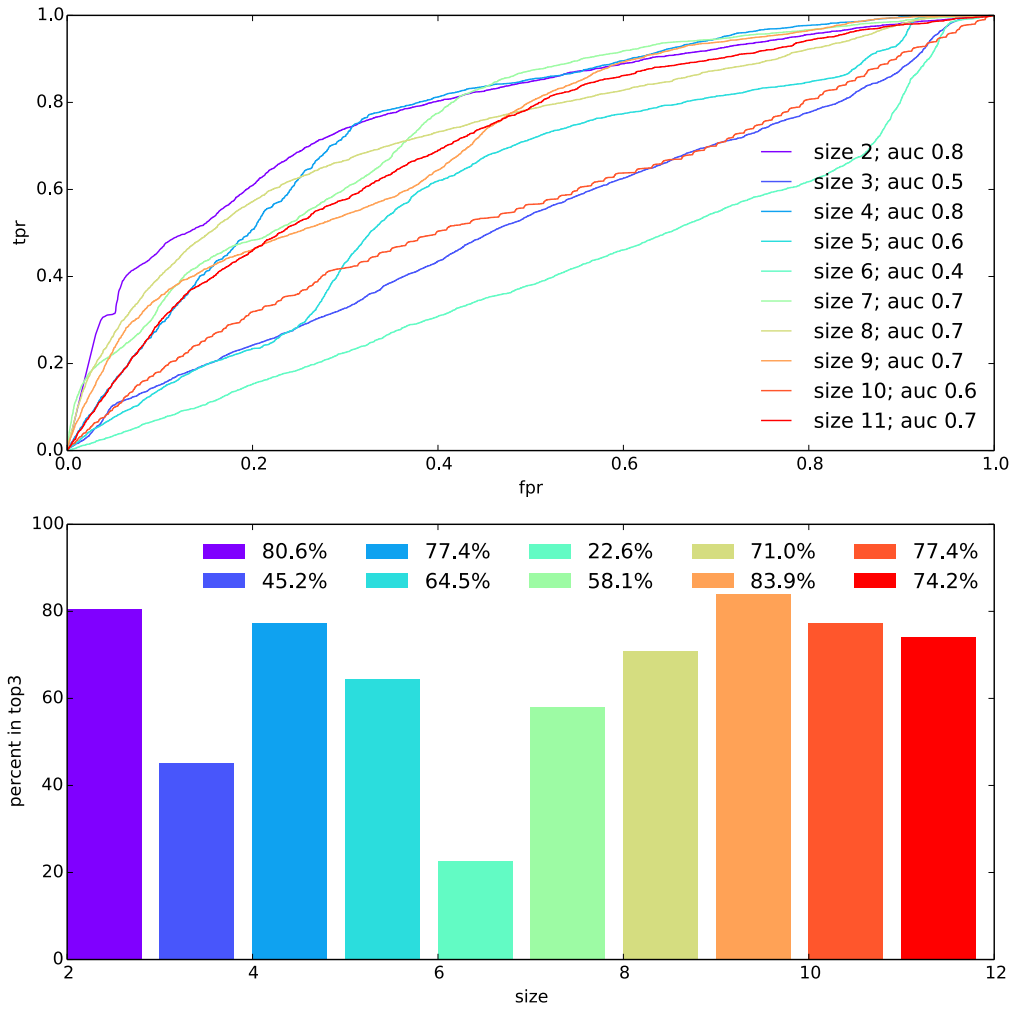


Figure S4: Regression-classification ANN: ROC plots (top panel) and the percentages of finding allosteric pockets among the top three of the ranked pockets on all proteins in the test dataset (bottom panel).

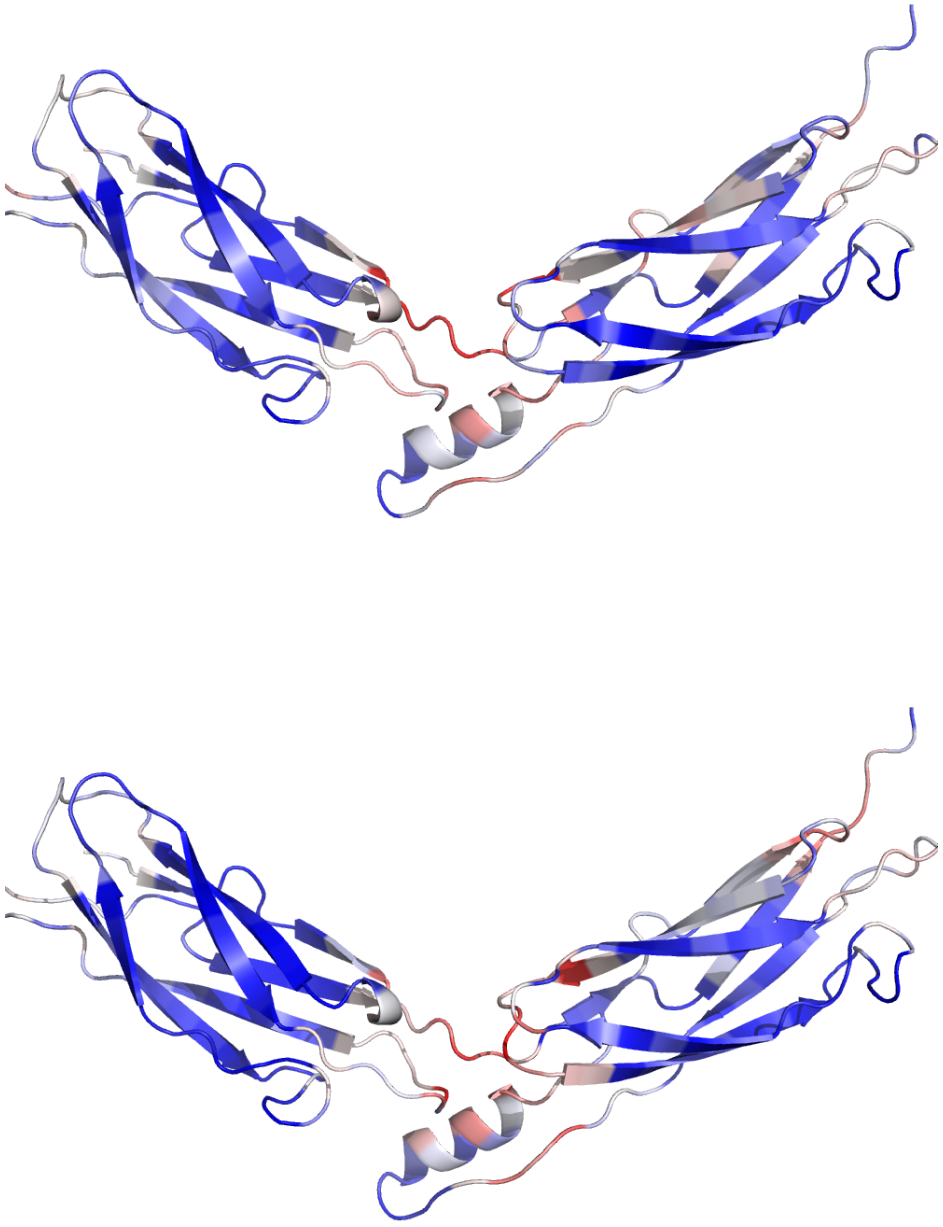


Figure S5: Pocket density on cbd12_1_4 (activation) apo (top) and holo (bottom), blue, white and red represent sparse, medium, and dense regions.

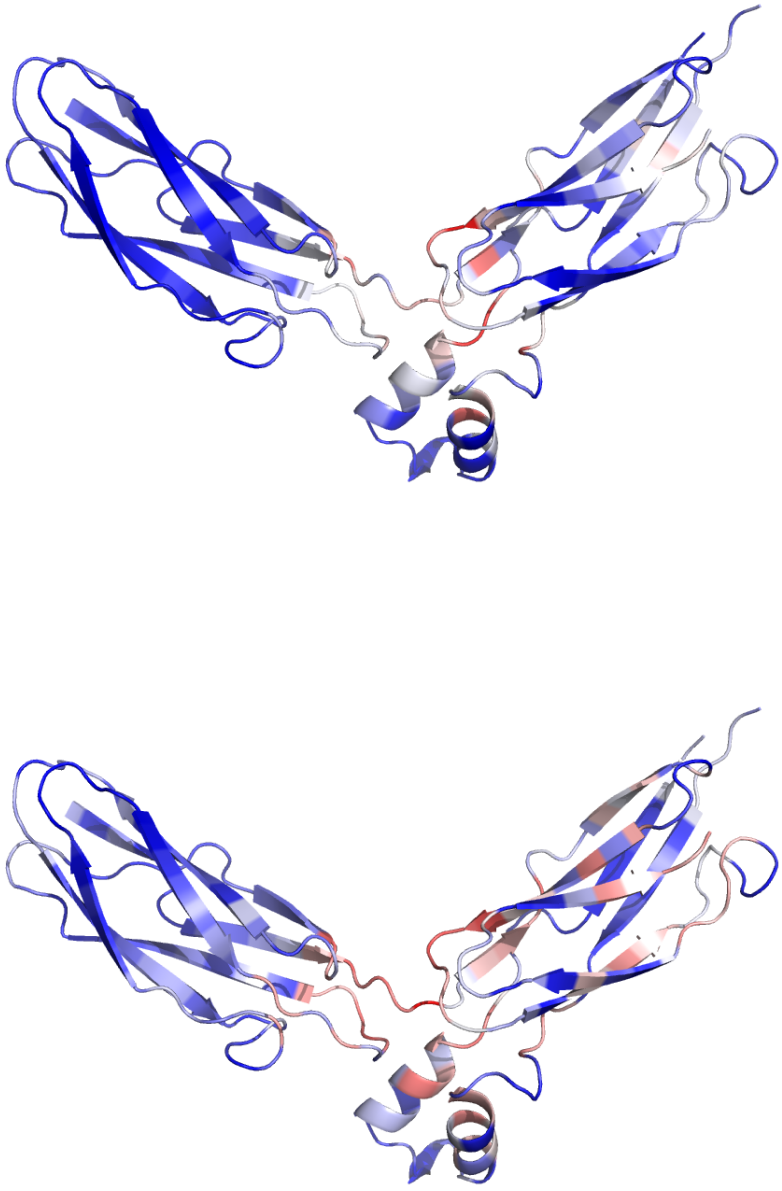


Figure S6: Pocket density on cbd12_1_2 (no response) apo (top), holo (bottom), blue, white and red represent sparse, medium, and dense regions.

