# Developments on Enhanced Sampling and Machine Learning Analysis Techniques for Understanding Biomolecular Events

*Pedro Juan Buigues Jorro*

A dissertation submitted in partial fulfillment

of the requirements for the degree of

**Doctor of Philosophy**

of

**University College London**.

Department of Physics and Astronomy

University College London

November 23, 2023

I, Pedro Juan Buigues Jorro, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

# Abstract

The research described in this work rises from the current challenges in molecular dynamics (MD) simulations. Although these simulations provide accurate and high-resolution insights on the dynamics of biomolecular events, the timescales needed to observe relevant events such as ligand-unbinding, protein-protein interactions and protein folding, for instance, are not currently reachable for most scientists with classical MD methods. Additionally, MD simulations are intrinsically complex and high-dimensional, which makes it often difficult to elucidate and gain insights from.

To tackle the challenges in MD, an iterative protocol for ligand unbinding followed by a machine learning (ML) analysis allowed for the investigation of the unbinding of Cyclin-Dependent Kinase 2 (CDK2) inhibitors and the long-acting muscarinic antagonists for the human Muscarinic Receptor 3 (hMR3). This approach allowed a deeper understanding of the unbinding path and the underlying protein-ligand interactions. This was achieved by obtaining an approximated transition state (TS) from the unbinding path and generating downhill simulations to train two ML models to predict the outcome. ML is a powerful tool for learning to predict from complex data. However, one of the key challenges is that many models are often considered black boxes. With explainable AI techniques it is possible to gain insights from models and understand how the relationship between input features and their predictions. In this work, we developed a protocol for assessing this in a model-agnostic way and develop a framework to test this for correlated time-series data with both 1D and 2D analytical datasets.

Additionally, a problem-tailored Hamiltonian replica exchange methodology was also developed to aid in the research of systems mainly governed by electrostatic interactions. This is useful especially for phosphate-related enzymes where metal ions play a role in catalysis and active site geometries. This was tested on several systems leading to the CRISPR Cas1/Cas2 system. Results on the modelled complex hinted at a possible two-metal ion coordination in the active site due to major rearrangements and a $K^+$ ion transitioning from the bulk to form part of the coordination.

# Impact Statement

The methods I have further develop together with my collaborators during the time up to this thesis have contributed in efforts towards challenges in molecular dynamics. This was done in hopes they become useful for researchers when looking at protein-ligand interactions, more specifically drug-target interactions.

Having always drug design in mind, the MLTSA is able to give insights in drug discovery for improving the quality from a hit to lead molecule. Structural motifs and further mutations are suggested to explore the role of this interactions, as well as pharmacophores. This method is open sourced and available as a GitHub repository with Jupyter Notebook tutorials and documentation. We also added new information to the already increasing pool of knowledge on the unbinding of CDK2 inhibitors for cancer treatment, as well as the bronchodilators for hMR3. All of this information, with GIF and PDB trajectories including the unbinding for these systems is available in a GitHub repository.

Additionally, the ACHREMD will also help fundamental science in deepening the understanding of phosphate-related enzymes which are crucial in life, more importantly in human regulation and downstream signaling. The method will allow to study them in more detail and validate their conformations and mechanisms. Future academic output is expected from the unpublished results, as well as further experimentation on CRISPR's Cas1/Cas2 complex which is often overshadowed by Cas9, while still being a crucial step on the whole system.

Computational drug discovery is at its peak right now, getting the attention and trustworthiness it deserves, and hopefully expanding from the funding. Machine Learning is key to advance this field to the next step and pushing its boundaries is

important.

Hopefully, both academia and the public can make use of the tools developed and the knowledge gained during this thesis, as well as pave the way for future collaborations and studies on these systems. Future efforts on making the MLTSA and ACHREMD a user-friendly package are also on the way.

# Acknowledgements

I wish I could acknowledge all of the people that made this journey easier for me and supported me. I'd like to start with my supervisor Edina Rosta, for her full support from before even starting my PhD, giving me an opportunity to grow as a scientist and her invaluable feedback during this research journey. I also would like to thank all of the past and present members of the Rosta group, who I consider as *a bunch of great people* to begin with. Without their help and support I wouldn't be where I am today, neither does this work. Thank you for the never ending discussions, the precious moments we shared together and your helpful contributions.

To the amazing people I've met in London, thank you for making my stay much more enjoyable. I'd like to thank my group of close friends which welcomed me with open arms, in particular Ivan, Aura and, of course, my beloved Ana, whose unconditional support and love helped me even in my darkest moments. I'd like to thank her parents and sister as well, for making me feel at home even when COVID strayed us all, and welcomed me as one in their family.

Lastly, thanks to my beloved mother and my two siblings that always rooted for me. Thanks for encouraging me, for their love and understanding and for always believing in me. Thanks to my father, who sadly passed away during this journey, for always feeding my scientific curiosity which always burned inside me and guided me here.

*”Life is not easy for any of us. But what of that? We must have perseverance and above all confidence in ourselves. We must believe that we are gifted for something and that this thing must be attained.”*

-MARIE SALOMEA SKŁODOWSKA–CURIE

*”If I have seen further than others, it is by standing upon the shoulders of giants.”*

-SIR ISAAC NEWTON

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introductory Material

Before delving too deep into the material needed to understand the concepts and background of the different studies within this thesis, I will provide some context to the reader on the title of the work that will be presented. *Developments* is referred to both the advancements proposed in this work as well as the literature review of the different state-of-the-art methods available mentioned in this work for both *enhanced sampling methods* and *machine learning analysis techniques*. All of these methods are outlined within the context of relevant *biomolecular events* which correspond to the systems used for validation and testing within this work.

In the **first chapter**, a general introduction to the field of biomolecular simulation, current challenges and future directions can be found. The principles of protein kinetics will be introduced. The sampling problem derived from the current limitations of computer simulations will be discussed, followed by the current attempts to solve the problem in the shape of enhanced sampling techniques. After that, a brief introduction to machine learning, deep learning models, and its usage on molecular dynamics and its analysis will end the introductory material, hopefully with enough material to understand the rest of the thesis.

After the introduction, I will present chapters relating to 2 independent projects: our Machine Learning Transition State Analysis (MLTSA) and the Atom-Charge Replica Exchange Molecular Dynamics (ACHREMD); that meet at the interface between enhanced sampling and machine learning, all within the field of molecular dynamics.

- **Chapter 2** will cover the general methodology followed for both projects. Each of the projects will have a sub-section explaining both data generation and analysis as well the algorithms involved. Any methodology not included in this general one will be appended to the result chapters 3,4 and 5 in the form of *Computational Details*.

- **Chapter 3** will further explore the idea behind MLTSA, its validation and its functionality in other dynamical contexts, as well as using improved ML architectures.

- **Chapter 4** will present the results of our unbinding-MLTSA original paper on Cycling-Dependent Kinase 2 (CDK2), which validated its suitability for obtaining ligand-unbinding and gaining insights on the unbinding path through ML techniques.

- **Chapter 5** will explore the study of inhibitor ligand unbinding on the human Muscarinic Receptor 3 (hMR3) and the thorough analysis on the unbinding through MLTSA as a follow-up of the previous paper.

- **Chapter 6** will present the preliminary results of our ACHREMD, developed to improve the sampling of electrostatic potential-energy relevant events such as Mg ion coordination and dissociation.

## Acknowledgement of Reproduced Material

Most of the results present in this thesis contain material reproduced as they were published or in an extended form. According to the copyright policy of the concerned publishers, this includes:

- Content reproduced from Ref. [1] with permission from the American Chemical Society.

- Content reproduced from Ref. [2] under the terms of the Creative Commons Attribution.

Reprinted with permission from Badaoui, M.; Buigues, P. J.; Berta, D.; Mandana, G. M.; Gu, H.; Földes, T.; Dickson, C. J.; Hornak, V.; Kato, M.; Molteni, C.; Parsons, S.; Rosta, E. *Journal of Chemical Theory and Computation* **2022**, *18*, 2543–2555 Copyright 2022 American Chemical Society.

## 1.1 Simulating Biomolecular Events

Simulating biomolecular events involves using computer models to simulate the behavior of molecules and chemical reactions in biological systems. This can include simulating protein folding, drug-protein interactions, and the dynamics of large biomolecular complexes [3]. These simulations can be useful in understanding the underlying biochemistry of biological systems and can also be used in drug discovery and design. Running computer simulations is often more cost-effective and accessible compared to conducting complex experimental studies. Experimental techniques involving complex equipment, reagents, and facilities can be expensive and require specialized expertise. Additionally, in simulations, researchers can control various parameters, such as temperature, pressure, and pH, to explore the effects of different conditions on biomolecular interactions. This level of control is often challenging to achieve in experimental settings. The access to molecular details, controlled environment, the time and length scales they can reach, and the quantitative data they can provide offer several advantages over real experiments, although not free of limitations, a combination of experiments and simulations is often the most powerful approach for gaining a comprehensive understanding of biomolecular events.

There are various methods and algorithms used in biomolecular simulation, such as molecular dynamics, Monte Carlo methods, and even computational docking. These simulations are performed using specialized software and require a good understanding of the physics and chemistry of biomolecules. An introduction to concepts such as protein kinetics, their relevance and usage can be found in this section.

### 1.1.1 Overview

Biomolecular simulations are a powerful tool for understanding the behavior of biological molecules such as proteins, nucleic acids and lipids [4]. It allows the study of complex interactions and reactions occurring within living systems, which can lead to new insights into disease mechanisms and drug developments.

There are several different types of biomolecular simulations [3], some of the most popular for biomolecular events are:

- **Molecular Dynamics (MD):** MD simulations use the laws of physics to model the motion of atoms and molecules over time. They can be used to study in atomic detail the structural and dynamic properties of biomolecules such as their conformational changes and their interactions with other molecules. They preserve the kinetics of the system and it is an all-purpose simulation method, however, it is unable to model the chemical process of bond-breaking and reactions explicitly and it can be computationally demanding especially for large systems.

- **Monte Carlo (MC):** MC simulations use random sampling to explore the possible configurations of a biomolecule. They can be used to study the thermodynamics properties of biomolecules such as their stability and binding affinity. However, they can be less reliable for studying dynamics and they require large number of simulations to provide accurate results.

- **Brownian Dynamics (BD):** BD simulations are a type of MD simulation that models the effects of thermal fluctuations and solvent on biomolecules. Despite being a simpler version, they can be used to study the diffusion and transport of biomolecules in solution. Flexible and simple yet they require a detailed representation of the solvent interaction with the molecules and can be computationally demanding as well since obtaining accurate results often require large times.

- **Coarse-grained Models (CG):** A simplified representation of biomolecules in a higher level of abstraction, such as the bead-spring model. These are less computationally demanding and are suited for large complex biomolecular systems.They can be less accurate than all-atom models and require developing detailed representations of the system including the mapping of the atoms to the CG beads and the potentials between them. There is a lack of standardization in the definition of CG models and accuracy can also depend on the models.

- **Docking:** Although they simulate the interaction between two partners, they do not explore any kinetics. They are used to predict the binding of small molecules or a protein to another protein. They can be used to study interactions of drugs with targets and identify new drug candidates. They can be sensitive to the choice of force field and the accuracy of the predictions can vary depending on the quality of the input structures. They also require a scoring function which is very challenging to accurately rank drug candidates with.

Of course, any **hybrid methods** between the previous types of simulations, such as **hybrid MC** and **CG-MD** can prove useful for detailing structural dynamic and thermodynamic information, especially in complex environments or macromolecules such as polymers, difficult to study with all-atom models.

These are the most common and widely used methods, but note these are mostly classical methods. Electronic structure methods are sophisticated enough to describe chemical reactions, although these methods provide much more accurate results they require challenging computations. However there are many other types of biomolecular simulations that have been developed for specific applications or to address specific questions, some other examples include: **Continuum solvent models**, without explicit solvent, **QM/MM**, quantum mechanics/molecular mechanics simulations to study their electronic structure, **Kinetic MC**, used to study the kinetics by combining MC sampling with the rate constants of chemical reactions, and many others. These are just a few examples, the field of biomolecular simulation is constantly evolving and new methods are being developed constantly.

The scope of biomolecular simulations is broad, they provide a wide range of

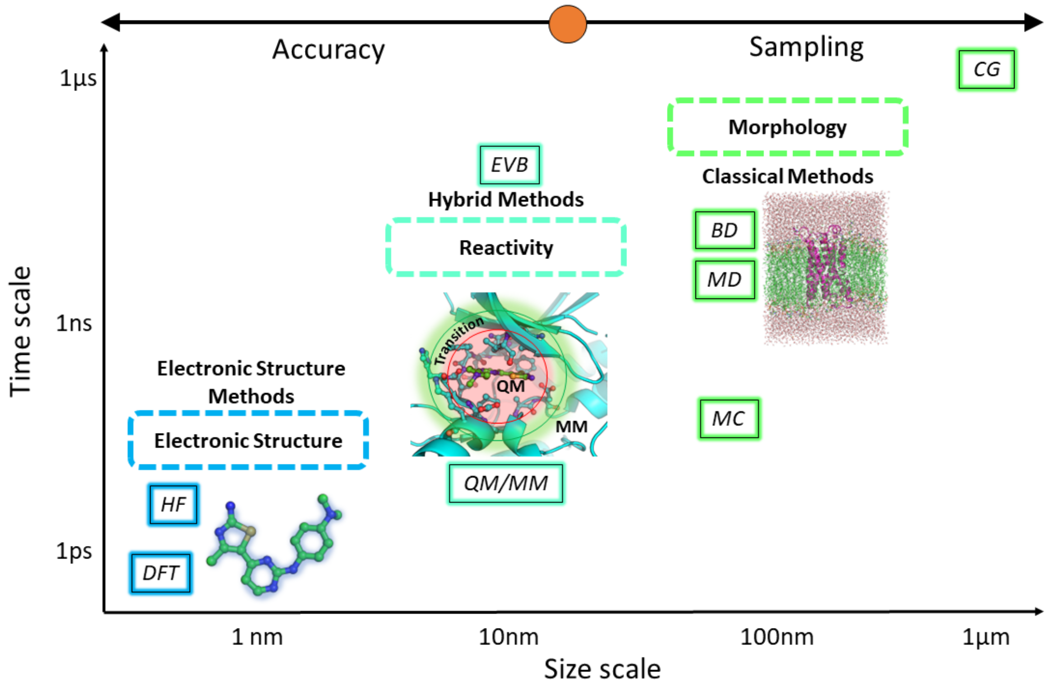


**Figure 1.1:** Most popular biomolecular simulation methods available for the different timescales and sizes to study. For the electronic structure methods (blue), DFT corresponds to density functional theory and HF to Hatree-Fock theory. For the hybrid methods (aquamarine), QM/MM to quantum mechanics molecular mechanics and EVB to Empirical Velence Bond methods. For classic methods (green), BD to Brownian dynamics, MD to molecular dynamics, MC to Monte Carlo and CG to coarse graining methods. Note that the bigger the size of observable events, the less accurate the methods can be and the more focus on sampling.

applications including durg discovery, protein engineering, enzyme design, and the study of biomolecular interactions. They can be used to predict the binding of small molecules or proteins to other proteins; enzyme mechanisms, to study enzymatic reactions; protein-protein interactions, complexes and their dynamics; to study lipid bilayers and membrane proteins, their structure and dynamics; nucleic acids, including DNA and RNA; polimers; and even biomolecular transport in solution such as diffusion and transport through complex environments.

Again, when choosing between a classical method or an electronic structure method the decision often depends on the timescales one wishes to simulate and the size of the relevant event to study to start with. This idea is illustrated in Fig.1.2 where the different events occurr at different time and scale sizes. In Fig.1.1 the dif-

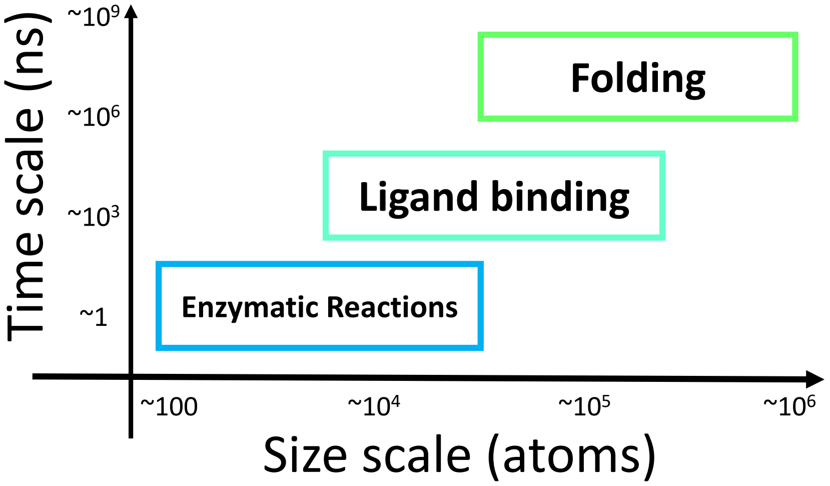


**Figure 1.2:** Diagram illustrating the relationship between the timescales and sizes of different biomolecular events of interest. The colours match the different available methods for electronic structure (blue), hybrid (aquamarine) and classic simulation methods (green).

ferent methods that are available depending on the size and the timescales to study can be found. Note that there is a relationship between the colours in the first figure and the former, since to be able to observe reactions one has to go down to the electronic structure, whereas classical methods can describe folding. One may wish to explore the affinity of a new protein inhibitor through docking and subsequent MD simulation, whereas an enzyme design effort might profit more from a QM/MM calculation in order to see reactivity. Since most of the work in this thesis involves protein-ligand interactions mostly relevant for drug discovery, MD has proven sufficient for this work. However, geometry and parameter optimizations using higher level methods have proven useful to improve the accuracy of the results.

### 1.1.2 Protein Kinetics

Understanding protein kinetics requires a basic understanding of several key concepts, including protein structure, protein dynamics, and the nature of protein interactions [5]. Both experimental and computational techniques can be used to study protein kinetics, and they provide valuable insights into the behavior of proteins over time [5].

**Protein structure** refers to the three-dimensional arrangement of atoms in a protein molecule, which determines the protein's function and stability. Proteins are made up of a linear sequence of amino acids, which fold into a specific structure that is stabilized by various types of interactions, including hydrogen bonds, ionic bonds, and hydrophobic interactions. This specific structure is called the **secondary structure**, which is mainly comprised of alpha helices, beta sheets and random coils. The final three-dimensional arrangement of a protein's secondary structure is called *tertiary structure* including different protein domains, whereas the protein complex combining several domains from different proteins is called *quaternary structure*.

**Protein dynamics** refers to the movements of individual atoms within a protein molecule over time. These movements are driven by thermal energy, and they play a critical role in determining the protein's stability, function, and interactions with other molecules. At higher temperatures, the thermal energy of the protein atoms is increased, leading to increased movements and fluctuations.

The nature of protein interactions is also important for understanding protein kinetics. Proteins interact with other molecules through a variety of mechanisms, including van der Waals interactions, hydrogen bonding, and ionic interactions. These interactions can either be static or dynamic, and they play a critical role in determining the stability and function of the protein.

In order to study protein kinetics, a number of experimental and computational techniques have been developed. Experimental techniques, such as fluorescence spectroscopy and mass spectrometry, provide valuable insights into the behavior of proteins over time [6]. Computational techniques, such as molecular dynamics simulations, provide a complementary approach for studying protein kinetics by allowing researchers to observe the behavior of proteins at the atomic scale.

### Protein-Ligand Complex

A **Protein-Ligand Complex** is formed when a protein binds to a ligand molecule. Proteins have specific binding sites where ligands can interact with them, and the binding process is often highly specific and reversible. The interaction between a protein and a ligand is characterized by two key rate constants: the association rate constant ($k_{on}$) and the dissociation rate constant ($k_{off}$). $k_{on}$ is a measure of how quickly a ligand binds to the protein. It is defined as the rate at which the protein-ligand complex is formed from the unbound protein and ligand. It is influenced by factors such as the collision frequency between the protein and ligand and their relative orientations. $k_{off}$, also known as the dissociation rate constant, is a measure of how quickly the protein-ligand complex dissociates into the unbound protein and ligand. It depends on factors like the stability of the protein-ligand interactions and the energy required for separation. $k_{on}$ and $k_{off}$ have units of $M^{-1}s^{-1}$ and $s^{-1}$ respectively. When in equilibrium, the binding transition $[P]+[L]->[PL]$ should be balanced by the unbinding transition $[PL]->[P]+[L]$, where $[P]$ is the concentration of unbound free protein receptors, $[L]$ the unbound free ligand concentration and $[PL]$ the protein-ligand complex concentration. This equilibrium would be represented as

$$k_{on}[P][L] = k_{off}[PL] \tag{1.1}$$

This association would be governed by a binding constant $K_a$ defined by

$$K_a = \frac{k_{on}}{k_{off}} = \frac{[PL]}{[P][L]} \tag{1.2}$$

Its inverse quantity, the dissociation constant ($K_d = 1/K_a$) for the formation of the protein-ligand complex can be calculated from the association and dissociation rate constants as well

$$K_d = \frac{k_{off}}{k_{on}} = \frac{[P][L]}{[PL]} \tag{1.3}$$

$K_d$ is a measure of the strength of the interaction between the protein and the ligand. A lower $K_d$ indicates a tighter binding between the protein and the ligand, while a higher $K_d$ indicates a weaker binding.

At equilibrium, the rate of formation of the protein-ligand complex is equal to the rate of dissociation, and the concentration of the protein-ligand complex remains constant. The equilibrium constant $K_d$ can be used to predict the concentration of the protein-ligand complex at equilibrium for a given concentration of protein and ligand. Overall, the association rate constant $k_on$ and the dissociation rate constant $k_off$ are important parameters that govern the formation and stability of protein-ligand complexes, and are used to describe the kinetics of ligand binding to proteins. As seen in Fig.1.3, the free energy profile of an unbinding may not be straight and have several transition states ($TS'/TS''$) as well as intermediate states ($[PL]'/[PL]''$). This could be related back to major movements of the protein such as loop movements, pockets opening, helices moving, or even accommodation of the water molecules. Although $k_{off}$ and $k_{on}$ are the kinetic rates, the binding free energies ($\Delta G$) are key to this process.

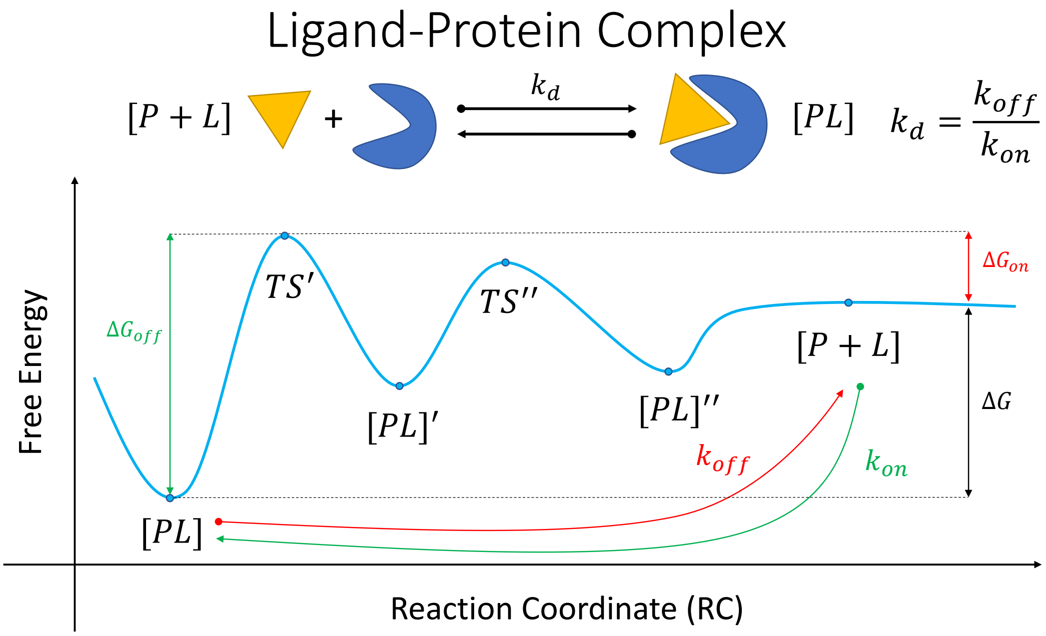


**Figure 1.3:** Diagram of the complex evolution of the free energy profile from the associated protein-ligand state [PL], to the dissociated state [P+L], passing through several intermediates ([PL]' and [PL]') and transition states (TS' and TS") for a given reaction coordinate (RC). The dissociation constant $K_d$ for the whole process is the relation between the equilibrium constants for dissociation $k_{off}$ and association $k_{on}$. The binding free energies associated ($\Delta G_{on}$ and $\Delta G_{off}$) to both processes relates back to the energy $\Delta G$ corresponding to the whole dissociation process.

The binding free energy ($\Delta G$) is the change in free energy that occurs when a ligand binds to a protein. It is defined as the energy difference between the bound ($\Delta G_{on}$) and completely unbound states ($\Delta G_{off}$) (see Fig.1.3). It is the sum of the enthalpy change ($\Delta H$) and the entropy change ($\Delta S$) of the system:

$$\Delta G = \Delta H - T\Delta S \tag{1.4}$$

where T is the absolute temperature in Kelvin. A negative binding free energy ($\Delta G < 0$) indicates that the binding is thermodynamically favorable, while a positive binding free energy ($\Delta G > 0$) indicates that the binding is unfavorable.

For the binding of protein and ligand molecules in solution, the molar Gibbs free energy $\Delta G$ or **binding affinity** is related to this constant by

$$\Delta G = RT \ln \frac{K_d}{c^o} \tag{1.5}$$

where $R$ is the ideal gas constant, $T$ temperature and $c^0$ the standard reference concentration (which is 1 mol/L).

However, with the usual free energy calculations done typically in MD, the binding free energy is obtained from the simulations and can be used to calculate the dissociation constant ($K_d$) for the protein-ligand complex:

$$K_d = e^{\frac{\Delta G}{RT}} \tag{1.6}$$

where R is the gas constant and T is the absolute temperature in Kelvin. A lower $K_d$ value indicates a stronger binding between the protein and the ligand. $\Delta G_{on}$ and $\Delta G_{off}$ represent the standard free energy change for both the binding and unbinding process, respectively. These are related to $K_d$ by

$$K_d = e^{\frac{\Delta G^\circ_{on}}{RT}} \tag{1.7}$$

$$K_d = e^{\frac{\Delta G^\circ_{off}}{RT}} \tag{1.8}$$

In summary, the binding free energy ($\Delta G$) is a measure of the thermodynamic stability of the protein-ligand complex, while the on-rate constant ($k_{on}$) and off-rate constant ($k_{off}$) are measures of the kinetic properties of the binding process. The dissociation constant ($K_d$) is a measure of the strength of the interaction between the protein and the ligand, and can be calculated from the binding free energy, on-rate constant, and off-rate constant.

Obtaining a free energy profile for an unbinding, one can derive the thermodynamic quantities, such as the ligand's residence time (the inverse of the kinetic constant of the drug-target unbinding $1/k_{off}$). This is particularly relevant for drug discovery, when studying protein agonists/antagonists. The concepts of agonist and antagonist are explored in the next subsection, as well as an overview of relevant pharmacological concepts to protein kinetics and molecular dynamics.

### Agonist, Antagonist and Residence Time

In protein complexes, agonists and antagonists are molecules that interact with the protein and modulate its activity. **Agonists** are molecules that bind to a protein and activate its activity. Agonists can either bind to the same site as an endogenous or natural ligand, or bind to a different site and induce a conformational change that promotes the activity of the protein (**Allostericity**). For example, adrenaline is an agonist that binds to the beta-adrenergic receptor in the plasma membrane of cells and activates the G protein-coupled signaling pathway [7]. **Antagonists**, on the other hand, are molecules that bind to a protein and inhibit its activity. Antagonists can either bind to the same site as the natural ligand and compete for binding, or bind to a different site and inhibit the conformational changes necessary for activity. For example, beta blockers are antagonists that bind to the same site as adrenaline in the beta-adrenergic receptor and prevent its activation by the natural ligand [8].

As seen in Fig.1.4, the concentration of the molecule in the body is directly related to its therapeutic effect and it is decisive on its toxicity as well. The duration of action of the molecule is influenced by the protein-ligand complex concentration reaching the necessary value for its effect.

Pharmaco-related quantities, such as the half-life of a drug molecule, are directly influenced by factors such as the absorption, distribution, metabolism, and excretion (ADME) of the ligand in the body, however, they can be greatly influenced by the kinetics of a ligand binding and unbinding from its target and even

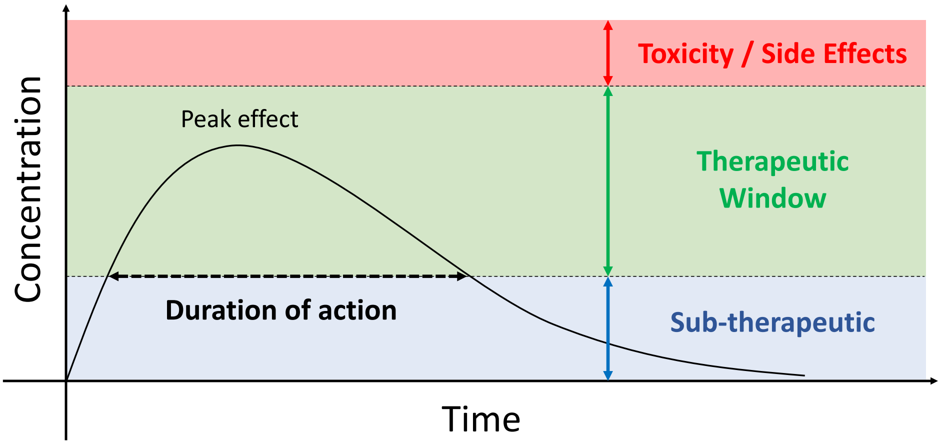


**Figure 1.4:** Evolution of concentration through time and the different effects produced in organisms. Concentration has to be inside the windows for the different effects. Red corresponds to the toxic or accentuated side effects window, green is the therapeutic window required for the maximum therapeutic effect and blue is the sub-therapeutic level of concentration window. The duration of the action is considered to be the time ($t$) spent above the therapeutic window lower limit, having its peak effect at the highest concentration level recorded.

their specificity. While a greater binding affinity may be desirable when searching for a drug, a greater residence time often means a longer therapeutic window, thus reducing the need from the patient to receive multiple doses during the day and ensuring its effect. It has been previously shown that rather than binding affinity, residence time is key to the survival rate [9, 10].

### 1.1.3 Simulating Protein-Ligand Unbinding

Protein-ligand unbinding is a crucial biological process that plays a critical role in many cellular processes, including signal transduction, catalysis and drug-target identification. Understanding the kinetics and thermodynamics of protein-ligand unbinding is of significant interest for both basic research and drug discovery. Simulating protein-ligand unbinding is a challenging task due to the complex nature of the process, which involves changes in both protein and ligand conformations and the release of energy [11]. This makes, in some cases, observing a ligand unbind from its host a *rare event* in the sense of requiring a timescale far greater than what the current computational power can get efficiently, requiring long computational time to get it sampled. Not only that but to recover accurate statistical information, this event has to be observed several times. One of the most successful techniques for observing this is the combination of MD with other sampling techniques and be able to obtain any kinetic information [12].

From simulation data, rate constants can be determined by applying the principles of transition state theory (TST) and the Arrhenius equation. TST states that the rate of a reaction is determined by the rate at which reactants cross a transition state barrier to form products (dissociated complex) or revert to reactants (protein-ligand complex).In simulations, the probability of observing the system in the transition state region can be quantified. Using this probability and the frequency of attempted transitions, the rate constant can be calculated. By extracting the relevant energy barriers and employing statistical mechanics, the rate constant can be calculated from simulation data, enabling the translation of molecular-level insights into experimentally relevant kinetics. The Arrhenius equation relates the rate constant ($k$) to temperature ($T$) and the energy barrier ($E_a$) for the transition state in the following manner:

Original form:

$$k = A \cdot e^{-\frac{E_a}{RT}} \tag{1.9}$$

Second form:

$$k = \frac{k_B T}{h} \cdot e^{-\frac{E_a}{k_B T}} \tag{1.10}$$

Where $k$ is the rate constant wanted, $A$ is the pre-exponential factor which might require calibration or theoretical estimates, $E_a$ is the activation energy obtained from the simulation data, $R$ is the gas constant and $T$ is the temperature in Kelvin for the simulation. For the second form, $k_B$ is the Boltzmann constant and $h$ is the Planck constant. This second form allows to connect the Arrhenius equation to the fundamental constants from statistical mechanics. Although useful, it's important to note that estimating rate constants from simulation data using the Arrhenius equation can have limitations in addition to assuming that the reaction follows an

Arrhenius-type behavior. The accuracy of the estimates depends on the accuracy of the energy landscape calculations, the choice of the reaction coordinate, and the proper determination of transition times. By analyzing the simulated trajectories, quantities such as residence times, binding rates, and dissociation rates can be extracted, providing direct information on the kinetics of the interaction. Comparing these simulation-derived kinetics with experimentally measured (macroscopic) rate constants allows for validation, enhancement, and a deeper understanding of the overall reaction kinetics and the interplay between molecular dynamics and macroscopic behavior.

### Molecular Dynamics

MD is a simulation method involving the calculation of the positions, velocities and forces of each atom in a molecular system over time and using this information to predict the behaviour of the system over a range of time scales. Atoms are usually represented as points in three-dimensional space and bonds are represented as springs. A typical MD protocol involves the calculation of the initial energy of the system to compute the new forces and, through numerical integration predict the next position, velocities and forces in an iterative fashion for a given number of steps.

MD simulations use the laws of classical mechanics to describe the interactions between atoms. For example, the bonds between atoms can be represented using a bond-stretching potential, which describes the energy associated with stretching or compressing the bond between two atoms Similarly, the angles between bonds can be represented using a bond-angle bending potential which describes the energy associated with changing the angle between two bonds. More complex potential functions can also be used to represent other types of interactions between atoms, such as hydrogen bonding, *Van der Waals* interactions and electrostatic interactions. These potential functions are typically based on empirical or theoretical models of the interactions. As Fig.2.1 portrays, the combination of potential functions and parameters assigned to each atom type that describe interactions within the system is called a *force field*. Several force fields have been developed for several systems that are tuned with specific parameter values or term corrections that suit the needs for more experimentally accurate results.

The key advantage of MD simulations is that they can provide a detailed, atomistic description of complex systems, allowing researchers to study the behaviour of individual atoms and the interactions between them. This makes them an important tool for understanding biological and chemical processes as well as designing new drugs, materials and technologies. While being accurate descriptors for morphological features and their dynamics, they allow atomic resolution at bigger timescales. However, note that MD is generally considered to suffer from three main limitations:

- the interaction models or force field may not be accurate enough to describe the desired insights, i.e. bond breaking and others.
- the trajectories are high-dimensional, noisy and present challenges in interpretation and simplification for understanding.
- the limitation on the timestep having to be small enough to integrate accurately and stable makes the sampled timescales shorter than the process of interest.

## Simulation protocol

In order to simulate and study a ligand unbinding event, a **starting structure** is needed. As shown in Fig.1.5, several experimental techniques such as X-Ray crystallography, cryogenic electron microscopy (Cryo-EM) and nuclear magnetic resonance (NMR) can be used to solve a protein's three dimensional structure for simulation purposes. However, often times not all protein sequences are available for study due to experimental challenges or they posses missing loops not able to be solved because of their flexibility. In those cases, *homology modelling* techniques can aid in modelling missing regions or loops after similar already resolve structures, e.g. modelling the active site of a human enzyme after a mouse's. A more ambitious approach is the modelling of proteins with no experimental data available. In this case the newly ML-assisted techniques such as AlphaFold, OpenFold, RosettaFold, an others have proven to be successful in predicting the structure of proteins. Once one has a suitable structure to use, if this includes the ligand in the correct position it can move onto simulation. Otherwise, a docking may be needed to fit in the molecule or manually placing the small molecule by homology as well.

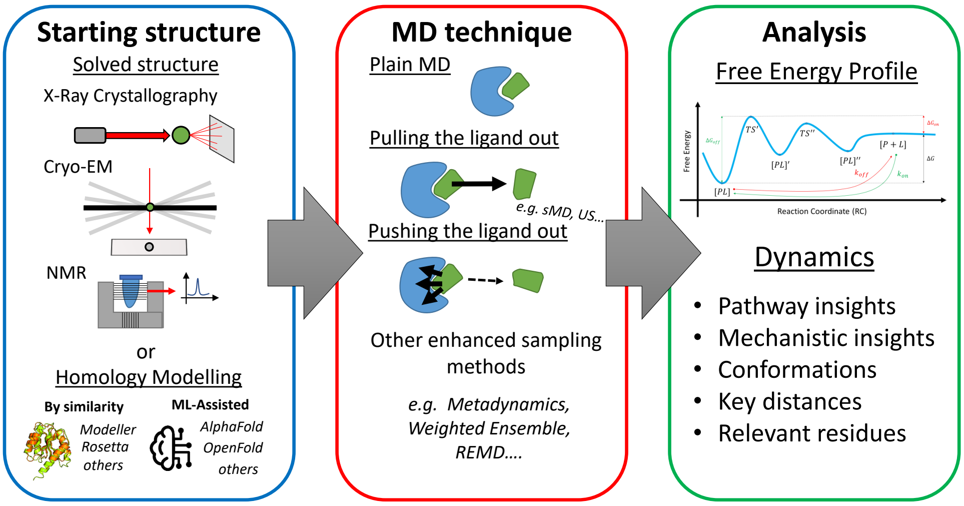


**Figure 1.5:** Diagram of a typical ligand-unbinding MD simulation protocol. Starting with the left, experimental and computational techniques for obtaining a structure. Cryo-EM is Cryogenic electron microscopy, NMR is nuclear magnetic resonance. In the middle, MD techniques for trying to observe the dissociation. Steered MD (sMD) and umbrella sampling (US) are biasing techniques to push or pull the ligand from the binding pocket. In the far right, the free energy profile and dynamics are analyzed to obtain insights and quantities from the obtained trajectories.

After having a suitable structure with a ligand complexed, one has to chose from the range of **MD techniques** available. Although the option of plain running MD is the easiest, the unbinding usually takes long to observe, whereas other techniques involving biasing the interaction directly can be much quicker. Note that biasing the data may involve misleading results that have to be accounted for. We will dig deeper into this in the next sections. However, it is not mandatory to bias the ligand directly, other approaches such as weighted ensemble (WE) and replica exchange MD (REMD) are capable of better sample the complex landscape without having to bias the data and obtain unbiased data.

The final step in the protocol is to **analyze** the obtained trajectories once the unbinding has been observed. Analyzing interactions, distances, and other projections can provide valuable insights onto the mechanisms and pathway the ligand takes on its way out. Additionally, a free energy profile can also be obtained, allowing to calculate kinetic rates and affinities.

Although some examples to tackle the lack of resources to simulate this events have been shown, the sampling problem and the proposed solutions will be explored in the next sections.

## 1.2 The Sampling Problem

In MD simulations, one uses computer algorithms to study the motions and interactions of atoms and molecules over time. These simulations can provide insights into the behavior of complex biological and chemical systems, such as proteins and drugs, that are difficult or impossible to observe experimentally. However, one of the major challenges in MD is **the sampling problem**, which arises from the fact that the timescale of molecular motion can be much longer than the timescale of the simulation. This leads to uncertainties in the accuracy of the simulation results, which can affect our ability to draw meaningful conclusions about the system being studied [13].

### 1.2.1 Timescales

**The timescale problem** refers to the fact that many biologically relevant processes occur on timescales that are much longer than the timescales accessible to simulation (see Fig.1.6). To illustrate the sampling problem in MD, let's consider the example of protein folding. Protein folding is a complex process that involves the formation of a three-dimensional structure from a linear sequence of amino acids. This process can take milliseconds or longer, depending on the protein and the environmental conditions. However, typical molecular dynamics simulations are on the order of nanoseconds to microseconds. This means that even if one runs a simulation for a long time, one may not see the relevant events or transitions that occur in the system. In other words, the simulation may not sample the relevant parts of the free energy landscape.

As previously mentioned, one of the advantages of biomolecular simulations is their quantitative analysis that allows one to obtain kinetic rates from trajectories. One has to make a key distinction here between **macroscopic rates** and **single-molecule kinetics**.

**Macroscopic rates** pertain to the overall behavior of a reaction on a larger scale, involving a large number of molecules. They describe how concentrations of reactants change over time, giving insights into the reaction's global kinetics. These rates are typically obtained from experiments involving bulk measurements and are governed by rate laws. Macroscopic rates obscure individual molecular details.

**Single-molecule kinetics** involves studying reactions or processes at the level of individual molecules. It provides a more detailed and dynamic understanding of molecular behaviors, such as binding/unbinding events or conformational changes. Alike simulations, experimental techniques like single-molecule fluorescence or atomic force microscopy enable the observation of individual molecules, revealing heterogeneity and stochastic behavior that macroscopic rates might overlook.

Macroscopic rates are averages over a large population of molecules and follow deterministic reaction kinetics described by rate laws. They represent the collective behavior of many individual reactions. In contrast, single-molecule kinetics captures the inherent variability and randomness of individual molecular events, often following probabilistic behavior. While macroscopic and microscopic rates are related, they offer distinct perspectives on the same reaction, with single-molecule kinetics providing insights into heterogeneity, rare events, and molecular mechanisms that contribute to the macroscopic behavior.

MD simulations offer a unique bridge between macroscopic rates and single-molecule kinetics by allowing researchers to explore the behavior of individual molecules in a dynamic environment. They capture the stochastic behavior and dynamic heterogeneity that characterizes single-molecule kinetics, providing insights into the underlying molecular mechanisms. While MD simulations excel at providing insights into single-molecule kinetics, they can also be used to derive macroscopic rates indirectly. By aggregating statistical information from multiple simulations or trajectories, researchers can calculate ensemble-averaged properties. These ensemble averages can provide valuable estimates of macroscopic rates that relate to bulk experimental observations. Thus, observing an event and the number of times it can be observed is relevant for accurate analysis.