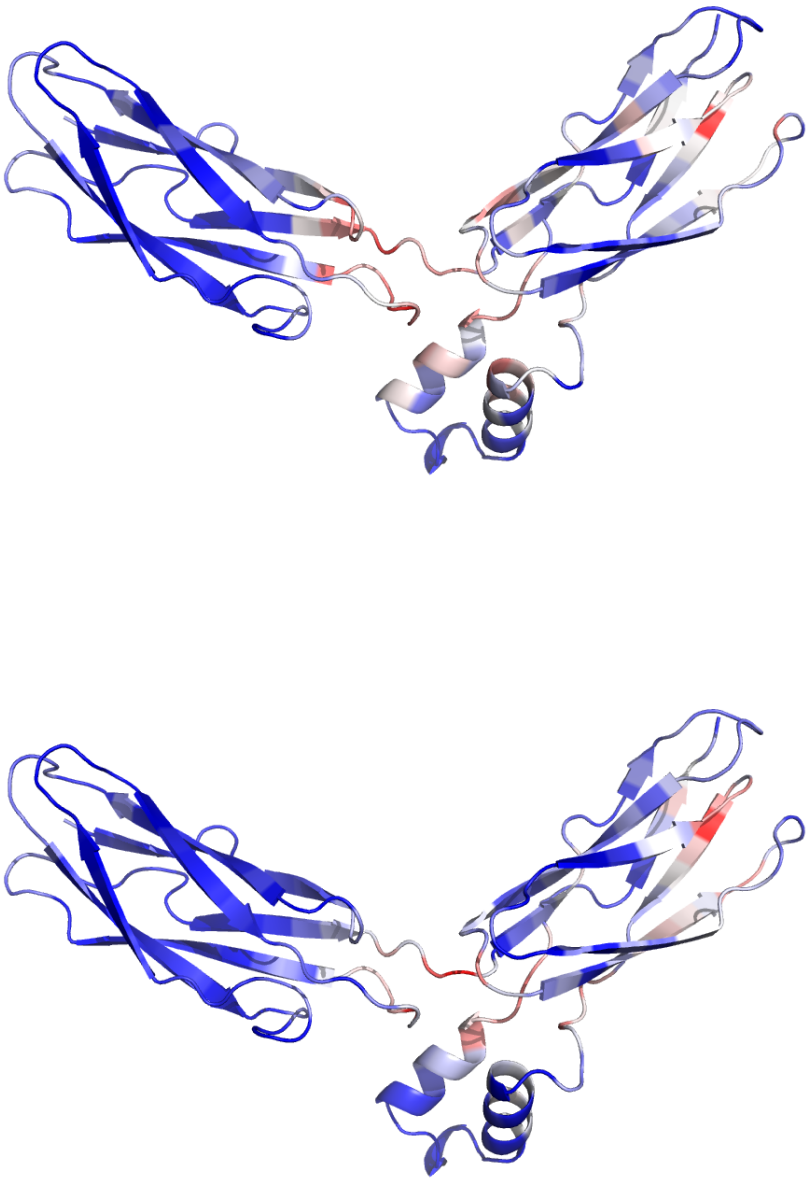


Figure S7: Pocket density on cbd12_1_1 (inhibition) apo (top) and holo (bottom), blue, white and red represent sparse, medium, and dense regions.

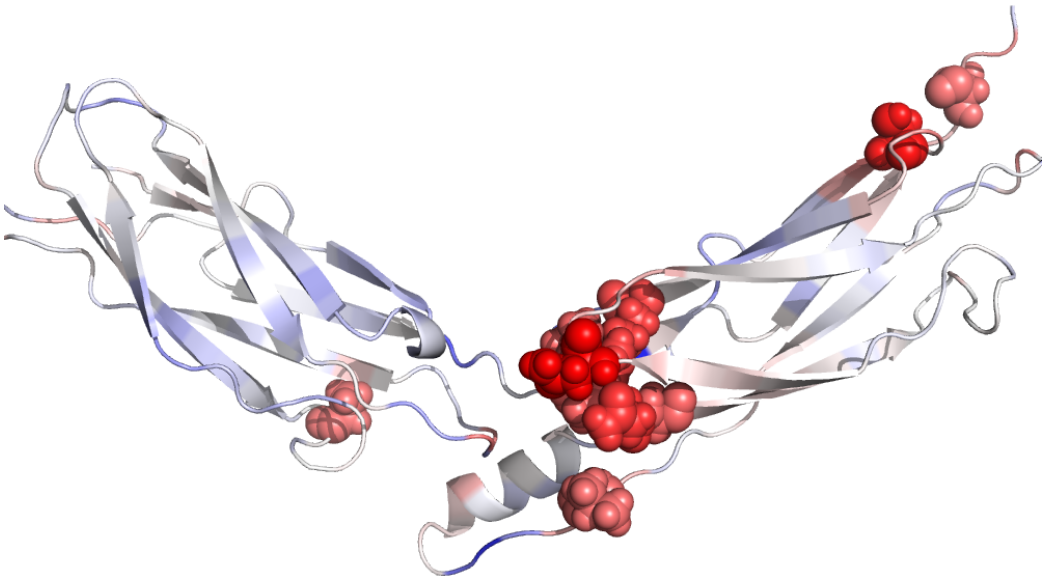


Figure S8: The difference of the pocket density in holo and apo structures of cbd12_1_4 (activation). Blue: the pocket density in apo is higher than holo; white: the density in apo and holo is similar; red: the density in holo is higher than apo. Ten residues with the largest density are rendered as red spheres.

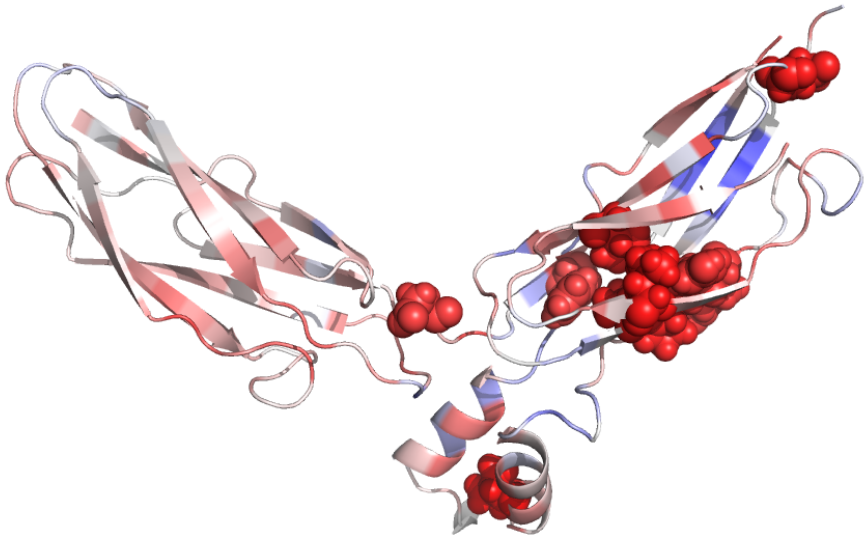


Figure S9: The difference of the pocket density in holo and apo structures of cbd12_1_2 (no response). Blue: the pocket density in apo is higher than holo; white: the density in apo and holo is similar; red: the density in holo is higher than apo. Ten residues with the largest density are rendered as red spheres.

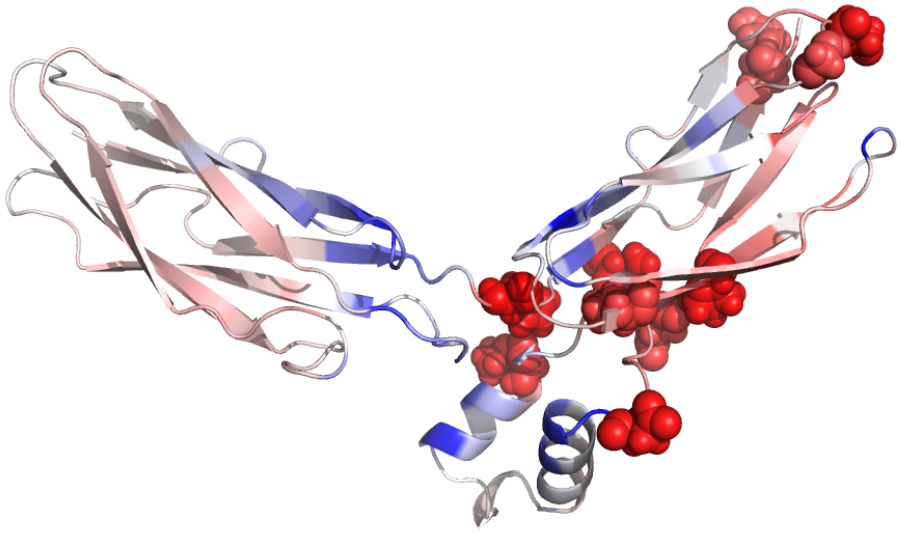
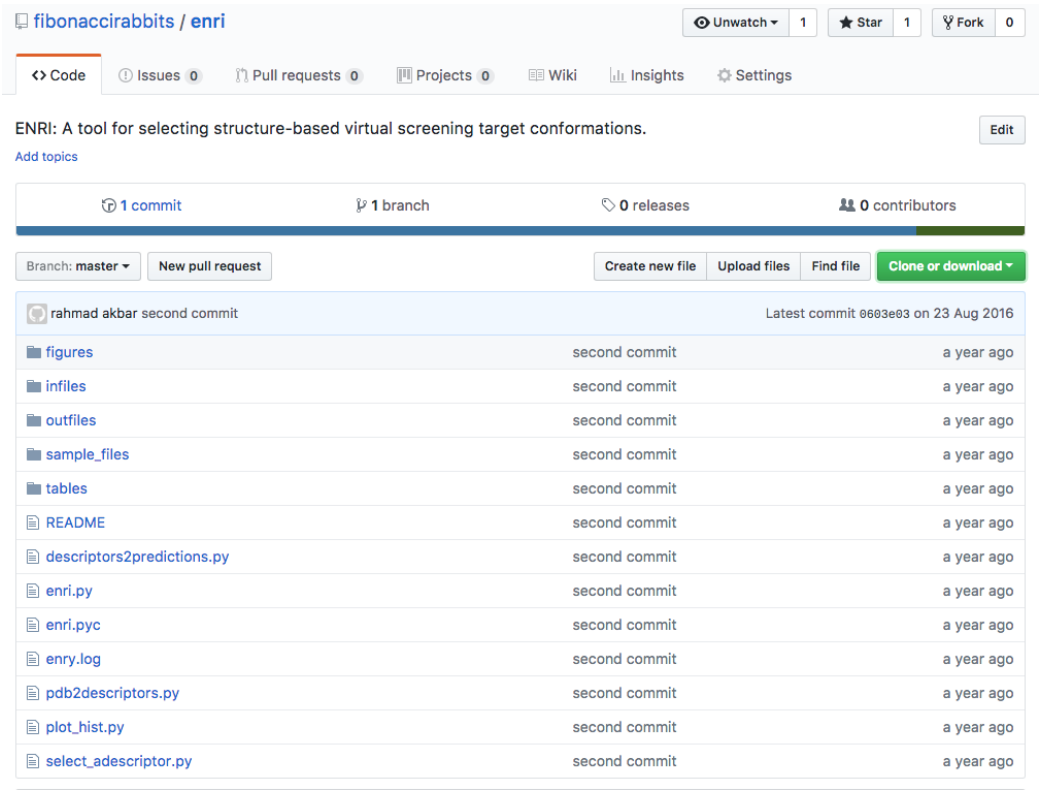