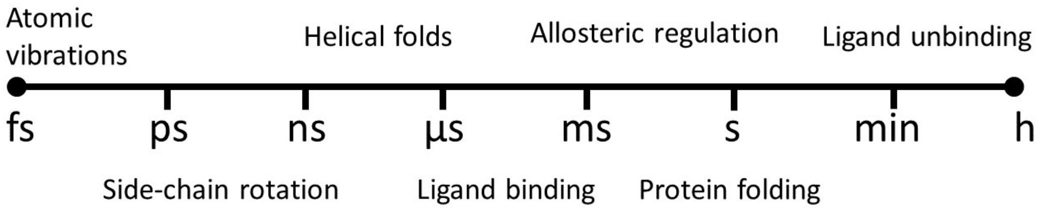


**Figure 1.6:** Scale of the biomolecular events at the required timescales to observe them.

In the context of protein-ligand unbinding, several steps have to be in place for the protein to release its ligand, as shown in 1.3, which all contribute to the overall energy necessary for unbinding. Not only that, but also the contributions from major protein rearrangements and the total exit of the ligand which may go through a protein channel or a membrane in the case of a GPCR, for example, where the residence time of a drug can take up to 24h (tiotropium). Although, with the fast development of computational sciences towards powerful computational hardware such as ANTON and graphics processing units (GPU), researchers have achieved miliseconds-long simulations of proteins of considerable size, these timescales are far smaller and computationally expensive than the ones needed for observing ligand-unbinding events (see Fig.1.6).

### 1.2.2 Uncertainty

**The uncertainty problem** refers to the fact that MD simulations are stochastic like nature. This means that the results can vary depending on the initial conditions and random fluctuations in the system. This can lead to uncertainties in the accuracy of the simulation results, even if one could simulate the system for an infinite amount of time. The uncertainty problem can be exacerbated by the timescale problem, as rare events and transitions may have a large impact on the overall behavior of the system, but are difficult to sample.

The extent of the uncertainty in a molecular dynamics simulation can depend on a number of factors, including the length of the simulation, the force field used to describe the interactions between the atoms, and the size and complexity of the system being studied. For example, simulations of larger systems or longer timescales are likely to have greater uncertainty due to the increased number of atoms and longer timescale of the simulation.

To address the uncertainty problem, researchers often perform multiple independent simulations, each with slightly different initial conditions, and then analyze the average behavior of the system. This approach is known as ensemble simulations and can provide a more accurate estimate of the thermodynamic properties of the system than a single simulation.

## 1.3 Enhanced Sampling Techniques

To address the sampling problem in molecular dynamics, researchers have developed a variety of **enhanced sampling techniques**. These techniques accelerate phase space sampling to make computed properties more reliable and demonstrate rapid convergence [14]. They accelerate MD to overcome high energy barriers using methods *biasing*, for example, allowing us to improve the sampling of the relevant parts of the free energy landscape.

Before getting into the classification of these methods, it is key to understand the concept of **biasing** and **collective variables** (CV).

### 1.3.1 Collective Variables in Molecular Dynamics

In the realm of MD analysis, **collective variables** (CVs) serve as essential tools for capturing and simplifying the complex behavior of biomolecular systems. These variables offer a higher-level perspective that condenses the multi-dimensional dynamics of atoms and molecules into a reduced-dimensional space, allowing researchers to extract meaningful insights from simulations.

CVs can be described as mathematical functions that quantify a chosen aspect of a molecular system's behavior. Rather than considering each atom's position and velocity individually, CVs aggregate information across multiple atoms to represent specific features, conformations, or interactions within the system. By focusing on these collective aspects, CVs provide a streamlined framework for understanding the dynamics and mechanisms governing biomolecular processes. They can be composed of various mathematical functions that define relevant aspects of the system. These functions often involve distances, angles, torsional angles, or combinations thereof. For example, a CV could be the distance between two specific atoms, representing the opening or closing of a binding site during a ligand-receptor interaction. The selection of appropriate CVs depends on the specific scientific question or process of interest. Careful consideration is given to choosing CVs that capture the essential dynamics while avoiding overcomplication. By constructing a minimal set of meaningful CVs, the dimensionality of the system can be reduced, aiding in visualization and analysis.

The primary purpose of CVs is to simplify the analysis and interpretation of molecular dynamics simulations. They enable researchers to monitor and describe the progress of specific events, transitions, or conformational changes in a more intuitive and comprehensible manner. CVs act as bridges between the intricate atomic-level interactions and the macroscopic observables, facilitating the extraction of meaningful mechanistic insights and enabling the exploration of rare events that might otherwise remain elusive. Additionally, some biasing methods allow for the biasing of a specific CV within a given system, allowing to sample regions previously hard to explore. By selecting and constructing appropriate CVs, researchers can uncover hidden insights, unravel dynamic mechanisms, and bridge the gap between microscopic interactions and macroscopic observables.

### 1.3.2 Biasing in Molecular Dynamics

A **biasing** energy is an energetic term $U^{bias}$ added that enables one to obtain a potential energy biased to behave a certain way $\widetilde{U}(x)$:

$$\widetilde{U}(x) = U(x) + U^{bias}(x) \tag{1.11}$$

In MD, this bias energy translates into a bias force $F^{bias}(x)$, thus yiedling:

$$F^{bias}(x) = -\nabla_x U^{bias}(x) \tag{1.12}$$

$$\widetilde{F}(x) = -\nabla_x \widetilde{U}(x) = F(x) + F^{bias}(x) \tag{1.13}$$

This $U^{bias}(x)$ is usually a function of a CV describing the system in a lower coordinate, such as a distance, angle, etc. That means one can apply a **biasing potential** to force the system to take the desired configuration, i.e. pushing a ligand outside of a pocket and moving helices and loops to sample the desired rare event. These methods are called **potential-based biasing methods**.In the same way, a system under the influence of a biased potential energy $\widetilde{U}(x)$, will sample a biased configurational distribution $\widetilde{\gamma}(x)$. Note that in all biasing methods, this biased distribution cannot be used for calculating the free energy directly, as the difference between states given by the biased potential energy has to be computed.

Another set of methods are available in which a separate, distinct probability distribution ($\widetilde{\gamma}(x)$) from the target one $\gamma(x)$ is sampled in a way that the ratio of the two is known or can be estimated numerically. This is done to focus sampling on regions of importance or to flatten the energy landscape towards a uniform distribution easier to sample. Some of these methods are part of the **non-potential-based biasing methods**, where the bias is not added as an additional term, but instead the existing ones are modified by some degree.

In the following subsections 1.3.3 and 1.3.4, an attempt to simplify to a binary classification (see Fig.1.7 top) the immense number of different flavours of enhanced sampling techniques has been made [15]. Note that some methods do not fully fall under one category alone and this is in broad terms, additionally many more methods may fall under the first category rather than the latter. Moreover, section 1.3.5 includes example of hybrid methods as well.

### 1.3.3 Biasing methods

Biasing methods introduce an external bias or force to the system to accelerate or steer the sampling towards certain regions of the phase space. The aim is to over-

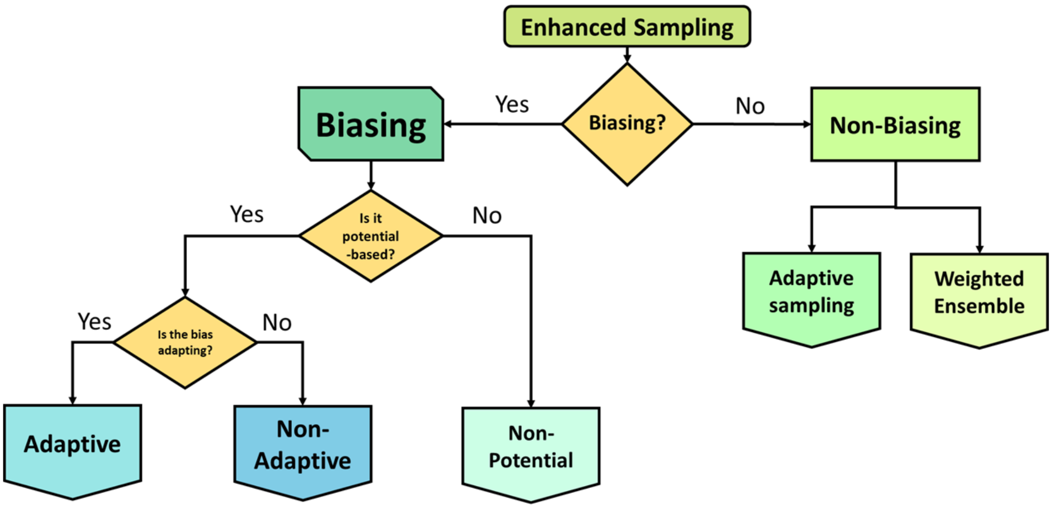


**Figure 1.7:** Diagram of the attempted binary classification of enhanced sampling techniques. The first dividing section is regarding to biasing the system or not. Note, however that the ”non-biasing” methods are information biasing in the sense of biasing the entropy term rather than the enthalpy term. The second level is dividing between potential and non-potential-based approaches. The third level divides them regarding the change of bias through the simulation (adapting or not). Adaptive approaches have a time-dependent feedback between the bias and the system which makes it adapt and ”learn” in a similar fashion to ML methods. Non-adaptive methods have a constant or ”static” bias being applied.

come energy barriers and increase the sampling of rare events or transitions that are difficult to observe in standard simulations. Biasing methods can be further categorized into two subcategories: potential-based methods and non-potential-based methods.

**Potential-based biasing methods**.These methods add an additional term to the potential energy of the system that acts as a biasing potential, which is designed to steer the system towards a particular state or region of the phase space. The biasing potential is typically a function of a CV that describes the progress of the system along a reaction coordinate or a particular degree of freedom. These include:

- Non-adaptive methods, which are designed to flatten the energy landscape in a static way, such as Accelerated MD (aMD) and Gaussian-accelerated MD (GaMD).

- Adaptive methods, where the bias is *learned* during the simulations and adapted on-the-fly. They include metadynamics and well-tempered metadynamics as well as its derived methods, such as infrequent, adaptive Gaussian, etc. Metadynamics methods are based on adding a temporary small bias deposited at the current location of the CV. These adaptive methods also include updating force biases on CVs (Adaptive biasing Force, ABF), cancelling the ensemble average, such as the ones provided by the Colvars Module. In the same fashion, standard, multiple-walker and extended-system ABF implementations are also available.

**Non-potential-based biasing methods**, on the other hand, do not add an external potential to the system but instead modify the dynamics of the simulation to enhance the sampling of rare events or transitions. In these methods, the original configurational distribution is preserved, and the sampling is enhanced by exploiting **transitions to other ensembles**. These include generalized ensemble and replica exchange methods:

- Replica exchange: Multiple coupled simulations are carried out in parallel and they periodically exchange configurations (structures) in thermodynamic states with each other. These states include changes in temperature (parallel tempering), Hamiltonian states (HREMD), etc.
- Expanded ensemble: The states are explored in a single simulation via a based random walk in state space. For example, climbing through a temperature ladder.

In both of them, a mixture of thermodynamics states are sampled within the same simulation framework. Simulations are able to hop between all states through a *hopping process*. The criteria for the transition between states is obtained through *sampler* algorithms following the Metropolis or Gibbs criteria, for example. The Metropolis criterion accept exchanges based on the energy difference between replicas while the Gibbs criterion considers both energy and temperature. This enables more efficient sampling. Both biasing methods, when done correctly, allow the estimation of the equilibrium expectations at each state as well as the free energy differences between the states.

### 1.3.4 Non-biasing methods

**Non-biasing methods**. The benefit of non-biasing methods is that they do not require any prior knowledge of the reaction coordinate or CVs. Additionally, very little post-processing is needed in order to obtain unbiased data. One example is the adaptive seeding methods. In these methods, the sampling is enhanced by specifying starting coordinates, in order to focus on sampling on productive or undersampled regions. Simulations are not restricted to regions, but they are initialized and terminated strategically. They can be divided into:

- Adaptive sampling : these methods search for a more accurate description of the ensemble and take advantage of Markov State Modelling (MSM) which are statistical methods that aim to build a model of the system's dynamics from a series of short, unbiased simulations.

- Weighted ensemble (WE) methods: also referred to as a splitting/replication approach, is capable of finding pathways between macrostates and evaluate transition rates between states. It is based on starting and replicating relatively short MD simulations towards a target state, while terminating simulations that are not making progress towards the state of interest or the end state. Note this target state can also be CV-based.

### 1.3.5 Hybrid methods

Additional hybrid schemes can be created by combining different principles from already enhanced sampling methods. One common approach is to combine a method that focuses on biasing specific degrees of freedom or CVs, with another one that enhances the sampling in a more general way for a larger number of degrees of freedom. This allows for biasing and exploring degrees of freedom that may be missing in the original biased CV when simulating complex events. Some examples are:

- Replica exchange umbrella sampling (REUS), where all replicas are at the same temperature, but their umbrella potential is centered at different locations. Thus allowing exchange between umbrella windows and improving convergence.

- Parallel-tempering metadynamics, that is used for improving the lack of orthogonal degrees of freedom that are not included in the biased CV set.

Many more combinations other than these examples exist, including combinations with metadynamics and ABF, path finding methods, etc.

### 1.3.6 Unbiasing data and free energy estimation

Unbiased properties of the original distribution can be obtained by **reweighting** the modified sampled distribution. This involves calculating averaged and free energies of a distribution using samples from a different one. In order to make use of the obtained data from biased methods, an unbiasing scheme has to be put in place.

**Free energy estimators** are central to the enhanced sampling methods. They allow the calculation of free energy differences between two states without the need for direct sampling.There are several free energy estimators available, each with its own advantages and limitations. Some of the strategies for these are:

- **Directly measured ratios:** This free energy estimator is the simplest and most straightforward method for calculating free energy differences. It involves calculating the ratio of the probability densities of the two states, typically through counting the number of configurations in each state. The free energy difference can be calculated as in: $\Delta F = -kBTln(P_1/P_2)$, where $P_1$ and $P_2$ are the probabilities of observing the system in the two states.

- **Estimations from transition matrices:** This involves constructing a transition matrix from a set of unbiased MD simulations and using it to estimate the free energy difference between two states. The transition matrix contains the probabilities of transitioning between different states of the system, and the free energy difference can be calculated using the ratio of these probabilities.

- **Thermodynamic integration (TI):** This free energy estimator involves integrating the potential energy of the system along a coupling parameter that gradually turns on or off a perturbation, such as a ligand-protein interaction or an external field. The difference in free energy between the forward and backward states can be calculated by integrating the potential energy differences between the two states as a function of the coupling parameter.

- **Bennett acceptance ratio (BAR):** is a method used to calculate free energy differences between two states of a system, typically referred to as the forward and backward states. The method involves computing the ratio of the probability densities of the forward and backward trajectories. The reweighting scheme involves modifying the probability densities of the forward and backward trajectories by a factor that accounts for the difference in statistical weights due to the different conditions.

- **Multistate Bennett acceptance ratio (MBAR):** this is an extension of the BAR method that allows for the calculation of free energy differences between multiple states. MBAR involves computing the ratio of the probability densities of all possible pairwise transitions between the states, and then combining these ratios to obtain the free energy differences between each state.

- **Weighted histogram analysis method (WHAM):** The method involves constructing a histogram of the probability distribution of configurations in each region and then weighting each histogram by a factor that accounts for the biasing potential used in that region. The weighted histograms are then combined to obtain an estimate of the free energy difference between the two states. This method is explained in more detail in the 2.2.1 section.

These are examples of the most common free energy estimators one can use to unbias data when working with enhanced sampling methods, or to simply construct back the energy distribution of a given set of simulations.

Now that the idea of the sampling problem, and its proposed solution (enhanced sampling) has been stated, another old problem from MD gets even more accentuated. As mentioned in section 1.1.3, now that the amount of sampled events and data is much bigger, it is difficult to interpret its meaning or understand the underlying dynamics of the processes, specially with no prior information. In the search of an automated approach to digest this big data, several machine learning approaches to ML in MD have raised, and not only towards that end. ML in MD will be explored in the next section, introducing ML, the context of ML for MD simulations ending with a focus on understanding MD data.

## 1.4 Machine Learning in Molecular Dynamics

Machine learning has emerged as a powerful tool in chemistry and molecular dynamics to predict and analyze molecular properties and behaviors. It involves training algorithms on large datasets of molecular structures and properties, allowing for the development of accurate models for predicting properties that are difficult to measure experimentally. In molecular dynamics, machine learning has been used to accelerate simulations, develop force fields, and enhance sampling methods. The application of machine learning in chemistry and molecular dynamics is rapidly expanding and has the potential to revolutionize the field by providing new insights and enabling the design of novel materials and drugs. Some of the core concepts and relevant ML methodologies will be explored in the next subsection 1.4.1.

### 1.4.1 Machine Learning Overview

This subsection will hopefully provide introduction to ML as a field and some key developments that have recently changed the paradigm and are relevant in this context.

#### Core Concepts

Machine learning is a type of artificial intelligence that involves training algorithms on data to make predictions or decisions without being explicitly programmed. ML involves training algorithms to automatically learn patterns and relationships in data. The goal of machine learning is to use these learned patterns to make accurate predictions or decisions on new, unseen data [16].

There are several types of machine learning, including:

- In **supervised learning**, the algorithm is trained on labeled data, where the correct output is known, and the goal is to learn a mapping between the input and output. Examples are: image classification, where the algorithm learns to identify objects from training with labeled images, and sentiment analysis, where given a text review, the algorithm classifies the review as positive or negative.

- In **unsupervised learning**, the algorithm is trained on unlabeled data, and the goal is to identify patterns or structure in the data. Examples are: clustering, where from a set of unlabeled data, the similar samples are grouped and dimensionality reduction, where given a high-dimensional set, the algorithm identifies and extracts the most relevant features capturing the underlying structure of the data.

- **Reinforcement learning** involves learning through trial and error, where the algorithm interacts with an environment and receives feedback in the form of rewards or penalties. For example: game playing, where an agent learns to play a game by interacting with the environment and receiving rewards or penalties for its actions, and in robotics, where a robot learns to perform tasks such as navigating a maze, adjusting its behaviour to maximize its reward.

Examples of algorithms involved in each type of learning can be found in Fig. 1.8, as well as an illustration to further clarify these concepts that classify ML models by training.

Machine learning models are typically trained using an **optimization algorithm** that adjusts the model's parameters to minimize a **loss function**, which measures the difference between the model's predictions and the true output values in the training data. During training, the model is exposed to a subset of the data, called the **training set**, and the goal is to learn a model that generalizes well to new, unseen data. However, there is a risk of overfitting or underfitting the training data. **Overfitting** occurs when the model becomes too complex and fits the noise in the training data, resulting in poor generalization to new data. **Underfitting** occurs

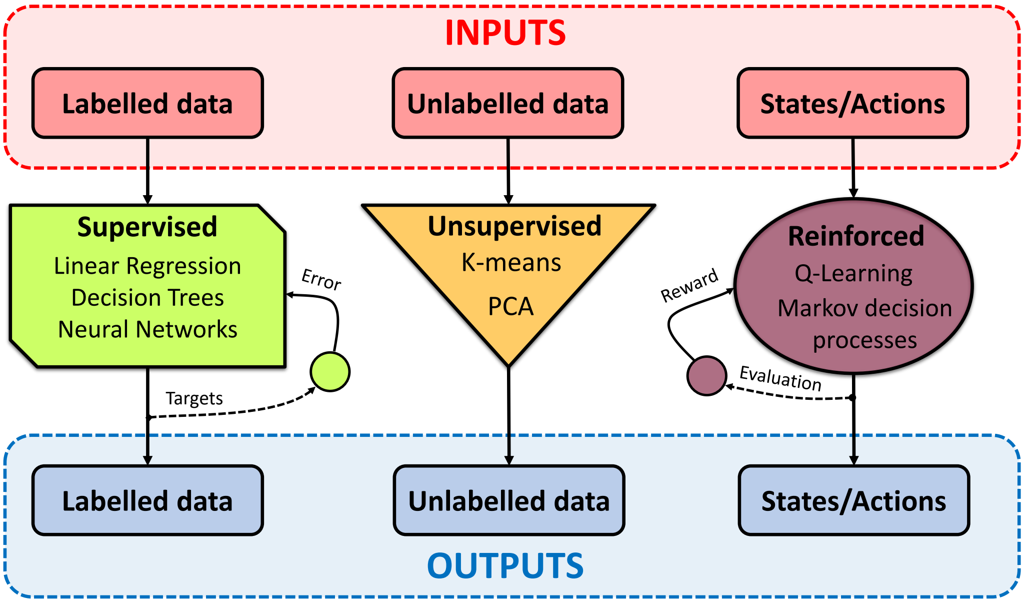


**Figure 1.8:** Diagram of the three types of ML (supervised, unsupervised and reinforcement) in which most ML models classify. Different types of inputs (red) apply to different models (shapes) to obtain different outputs (blue). Note that supervised learning has an extra step for calculating the loss to backpropagate comparing the targets with predictions in a similar way reinforcement learning evaluates the actions taken to compute the rewards.

when the model is too simple and cannot capture the underlying patterns in the data, again resulting in poor generalization.

To avoid overfitting or underfitting, **hyperparameter optimization** is often used to tune the model's hyperparameters, which are parameters that are set before training and cannot be learned from the data. Hyperparameters include things like the learning rate, the number of hidden layers in a neural network, or the number of clusters in a clustering algorithm. Hyperparameter optimization involves selecting the best combination of hyperparameters that result in the best performance on a validation set, which is a separate subset of the data that is not used for training but is used to evaluate the model's performance during training.

Now that the most common concepts of ML have been explained, the next section will move onto the evolution of the ML models for time series which is crucial for this work.

### 1.4.2 From MLP to Transformers

This section will take the reader on a journey through the evolution of neural network (NN) architectures from simple Multilayer Perceptron (MLP) models and progressing to more advanced models such as Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) models, and Gated Recurrent Units (GRUs), and finally, to the Transformer architecture. A visual summary of the different models is provided in Fig.1.9 to illustrate the main contributions towards models able to capture time-series relationships.

#### Multi-Layer Perceptron (MLP)

The MLP is a simple neural network architecture consisting of one or more layers of neurons, each of which is connected to the next layer. The input layer receives the input, and the output layer generates the output [17]. The hidden layers in between can be thought of as intermediate processing layers that transform the input into a more meaningful representation. MLPs use backpropagation to update the weights of the neurons during training.

MLPs are limited by their inability to handle sequential data, which means that they cannot be used for tasks such as natural language processing or speech recognition.

#### Recurrent Neural Networks (RNN)

RNNs were developed to address the limitation of MLPs in handling sequential data [18]. RNNs use a feedback mechanism that allows information to be passed from one time step to the next, making them ideal for sequential data processing. Each time step in an RNN processes the input and updates its internal state based on the current input and its previous state. This allows RNNs to capture temporal dependencies in the data.

However, RNNs suffer from the vanishing gradient problem, where gradients become too small to be useful when backpropagating through many time steps. This limits the ability of RNNs to capture long-term dependencies.

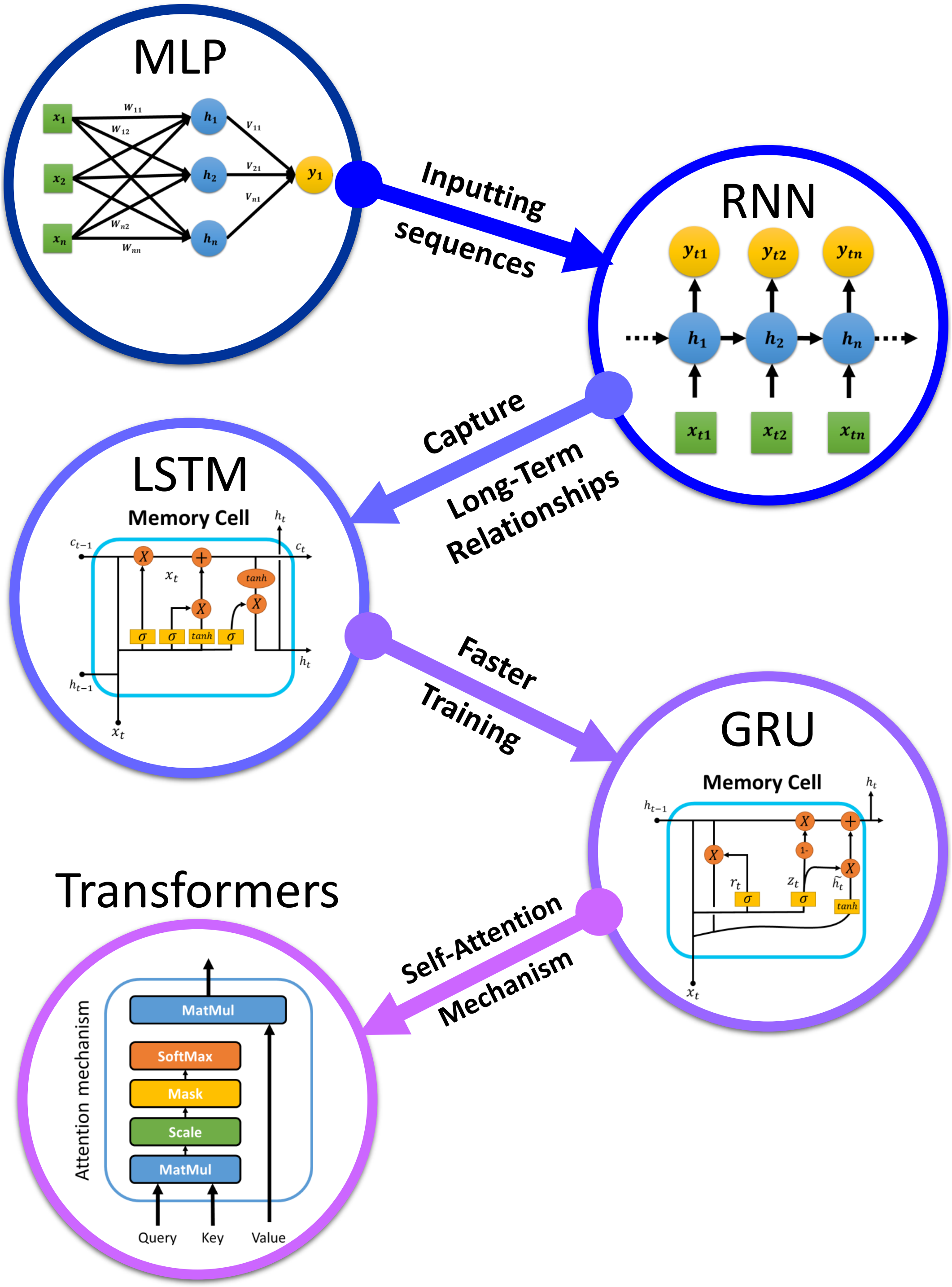


**Figure 1.9:** Diagram of the necessary evolution through all different DL models to get from the simple MLP, to the Transformer architecture with the attention layers. Each model has a diagram highlighting the contribution they brought to the field.

### Long Short-Term Memory (LSTM)

LSTM is a type of RNN that was developed to address the vanishing gradient problem [18]. LSTMs use a gating mechanism that allows them to selectively remember or forget information from previous time steps, making them better at capturing long-term dependencies. LSTMs have three gates: the input gate, the forget gate, and the output gate. The input gate controls whether new input is added to the memory, the forget gate controls whether previous memory is retained, and the output gate controls what is output from the memory.

LSTMs are able to handle long-term dependencies, but they are computationally expensive and can be difficult to train.

### Gated Recurrent Units (GRU)

GRU [19] is another type of gated RNN that is similar to LSTM but has fewer parameters, making it faster to train. GRUs have two gates: the update gate and the reset gate. The update gate controls how much of the previous state is retained, and the reset gate controls how much of the new input is added to the state.

GRUs are faster to train than LSTMs and are better suited for tasks with limited training data. Although not a breakthrough they are a relevant milestone towards the development of time dependence models.

### Transformer

The Transformer is a neural network architecture that was developed for natural language processing tasks, such as machine translation. Unlike RNN-based models, which process input sequentially, the Transformer processes the entire input at once. The Transformer uses a self-attention mechanism to weigh the importance of each word in the input based on the other words in the sequence [20]. This allows the Transformer to capture long-range dependencies without suffering from the vanishing gradient problem. The Transformer consists of an encoder and a decoder. The encoder takes the input sequence and generates a set of feature vectors that represent the input. The decoder then uses these feature vectors to generate the output sequence.

The next steps in the development of neural network architectures will likely involve incorporating attention mechanisms into other types of models, such as CNNs and LSTMs. Additionally, there is ongoing research into developing more efficient and effective training methods for neural networks, as well as exploring other types of neural network architectures that may be better suited for specific tasks.

### 1.4.3 Machine Learning in Molecular Dynamics

As mentioned earlier, ML methods can be used for a variety of applications in MD, such as predicting molecular properties, optimizing force fields, generating accurate potential energy surfaces, reducing the dimensionality of high-dimensional data, improving sampling efficiency, and more. By incorporating ML into MD simulations, researchers can obtain more accurate results, simulate larger systems, and explore new regions of the phase space [21].

Previous efforts in implementing ML methodologies for improving MD in different directions are:

- Mapping **potential Energy Surfaces** (PES) as a function of the atomic coordinates to simulate systems efficiently [22], where the Behler-Parrinello faster than DFT traditional methods, especially for material science.

- Parameterize **force fields** based on experimental and simulation data and optimize them for more accurate MD simulations [23] with NNs such as CGnet.

- Improving **reactive molecular dynamics** (RMD) and improve accuracy and efficiency of simulations for reactions at the MD level [24] using NNs for analysis and classification of local atomic structures.

- **Dimensionality reduction** of the high-dimensional data from MD simulations [25], using PCA, SketchMaps, t-SNE, etc.

- Improve the efficiency of **enhanced sampling schemes** such as in REMD predicting the probability distribution of temperatures of a system [19] using generative artificial intelligence.

- Predict **molecular properties** such as solubility, melting point, reactivity, even binding affinities [26, 27].
- Generate new configurations and improve **sampling** of the phase space of a system [28].

Still, some of the challenges that remain open in ML for MD with room for improvement are:

1. PES and free energy surfaces (FES) to accurately map and describe the system.
2. Coarse graining complex systems while preserving properties of the original system.
3. Estimating molecular kinetics models for transitions between configurations.
4. Sampling probability distributions to avoid unnecessary sampling.

To fight these challenges, recent efforts with different strategies have been explored. An improvement over the Behler-Parrinello network for learning and predicting PES from QM data, is the ANI [29]. While this network has been trained on QM data as well, it is capable to transfer the predictions to other organic molecules by developing transferable NN potentials for single atoms (ANI-1). Deep tensor neural network (DTNN) [30], inspired from language models to learn interactions, and SchNet [31], a deep convolutional nerual network, both learnt a multiscale representation of the molecular properties. They reached highly accurate predictions across the chemical space and configurations and are becoming popular due to their scalability. For coarse graining, CGnet [23] make use of constraints and complex architecture to featurize the cartesian coordinates into internal coordinates. This approach obtained similar results to that of an all-atom molecular dynamics approach. VAMPnets [32] automate constructing kinetic models by using an encoder to transform the molecular configuration to a latent space which is trained on pairs sampled from the MD. Boltzmann generators learnt to sample equilibrium distributions using a generative model and a reweighting procedure. With this, one is able to learn to generate thermodynamics such as the temperature-dependent free energy profiles. Although some of these models are transferable and scalable, not all of them can be applied to complex systems yet and there's still room for improvement.

ML methods benefit from the Chemistry and Physics knowledge to restrict the predictions to be meaningful and applicable [21]. Using physical constraints, like incorporating known physical laws into ML models, ensures that predictions adhere to fundamental principles, such as knowing bond lengths, angles and steric interactions in MD. Physics and Chemistry-informed ML models offer insights into complex systems that may not be apparent from data alone, such as mechanistic explanations, path classification, etc. Due to having multiple ways to generate high-resolution data at the detailed level, some ML methods (PCA, SVM, etc.) benefit from the high-dimensional accurate description, while others require a larger number of samples than features to be able to be implemented accurately. Nevertheless, MD simulations can both provide a lot of data comparable to that of big data with a great number of features and descriptors, due to the nature of the all-atom interactions. In recent years, ML has emerged as a promising approach for extracting meaningful insights from MD simulations data [21, 33].

### 1.4.4 ML for understanding MD

ML methods can be used to analyze various aspects of MD simulations, including the conformational space explored by biomolecules, the thermodynamics of protein-ligand interactions, and the kinetics of protein folding and unfolding [34]. The enormous capacity of the current computational infrastructures is able to generate milliseconds of biomolecular simulations with high resolution, generating terabytes of data. The automation, reproducibility and scalability of ML methods is attractive for researchers in this field. By leveraging the power of ML algorithms, researchers can gain deeper insights into the behavior of biological systems and identify novel targets for drug discovery such as pockets, regions, etc.

Biomolecular simulations are intrinsically high dimensional and noisy, while increasing their size with system complexity (domains, units, subunits, etc.). This fact makes extracting relevant features from data crucial for understanding biophys-

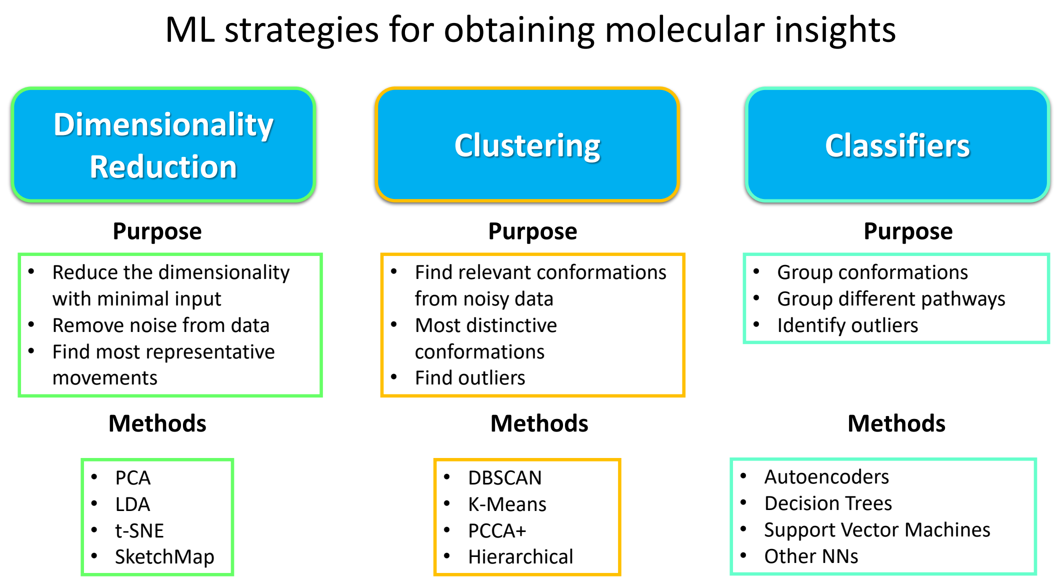


**Figure 1.10:** Examples of the different strategies already explored for understanding MD using atomic coordinates or other features derived from trajectories. On the left, dimensionality reduction algorithms, on the middle, standard clustering methods, and on the left other ML algorithms for classification.

ical properties of molecular processes. In our case, when working with ligand unbinding, complex systems may not produce a clear signal for a human mind to interpret as relevant. Dimensionality reduction models such as PCA have limited interpretability on their own as shown in [33] on calmodulin and the $\beta 2$ adrenergic receptor when using cartesian coordinates from simulation frames. However, their efforts exploring the applicability of supervised and unsupervised ML methods for obtaining molecular insights have shown that ML can quickly perform data-driven analysis of simulations and provide an interpretable overview of the relevant features [33] even in a complex case such as the $\beta 2$ adrenergic receptor.

Some of the strategies that have been explored previous to this work are shown in Fig.1.10. Some ML models such as decision trees allow the user to obtain the importance of the different features for their training due to their nature. On NNs, however, it is more challenging to infer feature importance due to the non-linear nature of the models.

In decision trees, **Gini feature importance** measures the total reduction of the Gini impurity caused by a particular feature. Gini impurity is a measure used in decision tree algorithms to quantify the level of impurity or disorder within a set of data. It calculates how often a randomly chosen element from the set would be incorrectly classified. Gini impurity is used to guide the splitting of nodes in decision trees, aiming to minimize impurity and create more homogenous subsets. A relevant feature will reduce the impurity at large. The features with the highest Gini importance or mean decrease impurity values then, are considered to be the most important for the classification task. Thus, features with higher Gini importance have a larger impact on improving the overall homogeneity of the decision tree. In a similar way, the **layer-wise relevance propagation** (LRP) works by propagating the relevance score backward through the NN, layer by layer, using a set of rules that ensure that the total relevance is conserved at each layer. The relevance scores are then visualized to identify which features are important for the prediction and how they are combined by the network. LRP can help to explain the reasoning behind the network's prediction and identify potential sources of error or bias. Fleetwood, et al. [33] also used this in combination with restricted Boltzmann machines (RBM), autoencoders and even an MLP classifier by training them to identify states and later checking their LRP values for each feature. Although these methods have been able to infer molecular insights from MD data [33] as previously mentioned, the nature of these datasets used renders the understanding somehow challenging for the user. When no prior knowledge is available to guide the creation of input features, the authors recommend using atomic coordinates. Predictions are accurate when using atomic coordinates, yet they are difficult to interpret and the relationships between the different relevant features are complex and non-linear. Additionally labelled data is also used to train a random forest (RF) classifier to distinguish between frames from a ligand-bound and ligand-unbound within the same dataset. The states studied in the study are explored from different trajectories, one of them using spectral clustering to identify the states. This approach, while still useful, also adds complexity to the understanding of how the features affect the outcomes.

## 1.5 Motivation and Overview

Since this work contains two different projects related, but with different motivation and output, I will introduce the motivations and an overview of the work for each of them in the following subsections.

### 1.5.1 ML for understanding biomolecular events

Reflecting back at the ideas explored during sections 1.4.3 and 1.4.4, there is a current need for development towards **the understanding of high-dimensional data** coming from MD simulations. For instance, in cases where one generates vast amounts of data, such as macromolecular complexes, making it challenging to visualize and analyze, making it worse in the case of intrinsically disordered proteins [35]; or cases where one desires to construct accurate predictive models for certain activity using relevant features [36]. This is of crucial importance in the context of drug discovery. Recently, there has been a huge increase in the development of virtual screening methods, allowing for virtual screenings of thousand of compounds to study their binding affinity. This situation renders at the highest priority novel methods for analyzing huge amounts of data in an aoutmated and robust way, in order to identify patterns and gain insights. Data-driven discovery for molecular interactions, functional motifs or conformational states can also be simplified [37] using ML methods. Ideally, in all of these cases the end goal would be to develop an automated analysis that can handle the high-dimensional data as MD simulations are bound to involve larger and sophisticated systems, and scale up in the future. All of these require interpretable ML models that offer insights into the physical or chemical basis of their observed behaviour. Although semi-automated approaches using atomic coordinates have been developed [33, 37, 38] as a first base approach, they do not allow to delve in the relevance of the features too deep, due to their complexity or their nature. Both relevance between features themselves and within the system are difficult to grasp. There is also a need for fully automated procedures with higher resolution on the descriptors affecting the system directly, i.e., internal coordinates **more interpretable** such as distances and angles. This is specially true for protein-ligand complexes, when one wishes to evaluate the relevance of the active site's protein residues interactions with the ligand more effectively and in a straight-forward way. At the same time, the long-range interactions between ligand and protein are also known to be relevant during binding/unbinding. Not only that, but looking only at the active site interactions does not provide a full picture of the trajectory of the ligand and the major contacts and interactions during entry or departure from the binding site (namely, loop movements, allosteric effects, conformational changes, etc).

Consequently, there is still room for developments on the strategies for **interpreting MD data in a fully automated way** despite previous efforts [33, 37, 38]. In this study I attempted to bring forward the field and explore new methodologies as well as improve upon existing ones (see Fig.1.11). I will explore the usage of ML models in the interpretation of ligand unbinding trajectories. Moreover, I will use trajectories starting from the transition state of the unbinding process and will train the ML methods to classify between the two end states of the simulations (bound and unbound, IN and OUT), but using only early on data near the transition state, which turns the task into a forecast or prediction problem. That is, data impossible to classify by simple methods or by manual check. This then will allow us to identify the crucial steps in guiding the system towards each of the outcomes. We call this the Machine Learning Transition State Analysis (MLTSA). The protocol pinpoints the relevant features for classification on the inputted training dataset, thus, enabling to assess different internal coordinates at different levels depending on the information used. From atomic distances between protein and ligand, to protein-protein and even water-ligand distances. This allows for creating an automated pipeline for screening internal coordinates, which with little to no prior knowledge of the system under study was able to pinpoint already identified relevant residues, hint at new ones, suggest relevant atoms of the ligand studied and even suggest allosteric effects. All of this contributes towards the understanding of complex protein-inhibitor unbinding events, with focus on the transition state interactions at the molecular level, which are useful in understanding the residence time of the drug. Understanding the main transition state interactions at the molecular level allows researchers to lower or raise the ligand's unbinding energy barrier by modifying their chemical properties, thus aiding in drug design efforts.

### 1.5.2 Problem-tailored replica exchange

Coming back to a much more fundamental problem, a perfect solution for **the sampling problem** discussed in section 1.2, is not available yet. The amount of flavours on enhanced sampling techniques reflect back the need for an all-in-one solution to the sampling problem [14]. However, while that solution has yet to come, and researchers usually pick the most suitable method for the problem at hand. As the biophysical knowledge grows, additional specific problems arise from the recent advancements in the field; namely, new available structures for multimeric proteins, complex pathways and protein-protein relationships, etc. One of these examples is the relevance of metal ions in **metalloproteins**, which is irrefutable [39]. Moreover, they are crucial to assemble **active site geometries** and catalysis [40, 41]. They are necessary for the functioning of the body and are found throughout all of the phosphate enzymes crucial for life. Although the resolution of crystal structures is increasing, there is a huge knowledge gap on correct metal ion coordination in active sites. Not only that but the correct geometry is often not the one adopted during crystallization; additionally, the presence of other ligands or allosteric effects may alter this coordination, rendering docking calculations not so accurate. Moreover, the ever-changing nature of proteins can also mean metal ions come and go from the active site (transient ions) and adopt different geometries depending on the state of the protein. That is, having the possibility to have a different, yet crucial, geometry arrangement necessary to visit a **catalitically competent conformation**. Adopting this conformation, especially for complex systems where many different parts are involved, such as multimeric enzymes with DNA/RNA and other substrates regulating, may take long timescales up to milliseconds due to major rearrangements.