Figure S10: The difference of the pocket density in holo and apo structures of cbd12_1_1 (inhibition). Blue: the pocket density in apo is higher than holo; white: the density in apo and holo is similar; red: the density in holo is higher than apo. Ten residues with the largest density are rendered as red spheres.

6.2 Developed Programs and User Manuals

6.2.1 ENRI

Available at `https://github.com/fibonaccirabbits/enri`



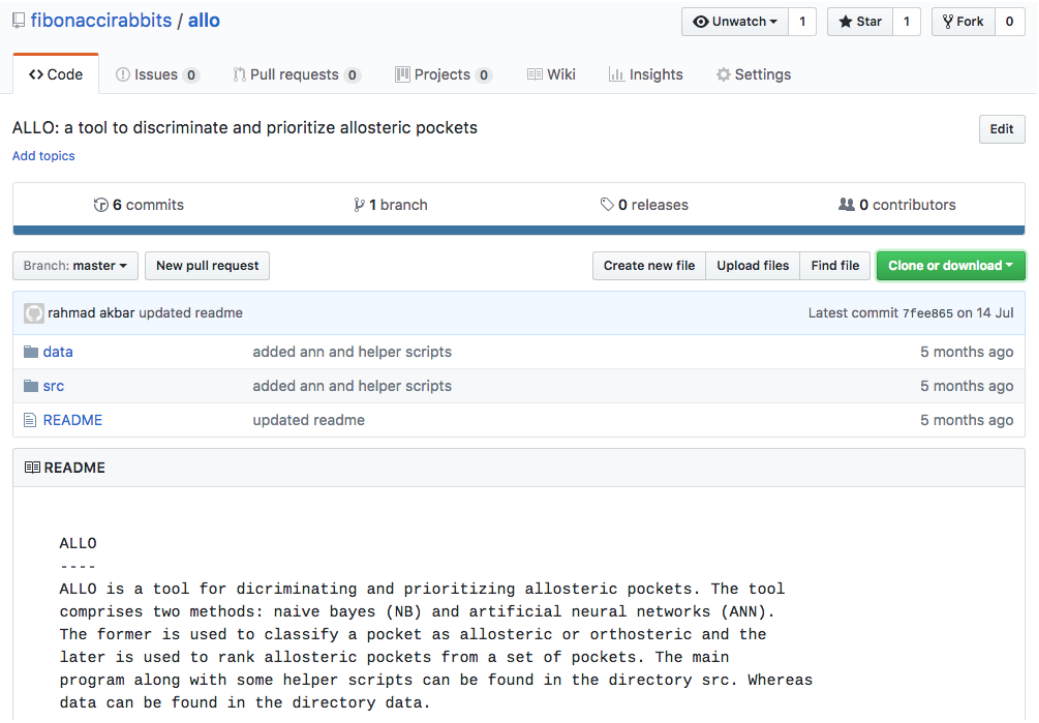

```
ENRI
----
ENRI is a tool for selecting structure-based virtual screening targets.
The tool is a binary classifier coupled with a synthetic over-sampling
procedure. ENRI, currently, comprises four programs: enri.py,
pdb2descriptors.py, descriptors2predictions.py, plot_hist.py.

PREQUISITES
-----------
A fully funtional DogSiteSCorer.
A python plotting library: matplotlib.*
A python tabulation library.**
```

```
*only when plotting is desired
**only when tabulation is desired

enri.py
------
This is the main program where all of ENRI's functionalities are defined.

pdb2descriptors.py
------------------
Extracts pockets and descriptors from pdb files.
Interfaces with DoGSiteScorer.
Please make sure you have fully funtional DoGSiteScorer.

INPUT: pdb_path
OUTPUT: desc_merged.txt
ARGUMENTS: pdb_path
USAGE: python pdb2descriptors.py pdb_path
EXAMPLE: python pdb2descriptors.py /enri_rc8/sample_files/pdbdir

descriptors2predictions.py
--------------------------

Predicts and writes an output file for top n predicted conformations. The
output file is written to the input directory

INPUT: desc_merged.txt
OUTPUT: *predicted*.txt
ARGUMENTS: input_path, number of desired output (n),
           over-sampling paramter (beta), ranker (wp or p)
USAGE: python descriptors2predictions desc_merged.txt, n, beta,ranker
EXAMPLE: python descriptors2predictions.py /path/to/input.txt 10 0.5 wp

plot_hist.py
------------
Plots histogram from a desc_merged.txt file.

INPUT: file_path
OUTPUT: *descriptorname.pdf
ARGUMENTS: file_path
USAGE: python plot_hist.py file_path
```

```
EXAMPLE: python plot_hist.py /path/to/input.txt
```

6.2.2 ALLO

Available at `https://github.com/fibonaccirabbits/allo`




```
ALLO
----
ALLO is a tool for dicriminating and prioritizing allosteric pockets.
The tool comprises two methods: naive bayes (NB) and
artificial neural networks (ANN). The former is used
to classify a pocket as allosteric or orthosteric and the
later is used to rank allosteric pockets from a set of pockets.
The main program along with some helper scripts
can be found in the directory src.
Whereas data can be found in the directory data.
```

```
PREQUISITES
-----------
Python libraries: Numpy, scipy
Output file from the program DoGSiteScorer.

nb.py
-------
An implementation of a naive bayes model.


predict_nb.py
-------
labels input as allosteric (A) or orthostreic (O)
Uses NB model (nb.py)
Uses ao.tsv
Usage: python predict_nb.py input_file.txt


example:

In your terminal, navigate to the directory src and execute
the following command:

python predict_nb.py test_input/A_ASD0023_2_1N5M_1_desc.txt

the output file is written in the same directory as the input file.


nn.py
-------
An implementation of a neural network model.


rank_nn.py
-------
ranks a set of pockets based on their allostery
Uses ANN  model (nn.py)
Uses aplc.tsv
Usage: python rank_nn.py input_file.txt

example:
```

```
In your terminal, navigate to the directory src and execute
the following command:

python rank_nn.py test_input/AS091022202_3PJG_complex.txt

the output file is written in the same directory as the input file.
```

Bibliography

[1] K. Kageyama, Y. Onoyama, H. Kogawa, E. Goto, and K. Tanabe. "The maximum and minimum water content and cell volume of human erythrocytes in vitro". In: *Biophysical Chemistry* 34.1 (Sept. 15, 1989), pp. 79–82. ISSN: 0301-4622. DOI: 10.1016/0301-4622(89)80044-4. URL: http://www.sciencedirect.com/science/article/pii/0301462289800444.

[2] Ron Milo. "What is the total number of protein molecules per cell volume? A call to rethink some published values". In: *Bioessays* 35.12 (Dec. 2013), pp. 1050–1055. ISSN: 0265-9247. DOI: 10.1002/bies.201300066. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3910158/.

[3] David L. Nelson, Albert L. Lehninger, and Michael M. Cox. *Lehninger Principles of Biochemistry*. Google-Books-ID: 5Ek9J4p3NfkC. W. H. Freeman, Feb. 2008. 1303 pp. ISBN: 978-0-7167-7108-1.

[4] Robert Hill. *The Chemistry of Life: Eight Lectures on the History of Biochemistry*. Google-Books-ID: IvM8AAAAIAAJ. CUP Archive, 1970. 278 pp. ISBN: 978-0-521-07379-0.

[5] Ida Schomburg, Antje Chang, Sandra Placzek, Carola Söhngen, Michael Rother, Maren Lang, Cornelia Munaretto, Susanne Ulas, Michael Stelzer, Andreas Grote, Maurice Scheer, and Dietmar Schomburg. "BRENDA in 2013: integrated reactions, kinetic data, enzyme function data, improved disease classification: new options and contents in BRENDA". In: *Nucleic Acids Research* 41 (Database issue Jan. 2013), pp. D764–D772. ISSN: 0305-1048. DOI: 10.1093/nar/gks1049. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3531171/.

[6] Urs Feller, Iwona Anders, and Tadahiko Mae. "Rubiscolytics: fate of Rubisco after its enzymatic function in a cell is terminated". In: *Journal of Experimental Botany* 59.7 (May 1, 2008), pp. 1615–1624. ISSN: 0022-0957. DOI: 10.1093/jxb/erm242. URL: https://academic.oup.com/jxb/article/59/7/1615/638322 (visited on 11/21/2017).

[7] F. Grant Pearce. "Catalytic by-product formation and ligand binding by ribulose bisphosphate carboxylases from different phylogenies". In: *Biochemical Journal* 399 (Pt 3 Nov. 1, 2006), pp. 525–534. ISSN: 0264-6021. DOI: 10.1042/BJ20060430. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1615894/.

[8] R. John Ellis. "Biochemistry: Tackling unintelligent design". In: *Nature* 463.7278 (Jan. 13, 2010), 463164a. ISSN: 1476-4687. DOI: 10.1038/463164a. URL: https://www.nature.com/articles/463164a (visited on 11/21/2017).