Most existing force fields are generally incapable of describing the interactions between metal ions and proteins accurately. Most of the problem comes from the fact that the ion is treated as a charged ball representation and its interactions with other molecules are Van der Waals potentials. This has been found problematic, for example on studies of a zinc protein when comparing ab initio and force field based calculations [42]. The force field overestimated the interaction between the ion and the negatively charged residues in the protein, showing that non-polarizable force fields cannot reproduce the interactions. Although previous efforts to tackle this problem have been proposed, most of them are built for specific cases, most of them involve a specific change in the force field parameters (i.e. Lennard-Jones potential) or how they are accounted for [43, 44]. Currently, ML methods based on NN potential models have become increasingly popular due to their efficiency and accuracy [45]. However, these are still constrained in terms of the training data which might be scarce for big and unexplored systems such as CRISPR Cas1/Cas2 and they have been applied to smaller protein systems so far.

Having all of this in mind, there is a need for developments in enhanced sampling schemes to aid in the study of metalloproteins and similar systems. In order to aid in this problem, I developed a **novel flavour of replica exchange**, more focused on the atomic charges (Atomic Charge Replica Exchange Molecular Dynamics or ACHREMD) (see Fig.1.11). While this is less advanced than a NN potential, it is more general and applicable to any system size and complexity. This method allows to flatten the free energy landscape by tampering with the atomic charges of relevant atoms. This can be applied to explore the association/dissociation of complexes and find stable conformations for active site geometries. It is a unique approach towards a better utilisation of the current methods while the development of novel methods is on the way. Its simplicity and scalability allows for quick exploration even in big protein systems.

Ultimately, Fig.1.11 summarizes the motivations that guided this thesis and give an overview of the expected projects.

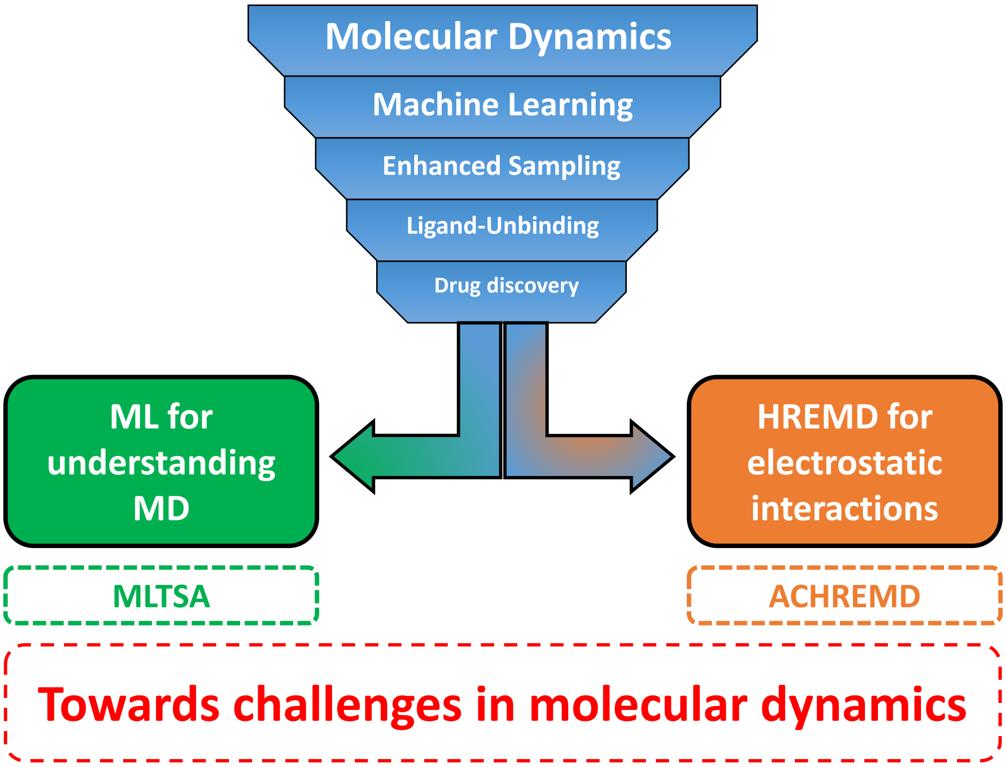


**Figure 1.11:** Visual summary of the motivations and overview of this thesis. Although it is applied on ligand-unbinding, MLTSA is designed to be used in any state-defined problem, including but not limited to other simulated biomolecular events. ACHREMD is most useful for electrostatic interactions, best suited to flatten the profiles and sample rare events such as dissociations and major rearrangements.

## Chapter 2

# General Methods

This chapter includes the theoretical/methodological information for the different techniques applied during the making of this thesis. The first section explains more in detail the principles of MD simulations, which is at the core of the study, followed by the two project's methods sections. The second section explains the methods necessary/developed for the MLTSA approach and the third section explains the methodology behind replica exchange MD and the development behind our ACHREMD version.

## 2.1 Molecular Dynamics Simulations

This section will explore the basis of MD and the steps involved in a typical MD protocol. This will include brief explanation for relevant concepts of the methods involved. MD is a simulation method involving the calculation of the positions, velocities and forces of each atom in a molecular system over time and using this information to predict the behaviour of the system over a range of time scales. Atoms are usually represented as points in three-dimensional space and bonds are represented as springs.

### 2.1.1 Basis of MD

#### Equations of motion

The **Newtonian equations of motion** are the cornerstone of MD simulations. These equations relate the forces acting on a particle to its acceleration, and provide a mathematical description of how forces determine the motion of particles in a system. The equations are expressed as $F = m * a$, where $F$ is the force acting on a particle, $m$ is its mass, and $a$ is its acceleration.

The **force** $F$ acting on a particle is a vector quantity, meaning it has both magnitude and direction. In the context of molecular dynamics simulations, this force is the cumulative effect of various interactions that a particle experiences due to other particles, bonds, electrostatic forces, etc. The force vector represents the combined influence of all these interactions on the particle. **Mass** $m$ is a scalar quantity that represents the amount of matter in the particle. It determines the particle's resistance to changes in motion (inertia). Heavier particles require more force to achieve the same acceleration as lighter particles. **Acceleration** $a$ is a vector quantity that represents the rate of change of velocity of the particle. It's the result of dividing the net force by the particle's mass ($a = F/m$). Acceleration determines how quickly the particle's velocity changes in response to the applied force.

In MD simulations, the motion of each atom or molecule in a system is described using a set of coupled equations of motion. These equations are obtained by considering the forces acting on each atom due to the intermolecular interactions present in the system. The forces acting on each atom are calculated using a chosen potential energy function, which represents the energy of the system as a function of the atomic coordinates.

Summing up, the interatomic force for particle $i$ would be described as

$$\vec{F_i^t} = m_i \cdot \vec{a_i} \tag{2.1}$$

where $m_i$ is the mass of particle i, $\vec{a_i}$ is its acceleration, and $\vec{F_i^t}$ is the total force acting on it. Then, the total force on particle $i$ is calculated as the sum of all forces exerted by other particles in the system, for example $j$. Then, including interatomic forces and any external forces, the force on $i$ would be

$$\vec{F_i^t} = \sum_j \vec{F_{ij}} + \vec{F_i^{ext}} \tag{2.2}$$

where $\vec{F_{ij}}$ is the interatomic force between particles $i$ and $j$, and $\vec{F_i^{ext}}$ is any other external force. The interatomic forces are typically calculated using interatomic potentials, such as the Lennard-Jones or Coulomb potentials. The specific form of these potentials depends on the nature of the interactions between the particles (e.g. van der Waals or electrostatic). The set of equations specifying these interactions is called a **force field**.

## Force Fields

Force fields can be either empirical or theoretical, meaning they are either derived from experimentally determined properties or derived from first principles based on quantum mechanics. Empirical force fields, such as AMBER, CHARMM, and GROMACS, are based on fitting parameters to experimental data, while theoretical force fields, such as quantum mechanical force fields, are derived from ab initio calculations [46]. Examples include ab initio molecular dynamics (AIMD) and Car-Parrinello molecular dynamics (CPMD).

The potential energy function ($E_{Total}$) used in MD simulations typically takes the form of a sum over all pairs of atoms in the system, with each term representing the energy of interaction between a given pair of atoms for both bonded ($E_{BondedTerms}$) and non-bonded ($E_{Non-BondedTerms}$) interactions. Fig.2.1 contains the different terms for the different illustrated interactions that may take place, following the same order as in the figure, these are:

**Bonded Terms**

- $E_{bonds}$ : To calculate the energy from the bonds, modelled with harmonic potentials having parameters for the ideal values of bond lengths ($r_{eq}$) using a force constant ($k_r$).

- $E_{angles}$ : Representing the energy for the ideal angle value ($\theta_{eq}$) with a harmonic potential using a force constant as well ($k_\theta$).

- $E_{dihedrals}$ : Dihedrals are described with a Fourier series, with a force constant $k_\phi$ for an angle $\phi$, $n$ for the multiplicity of the potential (depending on their minima) and $\gamma$ for the phase offset or angle at which the minimum occurs.

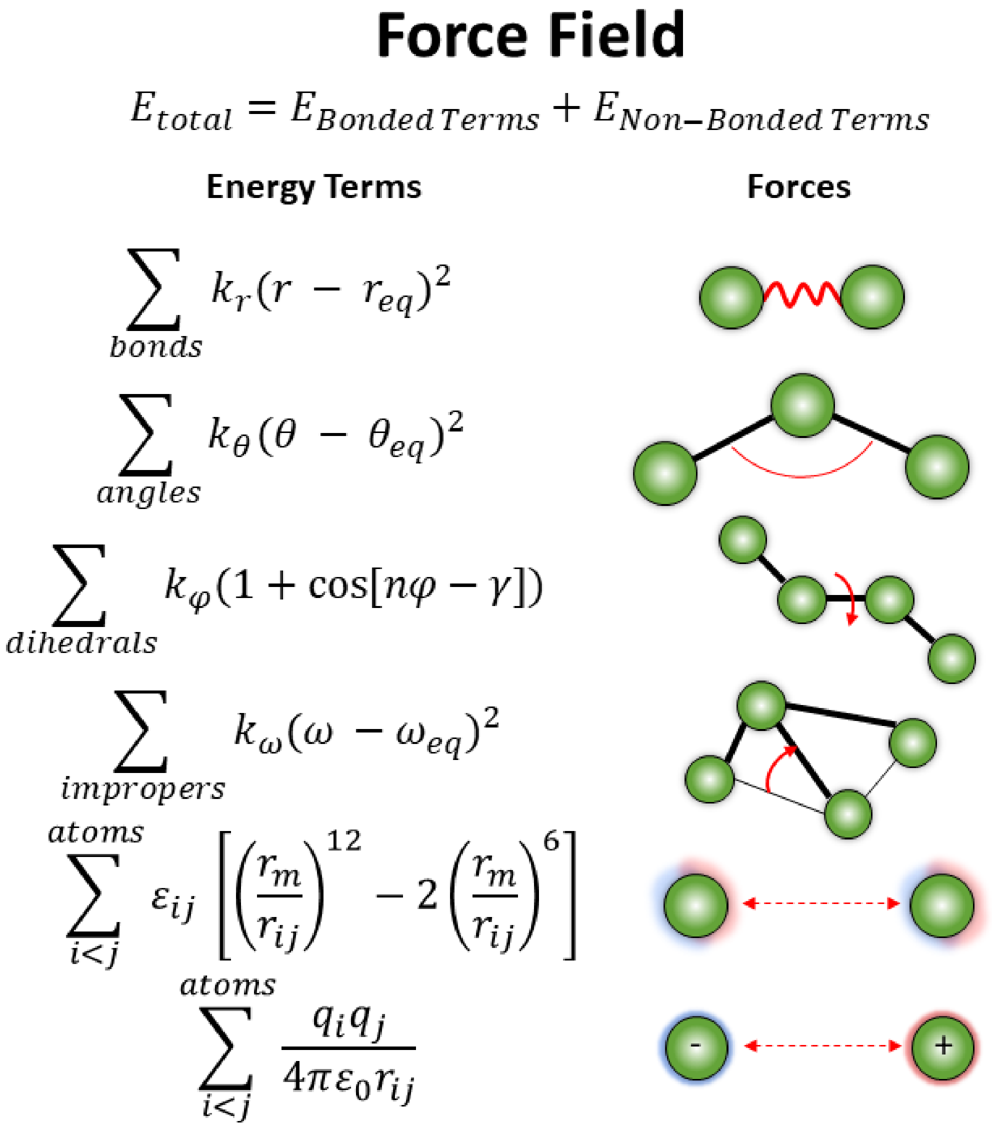

Force Field
$E_{total} = E_{Bonded\,Terms} + E_{Non-Bonded\,Terms}$
Energy Terms
Forces
$\sum_{bonds} k_r(r - r_{eq})^2$
$\sum_{angles} k_\theta(\theta - \theta_{eq})^2$
$\sum_{dihedrals} k_\varphi(1 + \cos[n\varphi - \gamma])$
$\sum_{impropers} k_\omega(\omega - \omega_{eq})^2$
$\sum_{i<j}^{atoms} \varepsilon_{ij} \left[ \left( \frac{r_m}{r_{ij}} \right)^{12} - 2 \left( \frac{r_m}{r_{ij}} \right)^{6} \right]$
$\sum_{i<j}^{atoms} \frac{q_i q_j}{4\pi\varepsilon_0 r_{ij}}$
-
+

**Figure 2.1:** Illustration of the most common interactions between atoms in force fields and the energy terms describing them. Atoms are represented in green and the relevant interactions in red. Black arrows represent bonds.

- $E_{impropers}$ : Term defined for aromatic and $sp^2$ atoms to maintain their planar structure. $\omega$ is the angle defined for atoms ABCD, where D is in bond with B not C. These are modelled with a harmonic potential as well and a $k_\omega$ to an ideal value $\omega_{eq}$.

### Non-Bonded Terms

- $E_{VdW}$ : Van der Waals interactions are accounted for with this term, that can be described by a variety of function forms, the Lennard-Jones is the most common one where $r_m$ is the optimal distance, $r_{ij}$ the actual distance, and $\varepsilon$ is the depth of the potential energy well.

- $E_{Coulomb}$ : One of the most contributing ones, the electrostatic interaction, which is accounted for with this term, having $q_i$ and $q_j$ as the point charges, $r_{ij}$ representing the distance between the atoms and $\varepsilon$ the dielectric constant.

Note that the contemporary versions of force fields may contain more terms accounting for corrections. The force acting on each atom is then obtained as the negative gradient of the potential energy function with respect to the atomic coordinates, and the resulting set of equations is **solved numerically** using an integration algorithm such as the Verlet or the Leapfrog algorithm. The **timescale problem** mentioned in 1.2 arises because processes, like unbinding, biomolecular folding or large conformational changes, occur on longer timescales than can be directly simulated due to computational limitations. Short time steps ensure numerical stability but limit the simulation to short timescales, making it difficult to capture slow processes. On the other hand, using larger time steps to simulate longer timescales risks introducing inaccuracies and destabilizing the simulation.

The numerical solution of the equations of motion allows us to obtain the coordinates of each atom as a function of time, allowing us to study the dynamic behavior of the molecular system. This includes the calculation of structural and dynamical properties such as the time-dependent pair distribution function, the mean square displacement, and the time correlation functions.

In essence, while force fields provide equations, solving them numerically encounters timescale limitations. Strategies like **enhanced sampling** or **coarse-graining** help bridge the gap between fast numerical integration and slower processes of interest in molecular dynamics simulations. These methods aim to speed up rare events by biasing simulations or reducing the level of detail, allowing longer timescales to be explored.

### Temperature and Pressure

In molecular dynamics (MD) simulations, there are three common ensembles used to control the thermodynamic conditions of the system: NPT, NVT, and NVE [47].

- **NVT** (Constant Number of Particles, Volume, and Temperature) is an ensemble in which the number of particles, volume, and temperature of the system are held constant. The temperature is controlled using a thermostat, and the volume is kept constant by not allowing the simulation box to change size.
- **NPT** (Constant Number of Particles, Pressure, and Temperature) is an ensemble in which the number of particles, pressure, and temperature of the system are held constant. The temperature is controlled using a thermostat, and the pressure is kept constant using a barostat.
- **NVE** (Constant Number of Particles, Volume, and Energy) is an ensemble in which the number of particles, volume, and total energy of the system are held constant. No thermostat or barostat is used in this ensemble, which makes it ideal for studying systems that are not in thermal and pressure equilibrium.

The choice of ensemble depends on the goals of the simulation and the physical conditions of the system being studied. For our systems of study, the choice for simulation ensemble is an NPT, due to the nature of the experiments. One wants to be able to track the energy through the process to be able to calculate the free energy profile of the events.

**Thermostats** . The temperature of the system is a measure of the average kinetic energy of the particles in the system. In MD simulations, the temperature is controlled through the use of thermostat algorithms. The most common thermostat algorithms are the Nosé-Hoover thermostat and the Berendsen thermostat. These algorithms work by adjusting the velocity of the particles in the system, thus controlling the temperature of the system [47].

**Barostats** . The pressure of the system is a measure of the force exerted by the particles in the system on the walls of the simulation box. In MD simulations, the pressure is controlled through the use of barostat algorithms. The most common barostat algorithms are the Berendsen barostat and the Parrinello-Rahman barostat. These algorithms work by adjusting the size of the simulation box, thus controlling the pressure of the system [47].

Thermostat algorithms and barostat algorithms are crucial in MD simulations as they ensure that the temperature and pressure of the system remain constant and uniform throughout the simulation. This helps to prevent the system from deviating from the desired temperature and pressure conditions and ensures that the simulation results are accurate and meaningful [47].

### 2.1.2 MD Simulation Protocol

In a typical MD protocol, a pipeline of several crucial steps has to be followed in order to successfully simulate a given system. This usually ranges from the setting-up step to the production run step. The described protocol here follows that of the CHARMM-GUI solution builder.

#### System preparation

To start with, the 3D structure of the protein of interest has to be determined (through X-ray crystallography, NMR or Cryo-EM), or modelled (through homology modelling), obtaining a set of atomic coordinates, usually a *.pdb* file with the cartesian coordinates (*XYZ*).

The next step after reading in the initial coordinates, is solvating the system with water molecules, which can be placed following different conventions. In the systems for this study the molecules were placed using a Monte-Carlo placing. In this step, the salt concentration has to be set as well using different cations and anions to emulate that of the environment one wishes to simulate, or neutralize the system [47].

After that, the molecular topology is built which defines the bonded and non-bonded interactions between the atoms in the system. This involves assigning partial atomic charges, types of bonds and angles and specifying non-bonded interactions.

### Periodic Boundary Conditions

Once the required molecular topology for simulation is built, the periodic boundary conditions (PBCs) are applied based on the system shape and size. PBCs are used to mimic an infinitely large system by repeating the simulation box in three dimensions.This creates a seamless environment where molecules can interact with their own periodic copies, enabling the simulation to explore various configurations and interactions, effectively sampling a broader range of possible molecular arrangements in space. The creation of PBCs involves defining the dimensions of the simulation box, which can be either cubic or orthorhombic, and determining the box vectors. The box vectors define the orientation and size of the simulation box and must be chosen such that the system fits comfortably within the box while the interactions between the periodic images are negligible.

Once the box dimensions and vectors have been defined, the atomic coordinates of the system are then wrapped into the simulation box, such that all atoms are within the simulation box and are separated by a minimum image convention. This convention ensures that each atom interacts with only one periodic image of its nearest neighbors, avoiding double counting of interactions.

In order to calculate long-range electrostatic interactions, the particle-mesh Ewald (PME) method is often used. The fast Fourier transform (FFT) grid information is essentially a 3D grid that is used to store the electrostatic potential of the system. The size of the grid is determined by the number of grid points in each dimension and is typically chosen such that the grid spacing is smaller than the cut-off distance of the non-bonded interactions. Thus, the FFT grid information is then used in the PME method to perform a 3D FFT on the electrostatic potential. This transforms the data from real space to reciprocal space, where the long-range interactions can be more efficiently calculated. The result of the FFT is then used to calculate the interaction energy between the periodic images.

### Initialization and Energy Minimization

A typical MD protocol involves the calculation of the initial energy of the system to compute the new forces and, through numerical integration predict the next position, velocities and forces in an iterative fashion for a given number of steps.

Energy minimization is a critical step in many MD simulations, as it helps to remove any artificial or random fluctuations in the initial atomic positions, and to prepare the system for the dynamics simulation. The goal of energy minimization is to generate a stable, representative starting configuration that is consistent with the conditions specified by the simulation parameters (e.g. temperature, pressure, etc.). The principle states that the simulation should strive to minimize the potential energy of the system, with the goal of finding the set of atomic coordinates that results in the lowest possible value of the potential energy. To achieve this, numerical optimization algorithms are used to iteratively adjust the atomic coordinates until the potential energy reaches a minimum value.

The most commonly used algorithm for the minimization of MD simulations is the steepest descent or gradient descent algorithm. This algorithm involves iteratively moving the system towards the direction of the negative gradient of the potential energy, until a local minimum is found. Another algorithm that is sometimes used is the conjugate gradient algorithm, which is more efficient than the steepest descent algorithm for systems with large numbers of degrees of freedom. The choice of minimization algorithm depends on the system being studied and the computational resources available. In general, the steepest descent algorithm is sufficient for most systems, but more complex algorithms may be necessary for systems with large numbers of degrees of freedom or complex energy landscapes.