[9] Robert J. Spreitzer and Michael E. Salvucci. "RUBISCO: Structure, Regulatory Interactions, and Possibilities for a Better Enzyme". In: *Annual Review of Plant Biology* 53.1 (2002), pp. 449–475. DOI: 10.1146/annurev.arplant.53.100301.135233. URL: https://doi.org/10.1146/annurev.arplant.53.100301.135233 (visited on 11/21/2017).

[10] Brian P. Callahan and Brian G. Miller. "OMP decarboxylase—An enigma persists". In: *Bioorganic Chemistry* 35.6 (Dec. 1, 2007), pp. 465–469. ISSN: 0045-2068. DOI: 10.1016/j.bioorg.2007.07.004. URL: http://www.sciencedirect.com/science/article/pii/S0045206807000508 (visited on 11/21/2017).

[11] Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. "General Principles of Cell Communication". In: (2002). URL: https://www.ncbi.nlm.nih.gov/books/NBK26813/ (visited on 11/27/2017).

[12] Richard Sever and Christopher K. Glass. "Signaling by Nuclear Receptors". In: *Cold Spring Harbor Perspectives in Biology* 5.3 (Mar. 2013). ISSN: 1943-0264. DOI: 10.1101/cshperspect.a016709. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3578364/ (visited on 11/27/2017).

[13] Thales Kronenberger, Oliver Keminer, Carsten Wrenger, and Björn Windshügel. "Nuclear Receptor Modulators — Current Approaches and Future Perspectives". In: (2015). DOI: 10.5772/59666. URL: http://www.intechopen.com/books/drug-discovery-and-development-from-molecules-to-medicine/nuclear-receptor-modulators-current-approaches-and-future-perspectives (visited on 11/28/2017).

[14] David J. Mangelsdorf, Carl Thummel, Miguel Beato, Peter Herrlich, Günther Schütz, Kazuhiko Umesono, Bruce Blumberg, Philippe Kastner, Manuel Mark, Pierre Chambon, and Ronald M. Evans. "The nuclear receptor superfamily: The second decade". In: *Cell* 83.6 (Dec. 15, 1995), pp. 835–839. ISSN: 0092-8674. DOI: 10.1016/0092-8674(95)90199-X. URL: http://www.sciencedirect.com/science/article/pii/009286749590199X (visited on 11/28/2017).

[15] Pablo C. Echeverria and Didier Picard. "Molecular chaperones, essential partners of steroid hormone receptors for activity and mobility". In: *Biochimica et Biophysica Acta (BBA) - Molecular Cell Research*. Molecular Chaperones and Intracellular Protein Transport 1803.6 (June 1, 2010), pp. 641–649. ISSN: 0167-4889. DOI: 10.1016/j.bbamcr.2009.11.012. URL: http://www.sciencedirect.com/science/article/pii/S0167488909002948 (visited on 11/28/2017).

[16] Christopher K. Glass and Michael G. Rosenfeld. "The coregulator exchange in transcriptional functions of nuclear receptors". In: *Genes & Development* 14.2 (Jan. 15, 2000), pp. 121–141. ISSN: 0890-9369, 1549-5477. DOI: 10.1101/gad.14.2.121. URL: http://genesdev.cshlp.org/content/14/2/121 (visited on 11/28/2017).

[17] J. Don Chen and Ronald M. Evans. "A transcriptional co-repressor that interacts with nuclear hormone receptors". In: *Nature* 377.6548 (Oct. 5, 1995), p. 454. ISSN: 1476-4687. DOI: 10.1038/377454a0. URL: https://www.nature.com/articles/377454a0 (visited on 11/28/2017).

[18] Christopher K. Glass, G. Bruce Wisely, Jeffery E. Cobb, Michael G. Rosenfeld, Michael V. Milburn, Millard H. Lambert, Riki Kurokawa, Robert T. Nolte, Stefan Westin, and Timothy M. Willson. "Ligand binding and co-activator assembly of the peroxisome proliferator-activated receptor". In: *Nature* 395.6698 (Sept. 10, 1998), p. 137. ISSN: 1476-4687. DOI: 10.1038/25931. URL: https://www.nature.com/articles/25931 (visited on 11/28/2017).

[19] Douglas B. Kitchen, Hélène Decornez, John R. Furr, and Jürgen Bajorath. "Docking and scoring in virtual screening for drug discovery: methods and applications". eng. In: *Nature Reviews. Drug Discovery* 3.11 (Nov. 2004), pp. 935–949. ISSN: 1474-1776. DOI: 10.1038/nrd1549.

[20] B. Lee and F. M. Richards. "The interpretation of protein structures: Estimation of static accessibility". In: *Journal of Molecular Biology* 55.3 (Feb. 1971), 379–IN4. ISSN: 0022-2836. DOI: 10.1016/

0022-2836(71)90324-X. URL: http://www.sciencedirect.com/science/article/pii/002228367190324X (visited on 05/08/2015).

[21] M. L. Connolly. "Solvent-accessible surfaces of proteins and nucleic acids". en. In: *Science* 221.4612 (Aug. 1983), pp. 709–713. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.6879170. URL: http://www.sciencemag.org/content/221/4612/709 (visited on 05/08/2015).

[22] R. Langridge, T. E. Ferrin, I. D. Kuntz, and M. L. Connolly. "Real-time color graphics in studies of molecular interactions". eng. In: *Science (New York, N.Y.)* 211.4483 (Feb. 1981), pp. 661–666. ISSN: 0036-8075.

[23] Chung-Jung Tsai, Buyong Ma, Yuk Yin Sham, Sandeep Kumar, and Ruth Nussinov. "Structured disorder and conformational selection". en. In: *Proteins: Structure, Function, and Bioinformatics* 44.4 (Sept. 2001), pp. 418–427. ISSN: 1097-0134. DOI: 10.1002/prot.1107. URL: http://onlinelibrary.wiley.com/doi/10.1002/prot.1107/abstract (visited on 05/08/2015).

[24] Irene Luque and Ernesto Freire. "Structural stability of binding sites: Consequences for binding affinity and allosteric effects". en. In: *Proteins: Structure, Function, and Bioinformatics* 41.S4 (Jan. 2000), pp. 63–71. ISSN: 1097-0134. DOI: 10.1002/1097-0134(2000)41:4+<63::AID-PROT60>3.0.CO;2-6. URL: http://onlinelibrary.wiley.com/doi/10.1002/1097-0134(2000)41:4+<63::AID-PROT60>3.0.CO;2-6/abstract (visited on 05/08/2015).

[25] D. E. Koshland. "Application of a Theory of Enzyme Specificity to Protein Synthesis". eng. In: *Proceedings of the National Academy of Sciences of the United States of America* 44.2 (Feb. 1958), pp. 98–104. ISSN: 0027-8424.

[26] P. Tuffery, C. Etchebest, S. Hazout, and R. Lavery. "A new approach to the rapid determination of protein side chain conformations". eng. In: *Journal of Biomolecular Structure & Dynamics* 8.6 (June 1991), pp. 1267–1289. ISSN: 0739-1102. DOI: 10.1080/07391102.1991.10507882.

[27] D. M. Lorber and B. K. Shoichet. "Flexible ligand docking using conformational ensembles." In: *Protein Science : A Publication of the Protein Society* 7.4 (Apr. 1998), pp. 938–950. ISSN: 0961-8368. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2143983/ (visited on 05/08/2015).

[28] David S. Goodsell and Arthur J. Olson. "Automated docking of substrates to proteins by simulated annealing". en. In: *Proteins: Structure, Function, and Bioinformatics* 8.3 (Jan. 1990), pp. 195–202. ISSN: 1097-0134. DOI: 10.1002/prot.340080302. URL: http://onlinelibrary.wiley.com/doi/10.1002/prot.340080302/abstract (visited on 05/08/2015).

[29] R. L. DesJarlais, R. P. Sheridan, J. S. Dixon, I. D. Kuntz, and R. Venkataraghavan. "Docking flexible ligands to macromolecular receptors by molecular shape". eng. In: *Journal of Medicinal Chemistry* 29.11 (Nov. 1986), pp. 2149–2153. ISSN: 0022-2623.

[30] Garrett M. Morris, David S. Goodsell, Robert S. Halliday, Ruth Huey, William E. Hart, Richard K. Belew, and Arthur J. Olson. "Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function". en. In: *Journal of Computational Chemistry* 19.14 (Nov. 1998), pp. 1639–1662. ISSN: 1096-987X. DOI: 10.1002/(SICI)1096-987X(19981115)19:14<1639::AID-JCC10>3.0.CO;2-B. URL: http://onlinelibrary.wiley.com/doi/10.1002/(SICI)1096-987X(19981115)19:14<1639::AID-JCC10>3.0.CO;2-B/abstract (visited on 05/08/2015).

[31] L.F. Ten Eyck, J. Mandell, V.A. Roberts, and M.E. Pique. "Surveying Molecular Interactions with DOT". In: *Supercomputing, 1995. Proceedings of the IEEE/ACM SC95 Conference*. 1995, pp. 22–22. DOI: 10.1109/SUPERC.1995.242670.

[32] P. Nuno Palma, Ludwig Krippahl, John E. Wampler, and José J.G. Moura. "BiGGER: A new (soft) docking algorithm for predicting protein interactions". en. In: *Proteins: Structure, Function, and Bioinformatics* 39.4 (June 2000), pp. 372–384. ISSN: 1097-0134. DOI: 10.1002/(SICI)1097-0134(20000601)39:4<372::AID-PROT100>3.0.CO;2-Q. URL: http://onlinelibrary.wiley.com/doi/10.1002/(SICI)1097-0134(20000601)39:4<372::AID-PROT100>3.0.CO;2-Q/abstract (visited on 05/09/2015).

[33] Peter. Kollman. "Free energy calculations: Applications to chemical and biochemical phenomena". In: *Chemical Reviews* 93.7 (Nov. 1993), pp. 2395–2417. ISSN: 0009-2665. DOI: 10.1021/cr00023a004. URL: http://dx.doi.org/10.1021/cr00023a004 (visited on 05/09/2015).

[34] Natasja Brooijmans and Irwin D. Kuntz. "Molecular Recognition and Docking Algorithms". In: *Annual Review of Biophysics and Biomolecular Structure* 32.1 (2003), pp. 335–373. DOI: 10.1146/annurev.

biophys.32.110601.142532. URL: http://dx.doi.org/10.1146/annurev.biophys.32.110601.142532 (visited on 05/09/2015).