### Equilibration

The equilibration step is a crucial pre-processing step that prepares the system for the production simulation. The goal of equilibration is to bring the system to a state of thermodynamic equilibrium, where the temperature, pressure, and particle distribution have reached their steady-state values. It typically involves two phases: an initial heating phase and a final equilibration phase.

In the initial heating phase, the system is rapidly brought up to the desired temperature by applying a thermostat. This phase is usually done rapidly in order to avoid any long-term effects that could alter the properties of the system.

In the final equilibration phase, the temperature, pressure, and particle distribution are allowed to reach their steady-state values over a longer period of time. The system is subjected to both a thermostat and a barostat to control the temperature and pressure, respectively. This phase is run for a sufficient duration to ensure that the simulated system has effectively reached a state of thermodynamic equilibrium, where its macroscopic properties remain relatively constant over time, reflecting the conditions under which the system is meant to be studied. This will result in a state of thermodynamic equilibrium, which is a stable and representative state of the system to start the simulation from.

Depending on the system, the final equilibration phase may have to be run for longer steps to ensure the relaxation of the systems. In MD simulations of proteins, constraints are often applied to the system to prevent bond stretching or bond angle bending that would result in unrealistic deformations of the protein structure. The deformations can happen in response to the sudden change in temperature or pressure, or during the rearrangement of the bulk. These constraints can be implemented using algorithms such as the SHAKE or LINCS algorithms, which maintain the bond lengths and angles within specified limits. Even after running longer equilibration steps, the amount of steps to relax the solvent may not be sufficient, it is usually recommended to follow up equilibrations with unconstrained production runs to fully relax the system and start considering data after the additional unconstrained production and consider it additional equilibration.

### Production

The production run in molecular dynamics (MD) simulations is the main simulation stage. The goal of the production run is to obtain representative and statistically meaningful data that can be used to study the physical properties of the system being modeled.During the production run, the system is allowed to evolve over time under the influence of the forces between the particles, and the temperature and pressure are maintained using a thermostat and barostat, respectively. The length of the production run depends on the desired accuracy of the simulation and the physical properties of the system being studied. Typically, the simulation is run for several nanoseconds or longer, with snapshots of the system taken at regular intervals for analysis.

It is at this stage that most techniques for enhanced sampling and other methods take place. This is the foundation that ensures a proper simulation experiment

## 2.2 MLTSA

This section will explain the methodology and techniques necessary to perform the MLTSA analysis on time data series. This has been applied on both biologically relevant data and toy model data. Being an *analysis*, it needs data to be used on as well as several additional steps to be performed and it can be used on a side as a complement of other approaches. However, in the context of this thesis, it will be treated as the main goal and the techniques needed to achieve it are under its umbrella. In order to successfully use MLTSA it is necessary to have both:

- Several trajectories from a time series that starts in the same state and it leads to different outcomes or states. This is not restricted to only 2 outcomes, but 2 and 3 outcome states are explored in this study. This could work with both, physical events or any other time-dependent event which starts at the same time and ends with a different outcome.

- A way to categorize each outcome to assign a label for each series and build the training data. This could be either from the input data, or using a different quantity or non-numerical variable not included in the training data, which is the most suited to get the most information out of the approach.

In the context of MD simulations, however, this could refer to, but it is not restricted to, a TS. Having a transition state (TS) structure allows the exploration of the reactant (A) and product (B) states starting several independent simulations also called *Downhill Simulations*. Having both outcomes A and B, and tracking CVs (see section 1.3.1 for reference) from the downhill trajectories is what is needed for the approach, which in return yields very insightful information about the internal coordinates under study, in the form of feature relevance for the outcomes. This enables highlighting the main driving factors at different levels (interatomic distances, cartesian coordinates, angles and other features). Although it sounds straightforward, this approach requires several steps to accomplish this, in specific for our ligand-unbinding cases of study it involves:

1. Obtain a first unbinding trajectory using our unbinding protocol.
2. Refine and optimize the unbinding path (see section 2.2.1 for the details on our case).
3. Explore the reconstructed free energy profile structures in order to approximate the TS of the unbinding (these come from string windows in our case).
4. Generate several downhill simulations starting from the identified TS to sample well enough.

Having that in mind, this section will focus on the steps necessary for obtaining the MD simulation data, additionally it includes the ML models used and the feature analysis techniques needed to perform the full analysis that has been done on all types of data.

### 2.2.1 Ligand Unbinding

The approach used in this thesis is described in the unbinding protocol of [1], published prior to the writing of this thesis. This section will explain the different parts involved in this protocol, which are necessary to apply MLTSA in ligand-unbinding downhill trajectories.

#### Unbinding Protocol

The unbinding protocol is an iterative algorithm that allows one to obtain optimized ligand unbinding paths from a bound crystal structure, as well as recover the free energy profile describing it. Note that this has been done in proteins so far but it is not restricted to them. Following the flowchart in Fig. 2.2, after building the system from the bound crystal structure, a first exploratory simulation identifies the initial interactions. These initial interactions, namely interatomic distances, allow us to define a first set of CVs to bias and start pushing the ligand outside of the pocket.

Using a $d_{in}$ as interaction cut-off, the distances between ligand and protein heavy atoms smaller than $d_{in}$ will generate a single one-dimensional CV as the sum of these distances. This main CV will be used for iteratively biasing the simulations until the ligand totally exits the binding pocket. However, as the ligand exits, new interactions are formed and old interactions vanish and are no longer needed to bias. To account for it, at the end of each iteration the biased trajectory is analyzed and novel interactions within $d_{in}$ are added, as longer than $d_{out}$ interactions are removed from the main CV. This provides a more natural approach to ligand dissociation than other similar methods which constantly bias the interactions between protein and ligand. Note that $d_{in}$ and $d_{out}$ are cutoffs for considering adding or removing interactions from the main CV, which is used for biasing ($V^n$) as described in eq.2.3.

The bias that is applied to this main CV has the form of a harmonic restraint

$$V^n = \frac{1}{2}k(D^n - \sum_{i=1}^{M^n} d_i^n)^2 \tag{2.3}$$

Where $D^n = D_0^n + M^n$, $M^n$ is the sum of the number of distances $d_i^n$. Here one aims to reach the target value $D^n$ for the main CV starting from the initial value $D_0^n$. The targeted value will be reached progressively within the given simulated steps with a $k$ force constant. Note this terms are different than $d_{in}$ and $d_{out}$ which are cutoffs for including distances in the main CV or not.

If a distance during the last 5ns of the trajectory exceeds $d_{out}$ or its variance exceeds $d_{var}$, the distance will be removed from the main CV for the next iteration and will no longer be biased. Similarly, such loosely interacting atom pairs have higher distance fluctuations, and thus this weak interaction does not need to be included in the bias. To reduce the number of interactions between the ligand and the protein to bias, and to remove redundancies, one has to combine atoms that are part of an equivalency group where a rotational degree of freedom can interchange the atoms from one to the other.

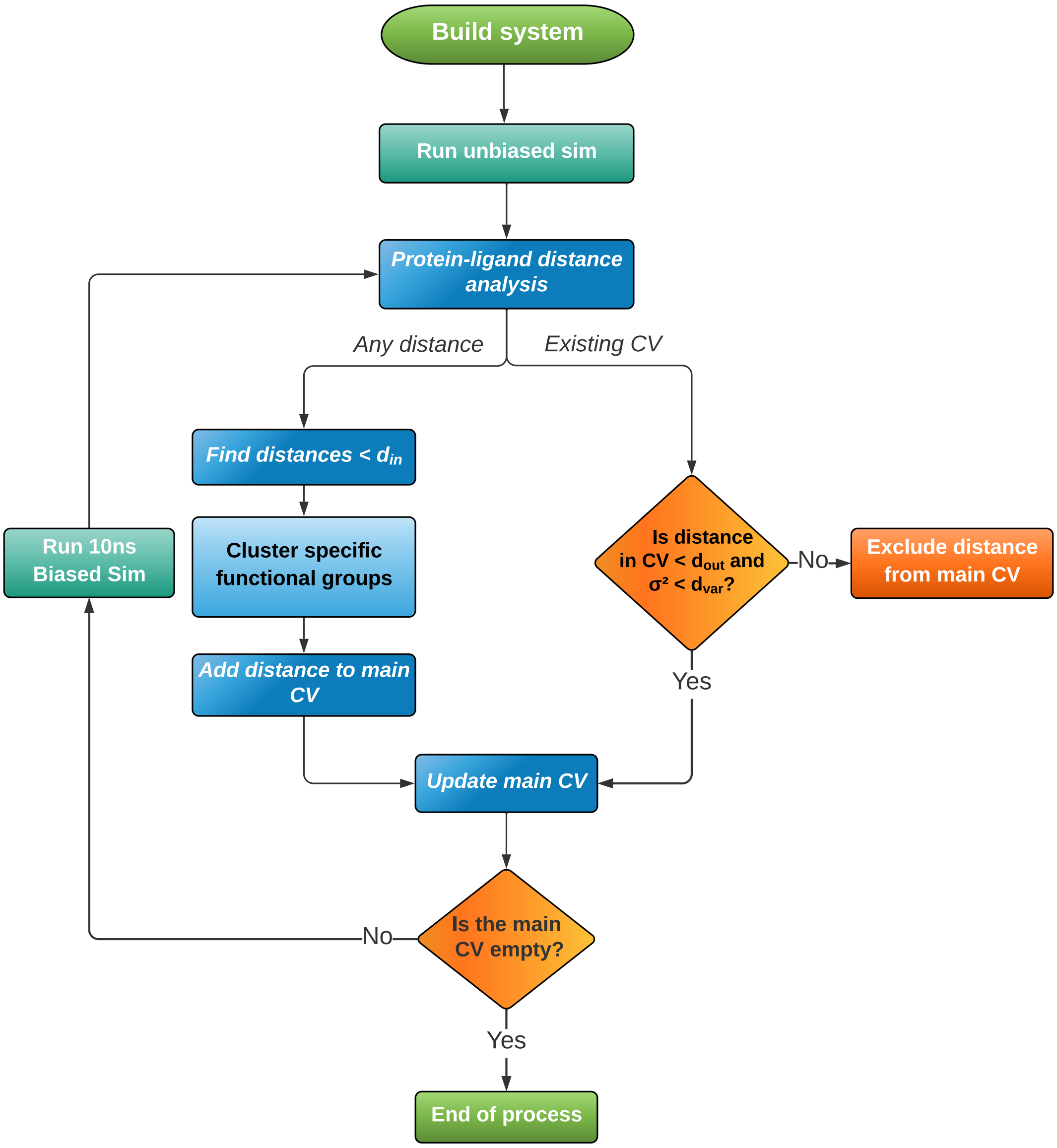

Build system
Run unbiased sim
Protein-ligand distance analysis
Any distance
Existing CV
Find distances < $d_{in}$
Cluster specific functional groups
Add distance to main CV
Is distance in CV < $d_{out}$ and $\sigma^2 < d_{var}$?
No
Exclude distance from main CV
Yes
Update main CV
Run 10ns Biased Sim
Is the main CV empty?
No
Yes
End of process

**Figure 2.2:** Flowchart illustrating the steps for the unbinding protocol.

### Finite-Temperature String Method

Once the initial unbinding path has been obtained, a subsequent finite-temperature string method, also known as *string method* [48], is run to obtain an optimized free energy pathway of the mechanism. This method is used for finding the minimum-energy path (MEP) between two stable states in a multi-dimensional potential energy surface. It is based on the idea of generating a string (or curve) in the high-dimensional space that connects the two stable states, and updating the string at each iteration so that it moves towards the MEP. The MEP is defined as the path of minimum energy that connects the two states. The string-like representation of the molecular configuration evolves towards the global minimum over time. It evolves according to an iterative minimization scheme that updates the string configurations using gradient-based optimization techniques. The finite-temperature aspect of the method refers to the use of MD simulations to sample the energy landscape, allowing the string to escape from local minima and explore the entire energy surface. The method has been applied to a wide range of molecular systems, including proteins, peptides, and small organic molecules, and has shown great success in finding the global minimum conformations of complex molecular systems [48, 49].

A string is initialized in the high-dimensional space, connecting the two stable states. This string is typically represented as a series of discrete points or nodes (*windows*). At each iteration, the nodes of the string are updated by moving them in the direction of the negative gradient of the potential energy surface. This results in the string gradually moving towards the MEP. After each iteration, the string is reparameterized to ensure that the nodes are evenly spaced along the path. This helps to prevent the string from becoming too dense in some regions and too sparse in others. The method includes a bias correction term to ensure that the string remains on the MEP even at finite temperatures. This correction term is based on the concept of force-bias, which is the difference between the force on a particle at a given point in space and the force that would be expected at that point based on the potential energy surface.

Applying this method iteratively until convergence can yield back accurate free energies for the unbinding path. In the context of the unbinding protocol, this method is applied using the starting bound structure and the final unbound structure as well as the obtained path during the protocol. This trajectory is then divided into windows (nodes) for the each of the distances that have been included in the main CV, building a string in the coordinate space. For each window and each CV a positional restrain is equidistantly placed along the initial fitted string, using a force constant. A high order polynomial fitting is applied using the average values for each CV to build the subsequent set of refined CV positions. This is done iteratively until the convergence of the free energy profile. Convergence is verified by ensuring that the maximal change of each CV between iterations is below 7%.

### Free Energy Profiles

After convergence of the string trajectories, the free energy profile from the unbinding path can be recovered despite the trajectories being biased by using a reweighting method. WHAM (Weighted Histogram Analysis Method) is a computational technique for the determination of free energy profiles and other thermodynamic properties from MD data [50]. The method allows for the estimation of the probability distribution of a given system parameter (e.g. the distance between two residues), taking into account the fluctuations in the simulation trajectory and the effect of different simulation conditions (e.g. temperature or pressure). The resulting free energy profile provides insights into the stability, conformations, and energetics of the system, which can be used to understand the system's behavior and make predictions about its behavior under different conditions. WHAM is commonly used in MD simulations and has been applied to various systems including proteins, lipids, and nucleic acids [15].

WHAM can handle biased data by weighting each data point in the histogram according to the bias energy. The bias energy acts as a correction factor that modulates the observed probability of the system. The corrected probability density function is then used to calculate the free energy profile. The weighting factor can be introduced into the WHAM equation in the following way, using the equation for the probability $P_i$ for a state $i$

$$P_i = \frac{n_i}{N} e^{-\beta(E_i - F_i)} \tag{2.4}$$

where $n_i$ is the number of observations in bin $i$, $N$ is the total number of observations, $\beta = \frac{1}{k_B T}$, $E_i$ is the average energy of the bin $i$, and $F_i$ is the corresponding free energy. The bias energy is incorporated as a correction factor in the calculation of the average energy $E_i$

$$E_i = \frac{\sum_{j=1}^{n_i} (E_j + V_b(x_j))}{n_i} \tag{2.5}$$

where $E_j$ is the energy of observation $j$, $V_b(x_j)$ is the bias potential for observation $j$ and $x_j$ is the collective variable value of observation $j$.

The binless implementation of the Weighted Histogram Analysis Method (WHAM) is an alternative method to the traditional binned implementation. It involves using a mathematical optimization algorithm, such as a maximum likelihood or minimum free energy approach, to directly determine the density of states without relying on a discretized representation. This can result in a more accurate and flexible representation of the density of states, as well as a reduced dependence on the choice of bin sizes and boundaries. However, it can also be computationally more intensive than the binned approach, and may require a more sophisticated optimization algorithm.

### Transition State Approximation

After recovering the profile for the unbinding path, an approximation of the TS is possible by interpolating the top of the energy barrier's reaction coordinate with the corresponding structure. In theory, a TS should be the configuration with the highest energy along the reaction coordinate path that has to be crossed in order to obtain a product from the reactants. Thus, starting new independent simulations from this high-energy state would yield, in theory, a 50/50 chance of ending in the reactant (bound) or product (unbound) states.

Following this idea behind the TS, one is able to approximate a TS structure by starting simulations using configurations from string windows around the corresponding high-energy point. Thus starting multiple simulations from string windows I was able to find the one yielding the closest 1:1 probability of going back to pocket or unbinding. Once the closest one is identified, it can be used to generate multiple trajectories to build the datasets needed for the MLTSA analysis.

### 2.2.2 Protein Datasets

In this sub-section the protocol for the generation of each type of dataset will be explained. Note that this methodology is shared between all systems studied and these are just details for the implementations. While it is recommended to follow the order of this datasets in order to succesfully apply the MLTSA analysis, it is not restricted to this alone. One may be focusing on the active site alone, or maybe only an event which is strictly protein-protein related. Additionally other types of datasets are also encouraged.

#### Closest Protein-ligand Distance

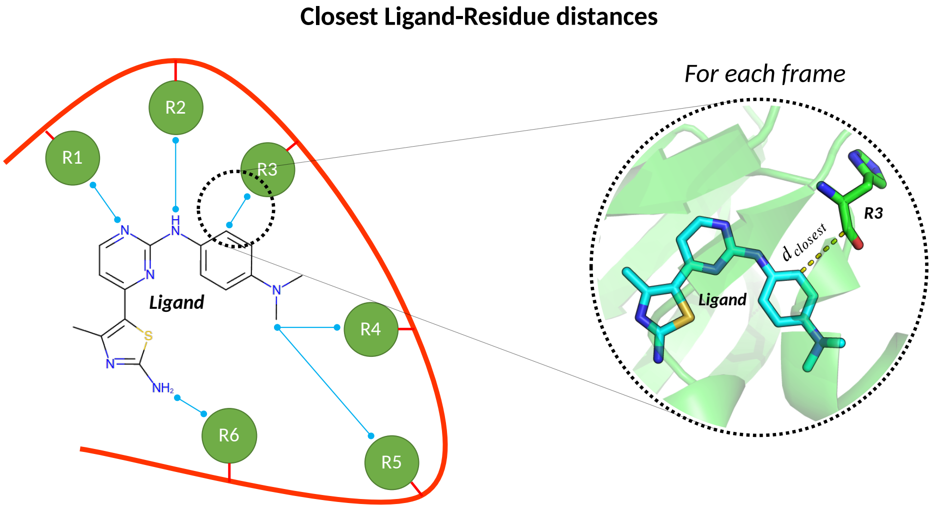


**Figure 2.3:** Illustration of the distances included in the analysis for an example *allres* dataet with the closest ligand-residue distances at each frame. Note that for every frame, the closest interatomic distance for each residue is calculated and added to the dataset.

To assess all protein-ligand contributions first without having to calculate every interatomic distance between them, a first **”closest protein-ligand”** dataset is created, also called *allres*. This dataset is a coarse picture of all residue distances with the ligand, by only taking into account the smallest distance between ligand and residue at each time, reducing this to only contacts. By tracking this through the trajectory, the *allres* dataset is created. Note that this is done not only for residues, it can also be done optionally for water molecules to try to evaluate their role or other entities such as ions, other ligands, cofactors, etc.

This is implemented by using mdtraj's `md.compute_contacts` and it uses the closest atom, heavy or not, for distance criteria [51]. That is, recalculating the closest atomic distance between a given residue and the ligand at each frame, and recording it's value to create the coarsed dataset.

### Protein-ligand Shell

The **Protein-ligand Shell** datasets correspond to interatomic distances of all atoms within a given distance cutoff from the ligand's atomic positions. That is, for example, all interatomic distances between ligand and protein residues within 5*Å* of any ligand's atom. Usually to cut down the number of distances, only distances between heavy atoms are added to the dataset. This is done by using mdtraj's `md.compute_neighbours()` implementation [51]. Please note that all atoms within a given *d* cutoff at the TS structure will be included in the dataset and its

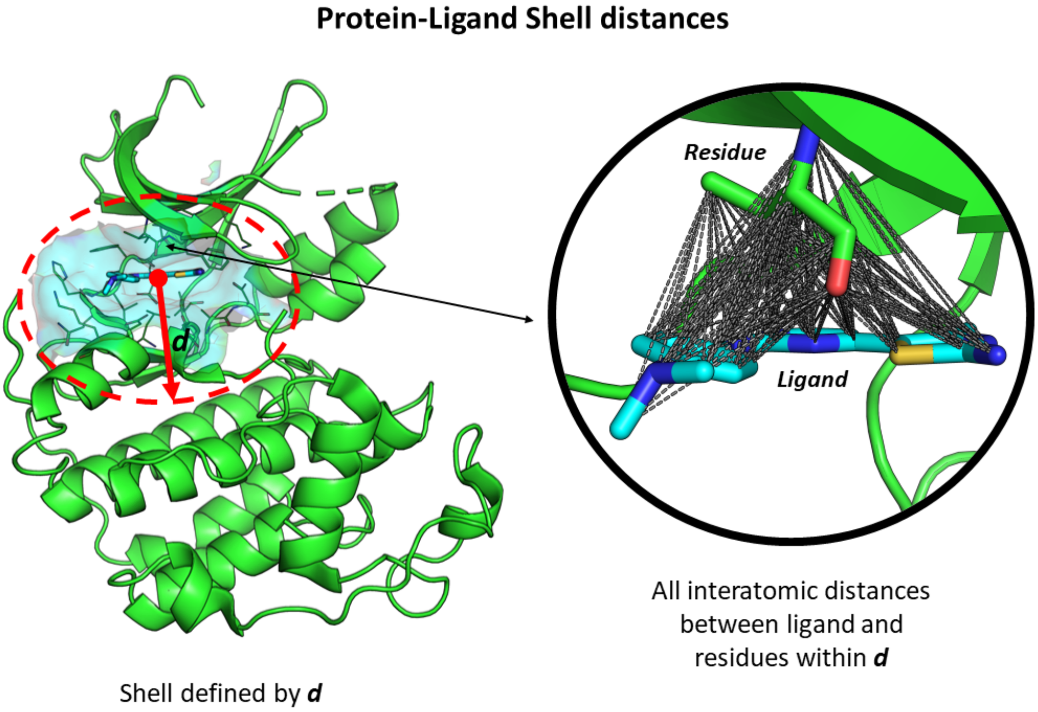

Protein-Ligand Shell distances
d
Residue
Ligand
All interatomic distances between ligand and residues within d
Shell defined by d

**Figure 2.4:** Diagram of the protein-ligand shell dataset construction. On the left, the shell defined by a given distance *d* cutoff, all protein residue atoms inside this shell will be used to calculate their interatomic distance to all ligand atoms and build the *protein-ligand shell* dataset.

distance at all times monitored.

### XYZ-PCA Features

To be able to assess all residue-residue interactions within the protein, the number of distances to assess would be overwhelming. As an attempt to identify relevant conformational changes and assess intraprotein interactions, the **XYZ-PCA dataset** was created.

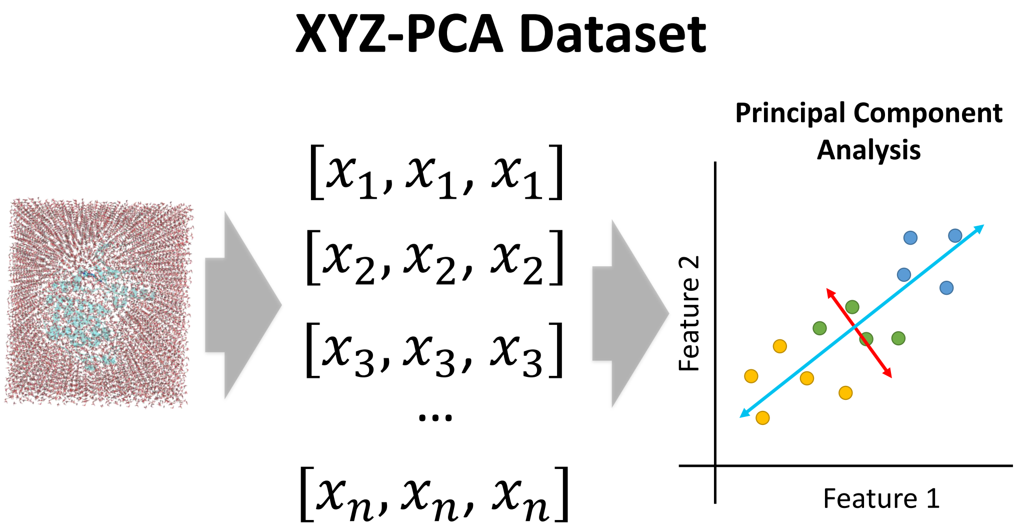


**Figure 2.5:** Atomic cartesian coordinates from the system are used to perform a dimensionality reduction with PCA. Then the PCA's top components are used as input features to create an *XYZ-PCA dataset* dataset

**Principal Component Analysis** (PCA) is a dimensionality reduction technique that is commonly used to analyze data. The goal of PCA is to transform the input features into a new coordinate system such that the first coordinate (i.e. the first principal component) captures the largest amount of variance in the data, the second coordinate captures the second-largest amount of variance, and so on.

Given a centered data matrix X of size NxD (N being the number of samples and D the number of features), the PCA algorithm computes the eigenvectors and eigenvalues of the covariance matrix

$$\Sigma = \frac{1}{N} X^T X \tag{2.6}$$

where $X$ is the input data matrix, $X^T$ is the transpose of X and $\Sigma$ is the covariance matrix.

The eigenvectors are denoted by $U$ and the eigenvalues by $\lambda$. The eigenvectors corresponding to the largest eigenvalues capture the most important patterns in the data, and thus form the basis of the lower-dimensional subspace.

The projected data matrix $Z$ then is obtained by transforming the data matrix X using the eigenvectors

$$Z = XU \tag{2.7}$$

where $Z$ is the projected data matrix, $U$ is the eigenvectors and $X$ is the original data matrix.

This projection of the data allows a dimensionality reduction from the original XYZ (3* $n_atoms$) coordinates, much more meaningful and gets rid of the noisy data. This can create as many projections as original features, however, the explained variance of each principal component (eigenvectors) is already indicative of the amount of new projections needed to avoid information loss. Taking just the amount of projections before the explained variance decays to 0, one can ensure there is no information being left on the projected data.

Thus, to create the XYZ-PCA dataset, the raw atomic coordinates of the protein can be concatenated and reduced to 1D projections by PCA to then be used in the MLTSA as well and gain insight on specific atoms/residues relevant to the outcomes. This can be done by looking at the specific component scores for each original input feature (cartesian), thus averaging to get the atomic relevance ($\overline{XYZ}$) and the residue relevance ($\overline{atoms}$). Doing this, one can interpolate the relevant features out of the MLTSA to the structural changes in the protein. More specifically, this can be translated to an XYZ movement depending on relevance, to see the most relevant areas in a more illustrative fashion.

### 2.2.3 Machine Learning Models

The methods and techniques described within this chapter are the ones used for the MLTSA analysis, which include neural networks and decision trees so far. A brief overview of the three models will be provided including architecture, parameters, and training procedure. Software used for training and testing will also be described.

### MLP

A **Multi-Layer Perceptron** (MLP) is a type of feedforward neural network that is widely used for supervised learning tasks such as classification, regression, and prediction. It consists of multiple layers of interconnected neurons, including an input layer, one or more hidden layers, and an output layer. As described in Fig.2.6 each neuron in the hidden and output layers computes a weighted sum of its inputs, passes the result through an activation function, and produces an output that is propagated to the next layer. In an MLP, the input is fed into the input layer, and the output is obtained from the output layer, as in Fig.2.7. The input values are propagated forward through the network, layer by layer, using a set of weights that are learned during the training process. The weights are updated using a backpropagation algorithm, which computes the gradient of the error function with respect to the weights and adjusts them to minimize the error. Let's assume that the input

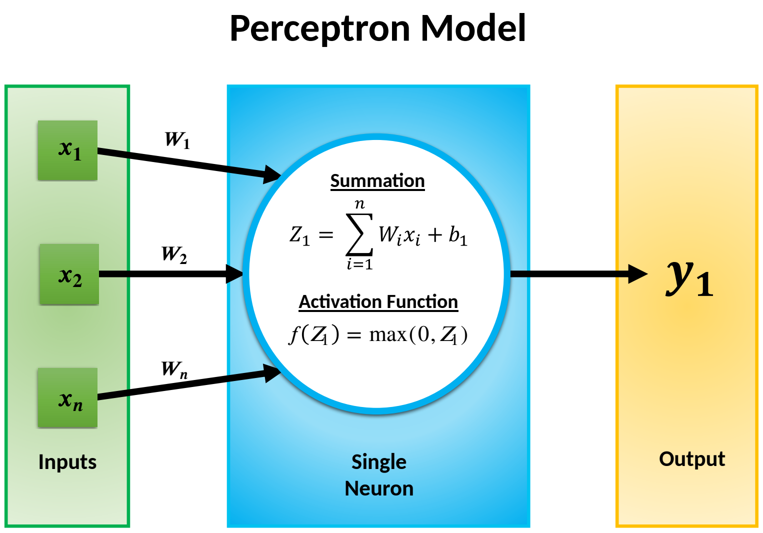


**Figure 2.6:** Representation of a simple perceptron diagram. The inputs (X) get multiplied by a weight (W) and summated in a neuron, then an activation function f() (in this case ReLU) produces the output value (y).

layer has $n$ inputs, the hidden layer has $m$ neurons, and the output layer has $p$ outputs. The weights between the input layer and the hidden layer are represented by the matrix $W$, and the bias term for the hidden layer is represented by the vector $b$. Similarly, the weights between the hidden layer and the output layer are represented by the matrix $V$, and the bias term for the output layer is represented by the vector $c$. The output of the k-th neuron in the hidden layer is denoted by $h_k$, and the output of the j-th neuron in the output layer is denoted by $y_j$.

The equations for computing the output $y$ of an MLP with three layers are

$$h_k = f(a_k) \qquad \text{where} \quad a_k = \sum_{i=1}^{n} W_{ki} x_i + b_k, \quad \text{for } k = 1, 2, \ldots, m \tag{2.8}$$

$$y_j = g(b_j) \qquad \text{where} \quad b_j = \sum_{k=1}^{m} V_{jk} h_k + c_j, \quad \text{for } j = 1, 2, \ldots, p \tag{2.9}$$

In these equations, f() and g() are activation functions that are applied to the weighted sum of the inputs to the hidden and output layers, respectively. Common activation functions include the sigmoid function, rectified linear unit (ReLU) function, or hyperbolic tangent function.

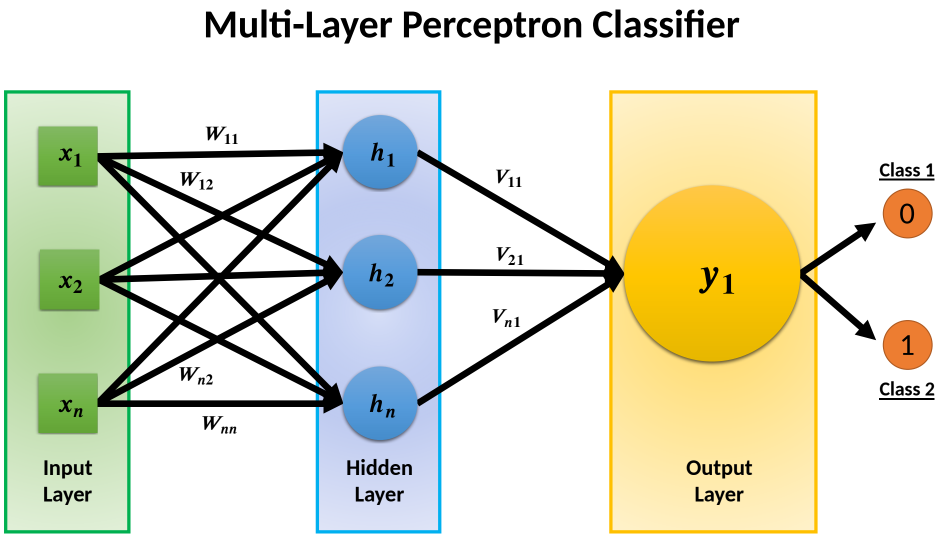


**Figure 2.7:** Representation of a Multi-Layer Perceptron classifier. This model has an input layer, a hidden layer and an output layer. The output layer's output value will determine how close the sample is to each class ( 0 for the first, or  1 for the sceond).

MLPs can be used with various activation functions, such as the sigmoid function, ReLU (Rectified Linear Unit) function, or the hyperbolic tangent function, and can have different numbers of neurons in each layer. The architecture of an MLP, including the number of layers and neurons, can be adjusted based on the complexity of the task and the size of the dataset. As Fig.2.7 represents, in this study, the MLP Classifier architecture is used for binary classification in most cases, using always a **ReLU** function for the hidden layer and having a **logistic function** in the only output neuron in the output layer. This is to predict the outcome of each simulation/data used by using one sample from it.

During training, the weights and biases of the MLP are adjusted to minimize the error between the predicted outputs and the desired outputs. This is typically done by using a **backpropagation** algorithm, which computes the gradient of the error function with respect to the weights and biases, and updates them using an optimization algorithm such as stochastic gradient descent. More specifically, for the optimization of the models used in this study, the **Adam** solver was used until convergence or upon reaching the maximum number of iterations. Convergence is determined by the tolerance and the number of epochs in the training with no change in loss. When having $n_{iter-no-change}$ consecutive epochs with less than $tol$ improvement on the loss, the training stops, and it is considered that the model has reached convergence.

In order to use this model, the input data has to have the shape (features, samples), where the architecture of the MLP has to match the number of features as well. Then the training is done in batches of samples and updated every epoch through backpropagation. This architecture was built using both the SCikit-learn implementation and TensorFlow. The Scikit-learn implementation has early stopping and dropout with a constant learning rate. This was also mimicked in TensorFlow, which allowed faster trainings and a more accurate training on some cases.

## LSTM

An **LSTM** (Long Short-Term Memory) is a type of recurrent neural network (RNN) that is capable of modeling long-term dependencies in sequential data (i.e. time series). Unlike traditional RNNs, which suffer from the vanishing gradient problem and struggle to remember long-term information, LSTMs use a specialized memory cell to selectively remember or forget information over time.

The basic building block of an LSTM is the memory cell, which is connected to the input and output gates and the forget gate. The **input gate** controls the flow of information into the memory cell, the **output gate** controls the flow of information out of the memory cell, and the **forget gate** controls which information is retained in the memory cell. The equations that describe the input gate, forget gate, and output gate are:

*Input gate:*

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \tag{2.10}$$

*Forget gate:*

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \tag{2.11}$$

*Output gate:*

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \tag{2.12}$$

And the equations describing the states of the cells updating during training are:

*New cell state:*

$$\tilde{c}t = tanh(Wxcx_t + W_{hc}h_{t-1} + b_c) \tag{2.13}$$

*Updated cell state:*

$$c_t = f_t c_{t-1} + i_t \tilde{c}_t \tag{2.14}$$

*Hidden state:*

$$h_t = o_t tanh(c_t) \tag{2.15}$$

Here, $x_t$ is the input at time $t$, $h_{t-1}$ is the hidden state at time $t-1$, $c_{t-1}$ is the cell state at time $t-1$, $i_t$, $f_t$, and $o_t$ are the input, forget, and output gates, respectively, and $\tilde{c}_t$ is the new candidate cell state. $W$ and $b$ are weight matrices and bias vectors, respectively, and $\sigma$ is the sigmoid activation function. The tanh activation function is used to compute the new cell state and the hidden state.

During training, the LSTM receives a sequence of inputs and uses the gates and memory cell to selectively store or discard information at each time step. The cell state is updated through the input and forget gates, and the output gate produces the output of the LSTM at each time step.

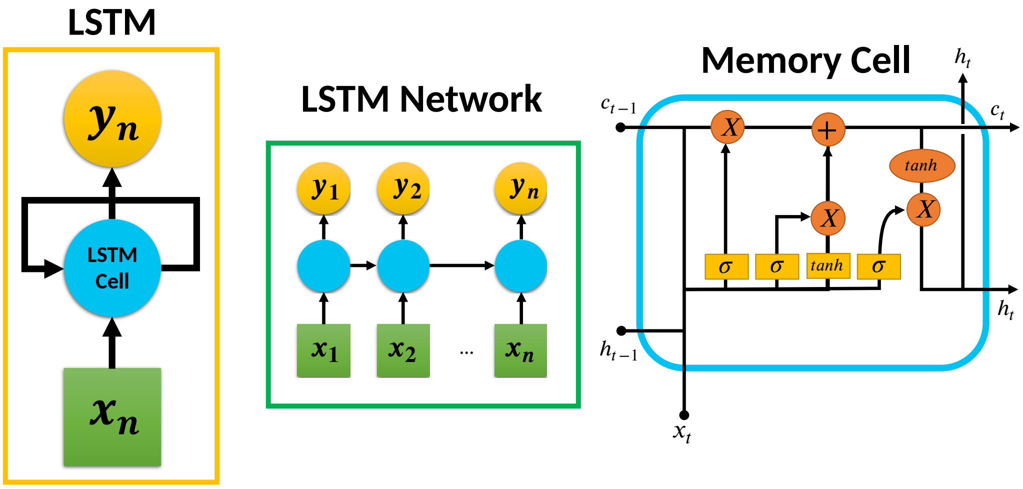


**Figure 2.8:** Diagram of an LSTM block (left), an LSTM network (middle) and the memory cell of an LSTM block (right). Note that for the left and middle diagrams, the output is in yellow, the memory cell in blue and the input in green. For the memory cell diagram, $\sigma$ is a sigmoid layer, *tanh* a tanh layer, $X$ is a scaling of information, $h_t$ is the current output's hidden state and $h_{t-1}$ the previous LSTM's block output, $C_t$ is the cell state and $C_{t-1}$ the previous cell state, and lastly $X_t$ is the current input.

Regarding the specifics for the purpose of this study, the LSTM model was used as a classifier capable of forecasting the outcome state using either a portion of the data or all data. For this, the data had to be in the form of (samples, features, time) or more specifically for simulations (simulations, features, frames), thus meaning that for each independent sample belonging to a class, there is a number of features for each data point in the time series. In contrast to MLP, this model allows the context of the series to be taken into account in the learning. This model was only implemented in TensorFlow in Python, having dropout and early stopping.

## GBDT

**Gradient Boosting Decision Tree** (GBDT) is an ensemble learning method that combines multiple decision trees to create a powerful predictive model. The basic idea behind GBDT is to iteratively add decision trees to the model in a way that corrects the errors of the previous trees. This is done by training each subsequent tree to predict the residual errors of the previous trees. The GBDT algorithm can be summarized in the following steps:
Initialize the model with a constant value:

$$f_0(x) = \frac{1}{N}\sum_{i=1}^{N} y_i \tag{2.16}$$

For each iteration $m = 1, 2, ..., M$:

- Compute the pseudo-residuals:

$$r_{im} = y_i - f_{m-1}(x_i) \tag{2.17}$$

- Fit a decision tree to the pseudo-residuals: $h_m(x)$
- Update the model by adding the decision tree:

$$f_m(x) = f_{m-1}(x) + \gamma h_m(x) \tag{2.18}$$

Here, $y_i$ is the true label for the $i$th example in the training set, $f_m(x)$ is the predicted output of the model at iteration $m$ for input $x$, $h_m(x)$ is the decision tree added at iteration $m$, and $\gamma$ is the learning rate, which controls the contribution of each decision tree.

The GBDT algorithm is essentially a gradient descent method that minimizes the loss function by iteratively adding decision trees that correct the errors of the previous trees. The loss function can be any differentiable function, but the most common choice is the mean squared error (MSE), which is given by:

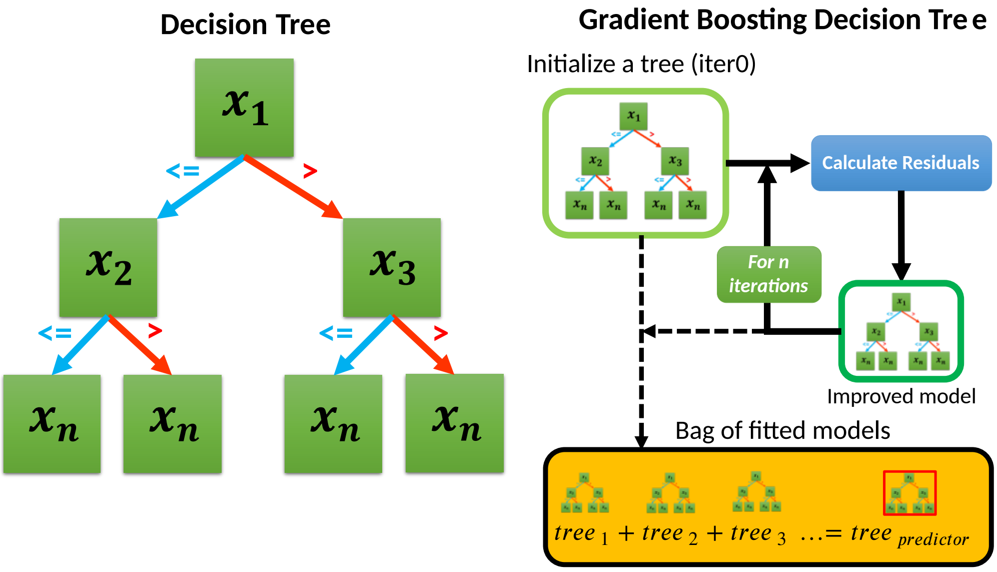


**Figure 2.9:** Diagram of a decision tree and the gradient boosting decision tree (GBDT) strategy. The left diagram shows the decision tree's splitting of features based on a threshold for feature values bigger than (>, in red) or smaller/equal than (<=, in blue). For the GBDT on the right, an initialized tree is subsequently assessed, and the next tree is built improving over the previous for a number of iterations ($n$). Then the bag of fitted models is used for ensemble prediction, resulting in a forest predictor.

$$L(y, f(x)) = \frac{1}{2}(y - f(x))^2 \tag{2.19}$$

The pseudo-residuals are computed as the negative gradient of the loss function with respect to the predicted output of the model:

$$r_{im} = -\frac{\partial L(y_i, f_{m-1}(x_i))}{\partial f_{m-1}(x_i)} \tag{2.20}$$

The decision tree added at each iteration is trained to fit the pseudo-residuals, using a standard tree-building algorithm such as CART. The tree is then added to the model, with its predictions weighted by the learning rate $\gamma$.

The learning rate $\gamma$ is an important hyperparameter in GBDT that controls the contribution of each decision tree to the final model. A smaller value of $\gamma$ results in a more conservative model that takes longer to converge, while a larger value of $\gamma$ results in a more aggressive model that may overfit the training data.