[35] Thomas Simonson, Georgios Archontis, and Martin Karplus. "Free Energy Simulations Come of Age ProteinLigand Recognition". In: *Accounts of Chemical Research* 35.6 (June 2002), pp. 430–437. ISSN: 0001-4842. DOI: 10.1021/ar010030m. URL: http://dx.doi.org/10.1021/ar010030m (visited on 05/09/2015).

[36] Elizabeth Yuriev, Jessica Holien, and Paul A. Ramsland. "Improvements, trends, and new ideas in molecular docking: 2012–2013 in review". en. In: *Journal of Molecular Recognition* (Mar. 2015), n/a–n/a. ISSN: 1099-1352. DOI: 10.1002/jmr.2471. URL: http://onlinelibrary.wiley.com/doi/10.1002/jmr.2471/abstract (visited on 05/10/2015).

[37] Zhe Zhang and Oliver F. Lange. "Replica Exchange Improves Sampling in Low-Resolution Docking Stage of RosettaDock". In: *PLoS ONE* 8.8 (Aug. 2013), e72096. DOI: 10.1371/journal.pone.0072096. URL: http://dx.doi.org/10.1371/journal.pone.0072096 (visited on 05/10/2015).

[38] Artem B. Mamonov, Steven Lettieri, Ying Ding, Jessica L. Sarver, Rohith Palli, Timothy F. Cunningham, Sunil Saxena, and Daniel M. Zuckerman. "Tunable, Mixed-Resolution Modeling Using Library-Based Monte Carlo and Graphics Processing Units". In: *Journal of Chemical Theory and Computation* 8.8 (Aug. 2012), pp. 2921–2929. ISSN: 1549-9618. DOI: 10.1021/ct300263z. URL: http://dx.doi.org/10.1021/ct300263z (visited on 05/10/2015).

[39] N. J. Gumede, P. Singh, M. I. Sabela, K. Bisetty, L. Escuder-Gilabert, M. J. Medina-Hernández, and S. Sagrado. "Experimental-Like Affinity Constants and Enantioselectivity Estimates from Flexible Docking". In: *Journal of Chemical Information and Modeling* 52.10 (Oct. 2012), pp. 2754–2759. ISSN: 1549-9596. DOI: 10.1021/ci300335m. URL: http://dx.doi.org/10.1021/ci300335m (visited on 05/10/2015).

[40] Martin Smieško. "DOLINA – Docking Based on a Local Induced-Fit Algorithm: Application toward Small-Molecule Binding to Nuclear Receptors". In: *Journal of Chemical Information and Modeling* 53.6 (June 2013), pp. 1415–1423. ISSN: 1549-9596. DOI: 10.1021/ci400098y. URL: http://dx.doi.org/10.1021/ci400098y (visited on 05/10/2015).

[41] Marcel Schumann and Roger S. Armen. "Systematic and efficient side chain optimization for molecular docking using a cheapest-path procedure". en. In: *Journal of Computational Chemistry* 34.14 (May 2013), pp. 1258–1269. ISSN: 1096-987X. DOI: 10.1002/jcc.23251. URL: http://onlinelibrary.wiley.com/doi/10.1002/jcc.23251/abstract (visited on 05/10/2015).

[42] Johannes Flick, Frank Tristram, and Wolfgang Wenzel. "Modeling loop backbone flexibility in receptor-ligand docking simulations". en. In: *Journal of Computational Chemistry* 33.31 (Dec. 2012), pp. 2504–2515. ISSN: 1096-987X. DOI: 10.1002/jcc.23087. URL: http://onlinelibrary.wiley.com/doi/10.1002/jcc.23087/abstract (visited on 05/10/2015).

[43] Daniel Mucs and Richard A Bryce. "The application of quantum mechanics in structure-based drug design". In: *Expert Opinion on Drug Discovery* 8.3 (Jan. 2013), pp. 263–276. ISSN: 1746-0441. DOI: 10.1517/17460441.2013.752812. URL: http://informahealthcare.com/doi/abs/10.1517/17460441.2013.752812 (visited on 05/11/2015).

[44] J. Andrew McCammon, Bruce R. Gelin, and Martin Karplus. "Dynamics of folded proteins". en. In: *Nature* 267.5612 (June 1977), pp. 585–590. DOI: 10.1038/267585a0. URL: http://www.nature.com/nature/journal/v267/n5612/abs/267585a0.html (visited on 05/12/2015).

[45] David E. Shaw, Ron O. Dror, John K. Salmon, J.P. Grossman, Kenneth M. Mackenzie, Joseph A. Bank, Cliff Young, Martin M. Deneroff, Brannon Batson, Kevin J. Bowers, Edmond Chow, Michael P. Eastwood, Douglas J. Ierardi, John L. Klepeis, Jeffrey S. Kuskin, Richard H. Larson, Kresten Lindorff-Larsen, Paul Maragakis, Mark A. Moraes, Stefano Piana, Yibing Shan, and Brian Towles. "Millisecond-scale molecular dynamics simulations on Anton". In: *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*. Nov. 2009, pp. 1–11. DOI: 10.1145/1654059.1654099.

[46] Jaroaw Meller. "Molecular Dynamics". en. In: *eLS*. John Wiley & Sons, Ltd, 2001. ISBN: 9780470015902. URL: http://onlinelibrary.wiley.com/doi/10.1038/npg.els.0003048/abstract (visited on 05/12/2015).

[47] Roman Petrenko and Jarosław Meller. "Molecular Dynamics". en. In: *eLS*. John Wiley & Sons, Ltd, 2001. ISBN: 9780470015902. URL: http://onlinelibrary.wiley.com/doi/10.1002/9780470015902.a0003048.pub2/abstract (visited on 05/12/2015).

[48] William L. Jorgensen, David S. Maxwell, and Julian Tirado-Rives. "Development and Testing of the OPLS All-Atom Force Field on Conformational Energetics and Properties of Organic Liquids". In: *Journal of the American Chemical Society* 118.45 (Jan. 1996), pp. 11225–11236. ISSN: 0002-7863. DOI: 10.1021/ja9621760. URL: http://dx.doi.org/10.1021/ja9621760 (visited on 05/13/2015).

[49] William L. Jorgensen and Julian. Tirado-Rives. "The OPLS [optimized potentials for liquid simulations] potential functions for proteins, energy minimizations for crystals of cyclic peptides and crambin". In: *Journal of the American Chemical Society* 110.6 (Mar. 1988), pp. 1657–1666. ISSN: 0002-7863. DOI: 10.1021/ja00214a001. URL: http://dx.doi.org/10.1021/ja00214a001 (visited on 05/13/2015).

[50] "Comparison of Protein Force Fields for Molecular Dynamics Simulations - Springer". In: ed. by Andreas Kukol. Methods Molecular Biology™ 443. Humana Press, 2008. ISBN: 978-1-58829-864-5, 978-1-59745-177-2. URL: http://link.springer.com/protocol/10.1007%2F978-1-59745-177-2_4 (visited on 05/13/2015).

[51] Elio A. Cino, Wing-Yiu Choy, and Mikko Karttunen. "Comparison of Secondary Structure Formation Using 10 Different Force Fields in Microsecond Molecular Dynamics Simulations". In: *Journal of Chemical Theory and Computation* 8.8 (Aug. 2012), pp. 2725–2740. ISSN: 1549-9618. DOI: 10.1021/ct300323g. URL: http://dx.doi.org/10.1021/ct300323g (visited on 05/13/2015).

[52] Sander Pronk, Szilárd Páll, Roland Schulz, Per Larsson, Pär Bjelkmar, Rossen Apostolov, Michael R. Shirts, Jeremy C. Smith, Peter M. Kasson, David van der Spoel, Berk Hess, and Erik Lindahl. "GROMACS 4.5: a high-throughput and highly parallel open source molecular simulation toolkit". en. In: *Bioinformatics* 29.7 (Apr. 2013), pp. 845–854. ISSN: 1367-4803, 1460-2059. DOI: 10.1093/bioinformatics/btt055. URL: http://bioinformatics.oxfordjournals.org/content/29/7/845 (visited on 05/14/2015).

[53] A. L. Samuel. "Some Studies in Machine Learning Using the Game of Checkers". In: *IBM Journal of Research and Development* 3.3 (July 1959), pp. 210–229. ISSN: 0018-8646. DOI: 10.1147/rd.33.0210.

[54] Marc Wieland and Massimiliano Pittore. "Performance Evaluation of Machine Learning Algorithms for Urban Pattern Recognition from Multi-spectral Satellite Images". In: *Remote Sensing* 6.4 (Mar. 31, 2014), pp. 2912–2939. DOI: 10.3390/rs6042912. URL: http://www.mdpi.com/2072-4292/6/4/2912 (visited on 11/29/2017).

[55] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. "Thumbs up? Sentiment Classification using Machine Learning Techniques". In: *arXiv:cs/0205070* (May 27, 2002). arXiv: cs/0205070. URL: http://arxiv.org/abs/cs/0205070 (visited on 11/29/2017).

[56] R. Burbidge, M. Trotter, B. Buxton, and S. Holden. "Drug design by machine learning: support vector machines for pharmaceutical data analysis". In: *Computers & Chemistry* 26.1 (Dec. 1, 2001), pp. 5–14. ISSN: 0097-8485. DOI: 10.1016/S0097-8485(01)00094-8. URL: http://www.sciencedirect.com/science/article/pii/S0097848501000948 (visited on 11/29/2017).

[57] Thomas Bayes. "LII. An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, F. R. S. communicated by Mr. Price, in a letter to John Canton, A. M. F. R. S". In: *Philosophical Transactions* 53 (Jan. 1, 1763), pp. 370–418. ISSN: 0261-0523, DOI: 10.1098/rstl.1763.0053. URL: http://rstl.royalsocietypublishing.org/content/53/370 (visited on 11/29/2017).

[58] Gail L. Rosen, Erin R. Reichenberger, and Aaron M. Rosenfeld. "NBC: the Naïve Bayes Classification tool webserver for taxonomic classification of metagenomic reads". In: *Bioinformatics* 27.1 (Jan. 1, 2011), pp. 127–129. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btq619. URL: https://academic.oup.com/bioinformatics/article/27/1/127/202209 (visited on 11/29/2017).

[59] Warren S. McCulloch and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity". en. In: *The bulletin of mathematical biophysics* 5.4 (Dec. 1943), pp. 115–133. ISSN: 0007-4985, 1522-9602. DOI: 10.1007/BF02478259. URL: http://link.springer.com/article/10.1007/BF02478259 (visited on 11/17/2015).

[60] F. Rosenblatt. "The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain". In: *Psychological Review* (1958), pp. 65–386.

[61] Angela Guerra, Pedro Gonzalez-Naranjo, Nuria E. Campillo, Javier Varela, María L. Lavaggi, Alicia Merlino, Hugo Cerecetto, Mercedes González, Alicia Gomez-Barrio, José A. Escario, Cristina Fonseca-Berzal, Gloria Yaluf, Jorge Paniagua-Solis, and Juan A. Páez. "Novel Imidazo[4,5-c][1,2,6]thiadiazine 2,2-dioxides as antiproliferative trypanosoma cruzi drugs: Computational screening from neural network, synthesis and in vivo biological properties". In: *European Journal of Medicinal Chemistry* 136 (Supplement C Aug. 18, 2017), pp. 223–234. ISSN: 0223-5234. DOI: 10.1016/j.ejmech.2017.04.075.

URL: http://www.sciencedirect.com/science/article/pii/S0223523417303574 (visited on 12/04/2017).

[62] J. Wehrmann, W. Becker, H. E. L. Cagnini, and R. C. Barros. "A character-based convolutional neural network for language-agnostic Twitter sentiment analysis". In: *2017 International Joint Conference on Neural Networks (IJCNN)*. 2017 International Joint Conference on Neural Networks (IJCNN). May 2017, pp. 2384–2391. DOI: 10.1109/IJCNN.2017.7966145.

[63] Mariusz Bojarski, Philip Yeres, Anna Choromanska, Krzysztof Choromanski, Bernhard Firner, Lawrence Jackel, and Urs Muller. "Explaining How a Deep Neural Network Trained with End-to-End Learning Steers a Car". In: *arXiv:1704.07911 [cs]* (Apr. 25, 2017). arXiv: 1704.07911. URL: http://arxiv.org/abs/1704.07911 (visited on 12/04/2017).

[64] Lior Rokach and Oded Maimon. "Clustering Methods". In: *Data Mining and Knowledge Discovery Handbook*. DOI: 10.1007/0-387-25465-X_15. Springer, Boston, MA, 2005, pp. 321–352. ISBN: 978-0-387-24435-8 978-0-387-25465-4. URL: https://link.springer.com/chapter/10.1007/0-387-25465-X_15 (visited on 12/04/2017).

[65] *Hierarchical cluster analysis on famous data sets - enhanced with the dendextend package*. URL: https://cran.r-project.org/web/packages/dendextend/vignettes/Cluster_Analysis.html (visited on 12/04/2017).

[66] Mai Kasai, Yuya Yasuda, Hiroshi Mizoguchi, Kohei Soga, Kazuhiro Kaneko, and Hiroshi Takemura. "In vivo tumor wavelength band selection using Hierarchical clustering and PCA with NIR-Hyperspectral Data". In: *Journal of Biomedical Engineering and Medical Imaging* 4.1 (Mar. 1, 2017), p. 01. DOI: 10.14738/jbemi.41.2799. URL: http://scholarpublishing.org/index.php/JBEMi/article/view/2799 (visited on 12/04/2017).

[67] Cele Abad-Zapatero. "Notes of a protein crystallographer: on the high-resolution structure of the PDB growth rate". In: *Acta Crystallographica Section D Biological Crystallography* 68.5 (May 2012), pp. 613–617. ISSN: 0907-4449. DOI: 10.1107/S0907444912004799. URL: http://scripts.iucr.org/cgi-bin/paper?S0907444912004799 (visited on 11/16/2015).

[68] Susanne Eyrisch and Volkhard Helms. "Transient pockets on protein surfaces involved in protein-protein interaction". eng. In: *Journal of Medicinal Chemistry* 50.15 (July 2007), pp. 3457–3464. ISSN: 0022-2623. DOI: 10.1021/jm070095g.

[69] Jesus Seco, F. Javier Luque, and Xavier Barril. "Binding Site Detection and Druggability Index from First Principles". In: *Journal of Medicinal Chemistry* 52.8 (Apr. 2009), pp. 2363–2371. ISSN: 0022-2623. DOI: 10.1021/jm801385d. URL: http://dx.doi.org/10.1021/jm801385d (visited on 07/23/2016).

[70] Sara E. Nichols, Riccardo Baron, Anthony Ivetac, and J. Andrew McCammon. "Predictive Power of Molecular Dynamics Receptor Structures in Virtual Screening". In: *Journal of Chemical Information and Modeling* 51.6 (June 2011), pp. 1439–1446. ISSN: 1549-9596. DOI: 10.1021/ci200117n. URL: http://dx.doi.org/10.1021/ci200117n (visited on 11/12/2015).

[71] Antonia Stank, Daria B Kokh, Jonathan C Fuller, and Rebecca C Wade. "Protein Binding Pocket Dynamics". In: *Accounts of chemical research* 49.5 (2016), pp. 809–815.

[72] James L. Melville, Edmund K. Burke, and Jonathan D. Hirst. "Machine learning in virtual screening". eng. In: *Combinatorial Chemistry & High Throughput Screening* 12.4 (May 2009), pp. 332–343. ISSN: 1875-5402.

[73] Corinna Cortes and Vladimir Vapnik. "Support-Vector Networks". en. In: *Machine Learning* 20.3 (Sept. 1995), pp. 273–297. ISSN: 0885-6125, 1573-0565. DOI: 10.1023/A:1022627411411. URL: http://link.springer.com/article/10.1023/A%3A1022627411411 (visited on 11/17/2015).

[74] Leo Breiman. *Classification and regression trees*. Wadsworth International Group, 1984. ISBN: 0-412-04841-8.

[75] David J. Hand and Keming Yu. "Idiot's Bayes: Not So Stupid after All?" In: *International Statistical Review / Revue Internationale de Statistique* 69.3 (2001), pp. 385–398. ISSN: 0306-7734. DOI: 10.2307/1403452. URL: http://www.jstor.org/stable/1403452 (visited on 11/17/2015).

[76] Evan E. Bolton, Yanli Wang, Paul A. Thiessen, and Stephen H. Bryant. "Chapter 12 - PubChem: Integrated Platform of Small Molecules and Biological Activities". In: *Annual Reports in Computational Chemistry*. Ed. by Ralph A. Wheeler and David C. Spellmeyer. Vol. 4.

Elsevier, 2008, pp. 217–241. URL: http://www.sciencedirect.com/science/article/pii/S1574140008000121 (visited on 11/16/2015).

[77] Bee Wah Yap, Khatijahhusna Abd Rani, Hezlin Aryani Abd Rahman, Simon Fong, Zuraida Khairudin, and Nik Nik Abdullah. "An Application of Oversampling, Undersampling, Bagging and Boosting in Handling Imbalanced Datasets". en. In: *Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013)*. Ed. by Tutut Herawan, Mustafa Mat Deris, and Jemal Abawajy. Lecture Notes in Electrical Engineering 285. DOI: 10.1007/978-981-4585-18-7_2. Springer Singapore, 2014, pp. 13–22. ISBN: 978-981-4585-17-0 978-981-4585-18-7. URL: http://link.springer.com/chapter/10.1007/978-981-4585-18-7_2 (visited on 07/23/2016).

[78] Haibo He and E.A. Garcia. "Learning from Imbalanced Data". In: *IEEE Transactions on Knowledge and Data Engineering* 21.9 (Sept. 2009), pp. 1263–1284. ISSN: 1041-4347. DOI: 10.1109/TKDE.2008.239.

[79] Andrea Volkamer, Daniel Kuhn, Friedrich Rippmann, and Matthias Rarey. "DoGSiteScorer: a web server for automatic binding site prediction, analysis and druggability assessment". en. In: *Bioinformatics* 28.15 (Aug. 2012), pp. 2074–2075. ISSN: 1367-4803, 1460-2059. DOI: 10.1093/bioinformatics/bts310. URL: http://bioinformatics.oxfordjournals.org/content/28/15/2074 (visited on 11/18/2015).

[80] *Maestro*. 2015.

[81] Michael M Mysinger, Michael Carchia, John J Irwin, and Brian K Shoichet. "Directory of useful decoys, enhanced (DUD-E): better ligands and decoys for better benchmarking". In: *Journal of medicinal chemistry* 55.14 (2012), pp. 6582–6594.

[82] Haibo He, Yang Bai, E.A. Garcia, and Shutao Li. "ADASYN: Adaptive synthetic sampling approach for imbalanced learning". In: *IEEE International Joint Conference on Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence)*. June 2008, pp. 1322–1328. DOI: 10.1109/IJCNN.2008.4633969.

[83] Andrea Volkamer, Axel Griewel, Thomas Grombacher, and Matthias Rarey. "Analyzing the Topology of Active Sites: On the Prediction of Pockets and Subpockets". In: *Journal of Chemical Information and Modeling* 50.11 (Nov. 2010), pp. 2041–2052. ISSN: 1549-9596. DOI: 10.1021/ci100241y. URL: http://dx.doi.org/10.1021/ci100241y (visited on 11/19/2015).

[84] Andrea Volkamer, Daniel Kuhn, Thomas Grombacher, Friedrich Rippmann, and Matthias Rarey. "Combining Global and Local Measures for Structure-Based Druggability Predictions". In: *Journal of Chemical Information and Modeling* 52.2 (Feb. 2012), pp. 360–372. ISSN: 1549-9596. DOI: 10.1021/ci200454v. URL: http://dx.doi.org/10.1021/ci200454v (visited on 11/19/2015).

[85] Alan C. Cheng, Ryan G. Coleman, Kathleen T. Smyth, Qing Cao, Patricia Soulard, Daniel R. Caffrey, Anna C. Salzberg, and Enoch S. Huang. "Structure-based maximal affinity model predicts small-molecule druggability". en. In: *Nature Biotechnology* 25.1 (Jan. 2007), pp. 71–75. ISSN: 1087-0156. DOI: 10.1038/nbt1273. URL: http://www.nature.com/nbt/journal/v25/n1/full/nbt1273.html (visited on 02/11/2016).

[86] Thomas A. Halgren. "Identifying and Characterizing Binding Sites and Assessing Druggability". In: *Journal of Chemical Information and Modeling* 49.2 (Feb. 2009), pp. 377–389. ISSN: 1549-9596. DOI: 10.1021/ci800324m. URL: http://dx.doi.org/10.1021/ci800324m (visited on 02/11/2016).

[87] Robert P. Sheridan, Vladimir N. Maiorov, M. Katharine Holloway, Wendy D. Cornell, and Ying-Duo Gao. "Drug-like Density: A Method of Quantifying the "Bindability" of a Protein Target Based on a Very Large Set of Pockets and Drug-like Ligands from the Protein Data Bank". In: *Journal of Chemical Information and Modeling* 50.11 (Nov. 2010), pp. 2029–2040. ISSN: 1549-9596. DOI: 10.1021/ci100312t. URL: http://dx.doi.org/10.1021/ci100312t (visited on 02/11/2016).

[88] Peter Schmidtke and Xavier Barril. "Understanding and Predicting Druggability. A High-Throughput Method for Detection of Drug Binding Sites". In: *Journal of Medicinal Chemistry* 53.15 (Aug. 2010), pp. 5858–5867. ISSN: 0022-2623. DOI: 10.1021/jm100574m. URL: http://dx.doi.org/10.1021/jm100574m (visited on 02/11/2016).

[89] Guido Rossum. *Python Reference Manual*. Tech. rep. Amsterdam, The Netherlands, The Netherlands: CWI (Centre for Mathematics and Computer Science), 1995.

[90] H. J. C. Berendsen, D. van der Spoel, and R. van Drunen. "GROMACS: A message-passing parallel molecular dynamics implementation". In: *Computer Physics Communications* 91.1–3 (Sept. 1995), pp. 43–56. ISSN: 0010-4655. DOI: 10.1016/0010-4655(95)00042-E. URL: http://www.sciencedirect.com/science/article/pii/001046559500042E (visited on 12/03/2015).

[91] Jacob D. Durrant, Lane Votapka, Jesper Sørensen, and Rommie E. Amaro. "POVME 2.0: An Enhanced Tool for Determining Pocket Shape and Volume Characteristics". In: *Journal of Chemical Theory and Computation* 10.11 (Nov. 2014), pp. 5047–5056. ISSN: 1549-9618. DOI: 10.1021/ct500381c. URL: http://dx.doi.org/10.1021/ct500381c (visited on 12/03/2015).

[92] X. Daura, W. F. van Gunsteren, and A. E. Mark. "Folding-unfolding thermodynamics of a beta-heptapeptide from equilibrium simulations". eng. In: *Proteins* 34.3 (Feb. 1999), pp. 269–280. ISSN: 0887-3585.

[93] G. Madhavi Sastry, Steven L. Dixon, and Woody Sherman. "Rapid Shape-Based Ligand Alignment and Virtual Screening Method Based on Atom/Feature-Pair Similarities and Volume Overlap Scoring". In: *Journal of Chemical Information and Modeling* 51.10 (Oct. 2011), pp. 2455–2466. ISSN: 1549-9596. DOI: 10.1021/ci2002704. URL: http://dx.doi.org/10.1021/ci2002704 (visited on 12/03/2015).

[94] K. Gunasekaran, Buyong Ma, and Ruth Nussinov. "Is allostery an intrinsic property of all dynamic proteins?" In: *Proteins: Structure, Function, and Bioinformatics* 57.3 (Nov. 15, 2004), pp. 433–443. ISSN: 1097-0134. DOI: 10.1002/prot.20232. URL: http://onlinelibrary.wiley.com/doi/10.1002/prot.20232/abstract (visited on 11/08/2016).

[95] Qiancheng Shen, Guanqiao Wang, Shuai Li, Xinyi Liu, Shaoyong Lu, Zhongjie Chen, Kun Song, Junhao Yan, Lv Geng, Zhimin Huang, Wenkang Huang, Guoqiang Chen, and Jian Zhang. "ASD v3.0: unraveling allosteric regulation with structural mechanisms and biological networks". In: *Nucleic Acids Research* 44 (D1 Jan. 4, 2016), pp. D527–D535. ISSN: 0305-1048, 1362-4962. DOI: 10.1093/nar/gkv902. URL: http://nar.oxfordjournals.org/content/44/D1/D527 (visited on 11/07/2016).

[96] Rachael P. Huntley, Tony Sawford, Prudence Mutowo-Meullenet, Aleksandra Shypitsyna, Carlos Bonilla, Maria J. Martin, and Claire O'Donovan. "The GOA database: gene Ontology annotation updates for 2015". In: *Nucleic Acids Research* 43 (Database issue Jan. 2015), pp. D1057–1063. ISSN: 1362-4962. DOI: 10.1093/nar/gku1113.

[97] Mathias Rask-Andersen, Markus Sällman Almén, and Helgi B. Schiöth. "Trends in the exploitation of novel drug targets". In: *Nature Reviews Drug Discovery* 10.8 (Aug. 2011), pp. 579–590. ISSN: 1474-1776. DOI: 10.1038/nrd3478. URL: http://www.nature.com/nrd/journal/v10/n8/full/nrd3478.html (visited on 11/07/2016).

[98] Aron W. Fenton. "Allostery: an illustrated definition for the 'second secret of life'". In: *Trends in biochemical sciences* 33.9 (Sept. 2008), pp. 420–425. ISSN: 0968-0004. DOI: 10.1016/j.tibs.2008.05.009. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2574622/ (visited on 11/07/2016).

[99] Laurent-Philippe Albou, Benjamin Schwarz, Olivier Poch, Jean Marie Wurtz, and Dino Moras. "Defining and characterizing protein surface using alpha shapes". In: *Proteins* 76.1 (July 2009), pp. 1–12. ISSN: 1097-0134. DOI: 10.1002/prot.22301.

[100] Teresa Paramo, Alexandra East, Diana Garzón, Martin B. Ulmschneider, and Peter J. Bond. "Efficient Characterization of Protein Cavities within Molecular Simulation Trajectories: trj_cavity". In: *Journal of Chemical Theory and Computation* 10.5 (May 13, 2014), pp. 2151–2164. ISSN: 1549-9618. DOI: 10.1021/ct401098b. URL: http://dx.doi.org/10.1021/ct401098b (visited on 11/07/2016).

[101] E. F. Nemeth, M. E. Steffey, L. G. Hammerland, B. C. Hung, B. C. Van Wagenen, E. G. DelMar, and M. F. Balandrin. "Calcimimetics with potent and selective activity on the parathyroid calcium receptor". In: *Proceedings of the National Academy of Sciences of the United States of America* 95.7 (Mar. 31, 1998), pp. 4040–4045. ISSN: 0027-8424.

[102] Wenkang Huang, Guanqiao Wang, Qiancheng Shen, Xinyi Liu, Shaoyong Lu, Lv Geng, Zhimin Huang, and Jian Zhang. "ASBench: benchmarking sets for allosteric discovery". In: *Bioinformatics (Oxford, England)* 31.15 (Aug. 1, 2015), pp. 2598–2600. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btv169.

[103] Corinna Cortes and Vladimir Vapnik. "Support-Vector Networks". In: *Machine Learning* 20.3 (), pp. 273–297. ISSN: 0885-6125, 1573-0565. DOI: 10.1023/A:1022627411411. URL: http://link.springer.com/article/10.1023/A:1022627411411 (visited on 11/08/2016).

[104] Warren S. McCulloch and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity". In: *The bulletin of mathematical biophysics* 5.4 (), pp. 115–133. ISSN: 0007-4985, 1522-9602. DOI: 10.1007/BF02478259. URL: http://link.springer.com/article/10.1007/BF02478259 (visited on 11/08/2016).

[105] Joe G. Greener and Michael J. E. Sternberg. "AlloPred: prediction of allosteric pockets on proteins using normal mode perturbation analysis". In: *BMC bioinformatics* 16 (Oct. 23, 2015), p. 335. ISSN: 1471-2105. DOI: 10.1186/s12859-015-0771-1.

[106] Wenkang Huang, Shaoyong Lu, Zhimin Huang, Xinyi Liu, Linkai Mou, Yu Luo, Yanlong Zhao, Yaqin Liu, Zhongjie Chen, Tingjun Hou, and Jian Zhang. "Allosite: a method for predicting allosteric sites". In: *Bioinformatics (Oxford, England)* 29.18 (Sept. 15, 2013), pp. 2357–2359. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btt399.

[107] Alejandro Panjkovich and Xavier Daura. "PARS: a web server for the prediction of Protein Allosteric and Regulatory Sites". In: *Bioinformatics (Oxford, England)* 30.9 (May 1, 2014), pp. 1314–1315. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btu002.

[108] Ji Guo Su, Li Sheng Qi, Chun Hua Li, Yan Ying Zhu, Hui Jing Du, Yan Xue Hou, Rui Hao, and Ji Hua Wang. "Prediction of allosteric sites on protein surfaces with an elastic-network-model-based thermodynamic method". In: *Physical Review. E, Statistical, Nonlinear, and Soft Matter Physics* 90.2 (Aug. 2014), p. 022719. ISSN: 1550-2376. DOI: 10.1103/PhysRevE.90.022719.

[109] *Pattern Recognition - An Algorithmic Approach | M. Narasimha Murty | Springer*. URL: //www.springer.com/de/book/9780857294944 (visited on 11/05/2017).

[110] Eric Jones, Travis Oliphant, and Pearu Peterson. *SciPy: Open Source Scientific Tools for Python*. URL: http://www.scipy.org/.

[111] *An Introduction to Statistical Learning - with Applications in R | Gareth James | Springer*. URL: //www.springer.com/de/book/9781461471370 (visited on 11/05/2017).

[112] Buyong Ma and Ruth Nussinov. "Druggable orthosteric and allosteric hot spots to target protein-protein interactions". In: *Current Pharmaceutical Design* 20.8 (2014), pp. 1293–1301. ISSN: 1873-4286.

[113] Vincent Le Guilloux, Peter Schmidtke, and Pierre Tuffery. "Fpocket: An open source platform for ligand pocket detection". In: *BMC Bioinformatics* 10 (2009), p. 168. ISSN: 1471-2105. DOI: 10.1186/1471-2105-10-168. URL: http://dx.doi.org/10.1186/1471-2105-10-168 (visited on 11/30/2016).

[114] José Ramón López-Blanco, Osamu Miyashita, Florence Tama, and Pablo Chacón. "Normal Mode Analysis Techniques in Structural Biology". In: *eLS*. John Wiley & Sons, Ltd, 2001. ISBN: 978-0-470-01590-2. URL: http://onlinelibrary.wiley.com/doi/10.1002/9780470015902.a0020204.pub2/abstract (visited on 11/30/2016).

[115] Rommie E. Amaro, Riccardo Baron, and J. Andrew McCammon. "An improved relaxed complex scheme for receptor flexibility in computer-aided drug design". In: *Journal of Computer-Aided Molecular Design* 22.9 (Sept. 2008), pp. 693–705. ISSN: 0920-654X. DOI: 10.1007/s10822-007-9159-2. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2516539/ (visited on 05/23/2017).

[116] Rahmad Akbar, Siti Azma Jusoh, Rommie E. Amaro, and Volkhard Helms. "ENRI: A tool for selecting structure-based virtual screening target conformations". In: *Chemical Biology & Drug Design* 89.5 (May 1, 2017), pp. 762–771. ISSN: 1747-0285. DOI: 10.1111/cbdd.12900. URL: http://onlinelibrary.wiley.com/doi/10.1111/cbdd.12900/abstract (visited on 05/23/2017).

[117] S. Ringer. "A further Contribution regarding the influence of the different Constituents of the Blood on the Contraction of the Heart". eng. In: *J. Physiol. (Lond.)* 4.1 (Jan. 1883), pp. 29–42.3. ISSN: 0022-3751.

[118] I. de B. Daly and A. J. Clark. "The action of ions upon the frog's heart". eng. In: *J. Physiol. (Lond.)* 54.5-6 (Mar. 1921), pp. 367–383. ISSN: 0022-3751.

[119] John G. McCarron, John V. Walsh, and Fredric S. Fay. "Sodium/calcium exchange regulates cytoplasmic calcium in smooth muscle". en. In: *Pflügers Arch.* 426.3-4 (Feb. 1994), pp. 199–205. ISSN: 0031-6768, 1432-2013. DOI: 10.1007/BF00374772. URL: https://link.springer.com/article/10.1007/BF00374772 (visited on 04/12/2017).

[120] G. Beauchamp, P. A. Lavoie, and R. Elie. "Effect of trimipramine on depolarization-induced and Na+-Ca2+ exchange-induced 45calcium uptake in synaptosomes from the cortex of the rat brain". In: *Neuropharmacology* 31.3 (Mar. 1992), pp. 229–234. ISSN: 0028-3908. DOI: 10.1016/0028-3908(92)90172-L. URL: http://www.sciencedirect.com/science/article/pii/002839089290172L (visited on 04/12/2017).

[121] M. Taglialatela, S. Amoroso, L. M. Canzoniero, G. F. Di Renzo, and L. Annunziato. "Membrane events and ionic processes involved in dopamine release from tuberoinfundibular neurons. II. Effect of the inhibition of the Na+-Ca++ exchange by amiloride". eng. In: *J. Pharmacol. Exp. Ther.* 246.2 (Aug. 1988), pp. 689–694. ISSN: 0022-3565.

[122] Kazuhiro Takuma, Toshio Matsuda, Hitoshi Hashimoto, Junichi Kitanaka, Shoichi Asano, Yoko Kishida, and Akemichi Baba. "Role of Na+-Ca2+ Exchanger in Agonist-Induced Ca2+ Signaling in Cultured Rat Astrocytes". en. In: *Journal of Neurochemistry* 67.5 (Nov.

1996), pp. 1840–1845. ISSN: 1471-4159. DOI: 10.1046/j.1471-4159.1996.67051840.x. URL: http://onlinelibrary.wiley.com/doi/10.1046/j.1471-4159.1996.67051840.x/abstract (visited on 04/12/2017).

[123] A. S. Yu, S. C. Hebert, S. L. Lee, B. M. Brenner, and J. Lytton. "Identification and localization of renal Na(+)-Ca2+ exchanger by polymerase chain reaction". eng. In: *Am. J. Physiol.* 263.4 Pt 2 (Oct. 1992), F680–685. ISSN: 0002-9513.

[124] Toshio Matsuda, Kazuhiro Takuma, and Akemichi Baba. "Na⁺-Ca²⁺ Exchanger: Physiology and Pharmacology". In: *The Japanese Journal of Pharmacology* 74.1 (1997), pp. 1–20. DOI: 10.1254/jjp.74.1.

[125] Jun Liao, Fabrizio Marinelli, Changkeun Lee, Yihe Huang, José D. Faraldo-Gómez, and Youxing Jiang. "Mechanism of extracellular ion exchange and binding-site occlusion in a sodium/calcium exchanger". en. In: *Nat Struct Mol Biol* 23.6 (June 2016), pp. 590–599. ISSN: 1545-9993. DOI: 10.1038/nsmb.3230. URL: http://www.nature.com/nsmb/journal/v23/n6/full/nsmb.3230.html#ref2 (visited on 04/12/2017).

[126] Moshe Giladi, Su Youn Lee, Reuben Hiller, Ka Young Chung, and Daniel Khananshvili. "Structure-dynamic determinants governing a mode of regulatory response and propagation of allosteric signal in splice variants of Na+/Ca2+ exchange (NCX) proteins". en. In: *Biochemical Journal* 465.3 (Feb. 2015), pp. 489–501. ISSN: 0264-6021, 1470-8728. DOI: 10.1042/BJ20141036. URL: http://www.biochemj.org/content/465/3/489 (visited on 04/12/2017).

[127] Michela Ottolia, Debora A. Nicoll, and Kenneth D. Philipson. "Roles of Two Ca2+-binding Domains in Regulation of the Cardiac Na+-Ca2+ Exchanger". en. In: *J. Biol. Chem.* 284.47 (Nov. 2009), pp. 32735–32741. ISSN: 0021-9258, 1083-351X. DOI: 10.1074/jbc.M109.055434. URL: http://www.jbc.org/content/284/47/32735 (visited on 04/12/2017).

[128] Moshe Giladi, Hilla Bohbot, Tal Buki, Dan H. Schulze, Reuben Hiller, and Daniel Khananshvili. "Dynamic features of allosteric Ca2+ sensor in tissue-specific NCX variants". eng. In: *Cell Calcium* 51.6 (June 2012), pp. 478–485. ISSN: 1532-1991. DOI: 10.1016/j.ceca.2012.04.007.

[129] Mousheng Wu, Shuilong Tong, Jennifer Gonzalez, Vasanthi Jayaraman, John L. Spudich, and Lei Zheng. "Structural basis of the Ca2+ inhibitory mechanism of Drosophila Na+/Ca2+ exchanger CALX and its modification by alternative splicing". eng. In: *Structure* 19.10 (Oct. 2011), pp. 1509–1517. ISSN: 1878-4186. DOI: 10.1016/j.str.2011.07.008.

[130] Moshe Giladi, Yehezkel Sasson, Xianyang Fang, Reuben Hiller, Tal Buki, Yun-Xing Wang, Joel A. Hirsch, and Daniel Khananshvili. "A Common Ca2+-Driven Interdomain Module Governs Eukaryotic NCX Regulation". In: *PLOS ONE* 7.6 (June 2012), e39985. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0039985. URL: http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0039985 (visited on 04/12/2017).

[131] William L. Jorgensen, Jayaraman Chandrasekhar, Jeffry D. Madura, Roger W. Impey, and Michael L. Klein. "Comparison of simple potential functions for simulating liquid water". In: *Journal of Chemical Physics* 79 (July 1983), pp. 926–935. ISSN: 0021-9606. DOI: 10.1063/1.445869. URL: http://adsabs.harvard.edu/abs/1983JChPh..79..926J (visited on 05/04/2017).

[132] B.R. Brooks, C.L. Brooks, A.D. MacKerell, L. Nilsson, R.J. Petrella, B. Roux, Y. Won, G. Archontis, C. Bartels, S. Boresch, A. Caflisch, L. Caves, Q. Cui, A.R. Dinner, M. Feig, S. Fischer, J. Gao, M. Hodoscek, W. Im, K. Kuczera, T. Lazaridis, J. Ma, V. Ovchinnikov, E. Paci, R.W. Pastor, C.B. Post, J.Z. Pu, M. Schaefer, B. Tidor, R. M. Venable, H. L. Woodcock, X. Wu, W. Yang, D.M. York, and M. Karplus. "CHARMM: The Biomolecular Simulation Program". In: *J Comput Chem* 30.10 (July 2009), pp. 1545–1614. ISSN: 0192-8651. DOI: 10.1002/jcc.21287. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2810661/ (visited on 01/20/2017).

[133] Mark James Abraham, Teemu Murtola, Roland Schulz, Szilárd Páll, Jeremy C. Smith, Berk Hess, and Erik Lindahl. "GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers". In: *SoftwareX* 1–2 (Sept. 2015), pp. 19–25. ISSN: 2352-7110. DOI: 10.1016/j.softx.2015.06.001. URL: //www.sciencedirect.com/science/article/pii/S2352711015000059 (visited on 01/20/2017).

[134] Eric Jones, Travis Oliphant, Pearu Peterson, et al. *SciPy: Open source scientific tools for Python*. [Online; accessed 2017-01-20]. 2001–. URL: http://www.scipy.org/.

[135] Li-Quan Yang, Peng Sang, Yan Tao, Yun-Xin Fu, Ke-Qin Zhang, Yue-Hui Xie, and Shu-Qun Liu. "Protein dynamics and motions in relation to their functions: several case studies and the underlying mechanisms". In: *J Biomol Struct Dyn* 32.3 (Mar. 2014), pp. 372–393. ISSN: 0739-1102. DOI: 10.1080/07391102.2013.770372. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3919177/ (visited on 04/26/2017).

[136] Moshe Giladi, Su Youn Lee, Yarden Ariely, Yotam Teldan, Rotem Granit, Roi Strulovich, Yoni Haitin, Ka Young Chung, and Daniel Khananshvili. "Structure-based dynamic arrays in regulatory domains of sodium-calcium exchanger (NCX) isoforms". eng. In: *Sci Rep* 7.1 (Apr. 2017), p. 993. ISSN: 2045-2322. DOI: 10.1038/s41598-017-01102-x.

[137] Andrew Brown. *J. D. Bernal: The Sage of Science*. Google-Books-ID: Jnb3uLRXvQ4C. OUP Oxford, Nov. 24, 2005. 594 pp. ISBN: 978-0-19-157950-9.