For the use case in this study, GBDT is used as a classifier following the Scikit-learn implementation in Python. In the same fashion as the previous models, the model is trained with a part of the data to predict each of the classes the time series ends in later on. All models minimize with a logistic loss function using the MSE to asses the split of the internal nodes.

### 2.2.4 Feature Analysis

For assessing the relevance of the different features of the datasets used in this work, several techniques are available such as feature permutation [52], Gini feature importance, local interpretable model-agnostic explanations (LIME), and others. Within this work, due to the nature of the models, only the Gini feature importance and our version of feature permutation called *accuracy drop* are used. Additionally in some parts, the layer-wise relevance propagation (LRP) is also used. Thus, this three different methodologies for feature analysis will be explained.

#### Accuracy Drop

**Feature permutation** is a technique used to measure the importance of features in a predictive model. The basic idea is to randomly permute the values of a single feature in the test set, re-evaluate the model on the permuted set, and measure the change in performance. If the performance decreases significantly when a feature is permuted, it suggests that the feature is important for the model's predictions.

The feature permutation importance score for a feature $i$ can be defined as the decrease in performance when the feature is permuted [52]. Let $X$ be the original test set, $y$ be the true labels, and $f(X)$ be the predicted outputs of the model on the test set. The performance measure can be any metric that evaluates the quality of the predictions, such as the mean squared error (MSE) or the mean absolute error (MAE).

The feature permutation importance score for feature $i$ is then given by

$$\text{Importance}(i) = \frac{1}{n}\sum_{j=1}^{n}(f(X_{\pi_{ij}}) - f(X))^2 \tag{2.21}$$

here, $n$ is the number of samples in the test set, $\pi_{ij}$ is a permutation of the $j$th sample in which the value of feature $i$ is randomly shuffled, and $X_{\pi_{ij}}$ is the permuted test set.

To compute the feature permutation importance score, I need to generate multiple permutations of each sample in the test set, and average the decrease in performance over all permutations. This ensures that the score is robust to the particular order in which the permutations are applied.

In a similar way, the **Accuracy Drop** used in this work, is inspired from both feature permutation and the previous efforts on doing feature selection with neural networks [53]. In these previous networks, an MLP (three-layered feed-forward neural network) was trained for a classification task in order to select a subset of features from a high-dimensional feature space that is most informative for a neural network model. They propose a *feature quality index* homologous to the *accuracy drop* used here, where the features are ranked after ”removal” of the feature, i.e. modifying the original feature for each training data point to zero, and subsequently predicting again. This is done on the premise that the output sensitivity to the input is high, thus using a similar value for each feature removed. Depending on the difference between the original prediction score and the modified data point prediction score, the original feature was relevant or not to begin with. That means that a higher drop in accuracy suggests a strong relevance for the model to predict the classes. In the original implementation, a randomly initialized set of networks is then trained and the features are eliminated one by one by setting their value to 0. Then the re-predicted accuracy for each model is calculated, and the feature ranking is obtained.

The methodology for the accuracy drop (AD) used in the present work is highly homologous to the previous, although in this case the difference is mostly in the way of eliminating the features one by one. Instead of setting their value to 0, the total average of each feature across time, also called **global mean** (GM), for the training data is used. This allows the elimination of the features by cancelling out their variance. This GM value is kept constant through all data points, i.e. time, denying the variance of the feature. This results in a decrease in accuracy for the features

**Algorithm 1** Accuracy Drop Calculation in MLTSA

```
$AD_{total} = 0$
for model ($m_i$) in models do
    Split original $Data$ in test ($D_t$) and validation ($D_v$)
    Train predictive $m_i$ with $D_t$
    Asses accuracy of $m_i$ with $D_v$
    $AD_i = 0$
    for feature ($f_j$) in $D_t$ do
        Calculate the global mean $GM_j$ of $f_j$
        Swap $f_j$ for $GM_j$ in $D_t$ to create $D_{swapped}$
        Predict accuracy ($AD_j$) of $m_i$ with $D_{swapped}$
        Append $AD_j$ to $AD_i$
    end for
    Append $AD_j$ to $AD_{total}$
end for
Average $AD_{total}$ over models
return $AD_{total}$
```

whose variance was relevant to predict the outcomes. This provides a more clear and straight-forward relationship for the features found relevant by AD.

The protocol for calculating AD across the desired number of replicas is described in Algorithm 1. Where, over a given number of models, each model $m_i$ is trained, its accuracy assessed, then the training data is iterated to be swapped with its GM each time and re-predicted and recorded. That is then averaged over the number of models to get the final $AD_{total}$. Plotting this value for each of the features allows one to assess the importance.

### Layer-wise Relevance Propagation

**Layer-wise relevance propagation** (LRP) is a method for explaining the predictions of a neural network by attributing a relevance score to each input feature. LRP can be used to understand how the network arrived at its prediction, and to identify the most important features that contributed to the prediction.

The basic idea of LRP is to propagate the output of the network back to the input layer, assigning relevance scores to each neuron along the way. The relevance scores represent the contribution of each neuron to the output of the network, and are computed based on the activations and connections between neurons.

The LRP algorithm can be described in the following steps:

- **Forward pass:** Run the input data through the neural network and compute the activations of all neurons in each layer.
- **Output relevance:** Set the relevance of the output layer to be the target output of the network, i.e., the predicted class probabilities.
- **Relevance propagation:** Starting from the output layer and moving backwards through the layers, propagate the relevance of each neuron to the neurons in the previous layer based on the connections between them. The relevance of a neuron in a given layer is distributed among the neurons in the previous layer based on the strength of their connections, and the relevance is multiplied by the activation of the neuron in the previous layer.
- **Normalization:** After propagating the relevance scores backwards through the network, normalize the scores so that they sum to the original prediction score of the input data.

The LRP algorithm can be written mathematically using the following equation for relevance propagation from layer k+1 to layer k:

$$R_k = \sum_j \frac{a_j w_{jk} + \varepsilon \mathrm{sign}(a_j w_{jk})}{\sum_i a_i w_{ik} + \varepsilon \mathrm{sign}(\sum_i a_i w_{ik})} R_{k+1,j} \tag{2.22}$$

where $R_k$ represents the relevance of layer $k$, $a_j$ represents the activation of neuron $j$ in layer $k+1$, $w_{jk}$ represents the weight between neuron $j$ in layer $k+1$ and neuron $k$ in layer $k$, and $R_{k+1,j}$ represents the relevance of neuron $j$ in layer $k+1$. The parameter $\varepsilon$ is a small positive constant used for numerical stability.

The numerator of the equation represents the relevance score assigned to neuron $j$ in layer $k+1$ based on its activation and connection to neuron $k$ in layer $k$. The denominator represents the normalization factor that scales the relevance score to account for the contributions of all neurons in layer $k+1$. The relevance score is then multiplied by the activation of neuron $k$ in layer $k$ to propagate the relevance backwards to layer $k$.

The relevance scores obtained from LRP can be used to identify the most important input features for a given prediction. Features with high relevance scores are considered more important, while features with low or negative relevance scores are considered less important or even detrimental to the prediction.

The LRP algorithm can be applied to a neural network by modifying the forward and backward passes of the network to compute and propagate the relevance scores. The specific implementation details of LRP depend on the architecture of the network and the type of task it is trained to perform.

Once the LRP algorithm is applied to a network and the relevance scores are obtained for each input feature, further analysis and interpretation may be necessary to draw meaningful conclusions about the importance of the features. Techniques such as visualization, feature selection, or regression analysis can be used to further analyze and interpret the relevance scores obtained from LRP. Thus, for some of the work on this thesis, the LRP has been used to validate the results from the AD. Note that although originally LRP is used in simple architectures such as MLPs, it has been proved useful in LSTM and RNN as well.

### Gini Feature Importance

In a GBDT, the **Gini importance** is a popular method for feature importance analysis. It is a measure of the total reduction in impurity achieved by each feature across all trees in the ensemble.

The impurity reduction of a feature $j$ at a given tree is defined as:

$$\Delta i(j) = i_{\text{parent}} - \frac{n_{\text{left}}}{n} i_{\text{left}} - \frac{n_{\text{right}}}{n} i_{\text{right}} \tag{2.23}$$

where $i_{\text{parent}}$ is the impurity of the parent node, $n_{\text{left}}$ and $n_{\text{right}}$ are the number of samples in the left and right child nodes, respectively, and $i_{\text{left}}$ and $i_{\text{right}}$ are the impurities of the left and right child nodes, respectively.

The Gini importance of a feature $j$ is then defined as the sum of the impurity reductions achieved by $j$ across all trees in the ensemble, normalized by the total number of trees:

$$\text{Gini importance}(j) = \frac{\sum_{t=1}^{T} \Delta i(j)_t}{T} \tag{2.24}$$

where $T$ is the total number of trees in the ensemble.

The Gini importance provides a ranking of the features based on their contribution to the impurity reduction achieved by the GBDT model. Features with higher Gini importance values are considered more important in the prediction task, while features with lower values can be considered less important or redundant. However, it should be noted that the Gini importance is not always a reliable measure of feature importance, especially in the presence of correlated features or interactions with other features. Thus, with the training of the GBDT model, one can obtain the Gini feature importances straight away.

### 2.2.5 Protocol Summary

To sum up the MLTSA's protocol, it can be divided in three parts:

- **Dataset Creation:** Independently from the type of data, any time series data be used. Recommended a system where different features can be extracted from.

- **Model Training:** Models have to be trained at different times throughout the trajectory. Depending on the trend of the test/validation accuracy a specific time is chosen.

- **Feature Analysis:** After successfully training the models, different strategies for pinpointing relevant residues are used depending on the model.

Fig. 2.10 summarizes the standard approach used in this work by a combination of the GBDT and MLP to investigate the different protein datasets as well as validation datasets with the 1D/2D models. Note that the data, model and feature analysis depicted are only examples.

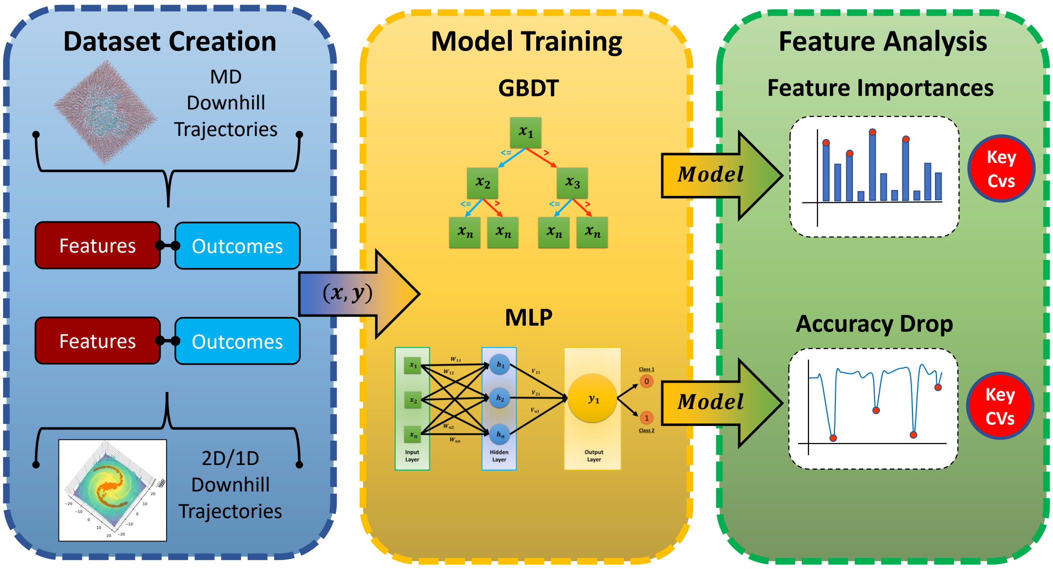


**Figure 2.10:** Diagram of the typical MLTSA protocol from the dataset creation to model training and subsequent feature analysis. The features extracted from the trajectories and their outcomes are used to train ML models to predict the final outcome, then, analysis is performed on the models to obtain the relevant features. The datasets, methods and models are examples used in this work.

## 2.3 ACHREMD

This section is a walk-through of the methodology that leads up to the ACHREMD, passing through HREMD and REMD, the original methodology that these are based off.

### 2.3.1 REMD

**Replica exchange molecular dynamics** (REMD) is a computational technique used to enhance the exploration of conformational space in molecular simulations [54]. The method involves running multiple replicas of a system at different temperatures and periodically exchanging the temperatures of adjacent replicas (see Fig.2.11). This allows replicas to explore regions of conformational space that would be inaccessible at a single temperature.

The exchange of temperatures is usually governed by a Metropolis criterion that ensures detailed balance is maintained between the replicas. Specifically, the probability of exchanging temperatures between two replicas i and j is given by:

$$P_{i\to j} = \min\left(1, e^{-(\beta_i-\beta_j)(E_j-E_i)}\right) \tag{2.25}$$

where $\beta_i = 1/k_B T_i$ is the inverse temperature of replica i, $E_i$ is the energy of replica i, and $k_B$ is the Boltzmann constant. The probability of exchanging temperatures from replica j to replica i is given by:

$$P_{j\to i} = \min\left(1, e^{-(\beta_j-\beta_i)(E_i-E_j)}\right) \tag{2.26}$$

In practice, the exchange attempts are made periodically during the simulation, and the frequency of exchange attempts can be adjusted to optimize the efficiency of the method. Overall, replica exchange molecular dynamics is a powerful technique for enhancing the sampling of conformational space in molecular simulations and has been used to study a wide range of biological and chemical systems [55].

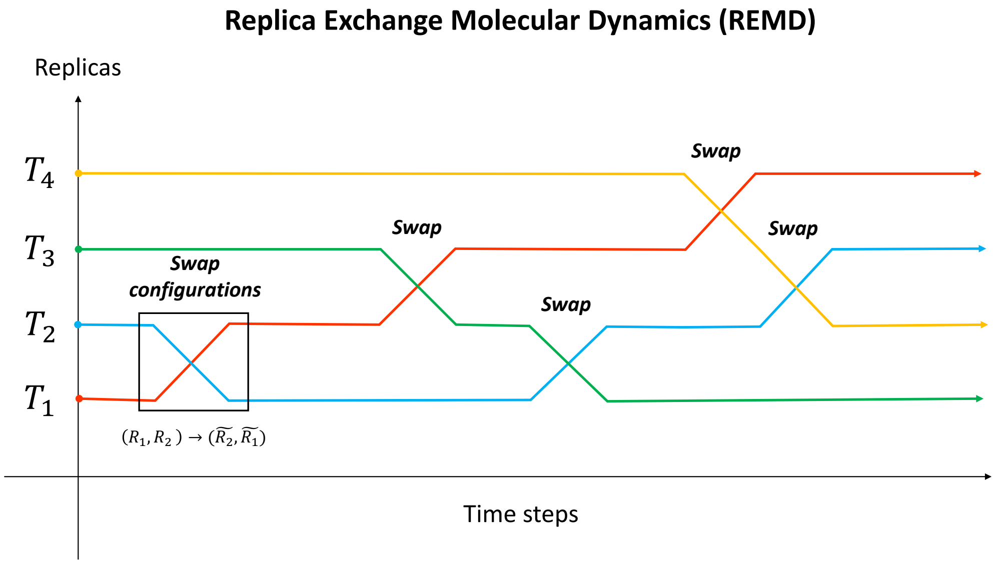

Replica Exchange Molecular Dynamics (REMD)
Replicas
$T_4$
$T_3$
$T_2$
$T_1$
Swap
configurations
$(R_1, R_2) \to (\widetilde{R_2}, \widetilde{R_1})$
Swap
Swap
Swap
Swap
Time steps

**Figure 2.11:** Protocol illustration of the temperature ($T$) replica exchange molecular dynamics (REMD) approach. Replicas run at different $T$ are attempted to exchange neighbouring temperatures at a given frequency. When an exchange is accepted, configurations are swapped and the replica continues to run at the same $T$ with the new configuration.

### 2.3.2 HREMD

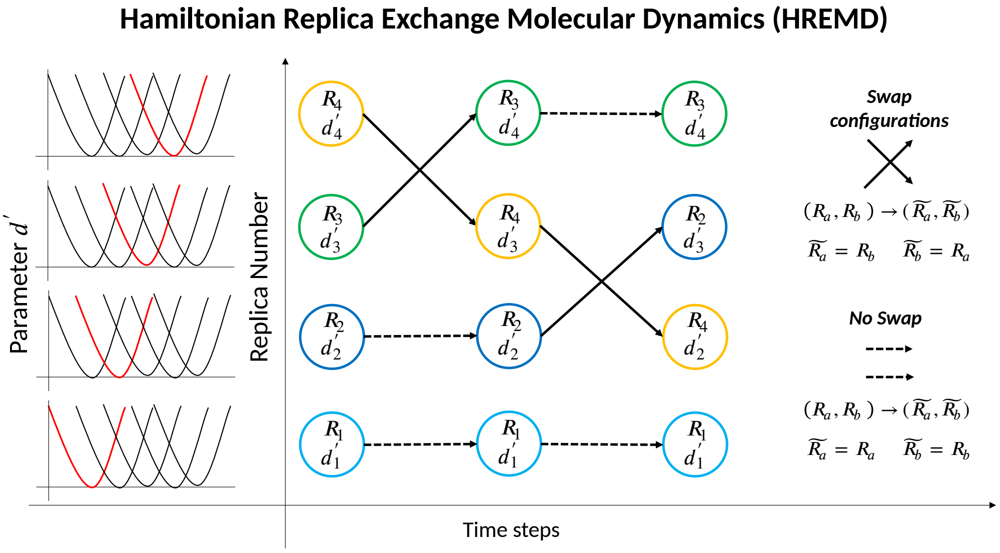

Hamiltonian Replica Exchange Molecular Dynamics (HREMD)
Parameter $d'$
Replica Number
$R_4$ $d'_4$
$R_3$ $d'_3$
$R_2$ $d'_2$
$R_1$ $d'_1$
$R_3$ $d'_4$
$R_4$ $d'_3$
$R_2$ $d'_2$
$R_1$ $d'_1$
$R_3$ $d'_4$
$R_2$ $d'_3$
$R_4$ $d'_2$
$R_1$ $d'_1$
Swap configurations
$(R_a, R_b) \to (\widetilde{R}_a, \widetilde{R}_b)$
$\widetilde{R}_a = R_b$ $\widetilde{R}_b = R_a$
No Swap
$(R_a, R_b) \to (\widetilde{R}_a, \widetilde{R}_b)$
$\widetilde{R}_a = R_a$ $\widetilde{R}_b = R_b$
Time steps

**Figure 2.12:** Protocol illustration for the Hamiltonian replica exchange molecular dynamics (HREMD) approach. A given parameter $d'$ governing different Hamiltonians is used in each replica. In a similar way to REMD, trajectories are run for a given number of steps and then attempted to exchange configuration. A successful swap will exchange configuration of neighbouring replicas, while a failed swap will have them remain unchanged.

**Hamiltonian Replica Exchange Molecular Dynamics** (HREMD) is a variant of REMD in which replicas are simulated at different Hamiltonians ($H$) or potential energy functions containing a given parameter to tweak with different values in each replica [56]. In Fig.2.12 an illustration of a typical HREMD protocol can be found. The probability of exchanging configurations between two replicas i and j is given by:

$$P_{i\to j} = \min\left(1, e^{-(\beta_i-\beta_j)(H_i-H_j)}\right) \tag{2.27}$$

where $\beta_i = 1/k_B T_i$ is the inverse temperature of replica i, $H_i$ is the Hamiltonian energy of replica i and $k_B$ is the Boltzmann constant. The probability of exchanging configurations from replica j to replica i is given by:

$$P_{j\to i} = \min\left(1, e^{-(\beta_j-\beta_i)(H_j-H_i)}\right) \tag{2.28}$$

In these equations, the exchange of configurations is governed by a Metropolis criterion, and the probability of exchange is influenced by both the energy difference between the replicas and the difference in their respective Hamiltonians. HREMD is a powerful technique for exploring the conformational space of complex biomolecules, including protein-ligand interactions and protein folding [56].

### 2.3.3 ACHREMD

The **Atomic Charge Hamiltonian Replica Exchange Molecular Dynamics** (ACHREMD) is another flavour of REMD derived directly from the HREMD. In this case, the parameter to modify specifically is the atomic charge of an atom ($q_{atom}$). To ensure detailed balance is kept, the original system and the replicas at different $q$ maintain the same charge overall. For example, a $Mg^{+2}$ ion may have been reduced from a +2 Charge to a +1 charge. To balance out the loss of charge and have the system with the same charge, two relevant atoms such as $Cl_{-1}$ will have their charge changed to -0.5. Thus, $Mg^{+2}$ + $2Cl^{-1}$ = $Mg^{+1}$ + $2Cl^{-0.5}$.

In this work, this idea has been explored with the change of atomic charges, either by modifying relevant interacting atoms or adding dummy atoms that have their charge acting as a buffer to balance out the charge. An illustration of one of the examples reported in this work can be found at Fig.2.13 for 4 replicas from +2.0 to +0.4 on an active site's $Mg$.

This methodology allows for the flattening of the free energy landscape, focusing specially on problems where the electrostatic interactions of a complexed system do not allow one to observe the dissociation in a reasonable timescale. It becomes very useful for problems like ligand-unbinding, molecule complexation, and complex active site rearrangements.

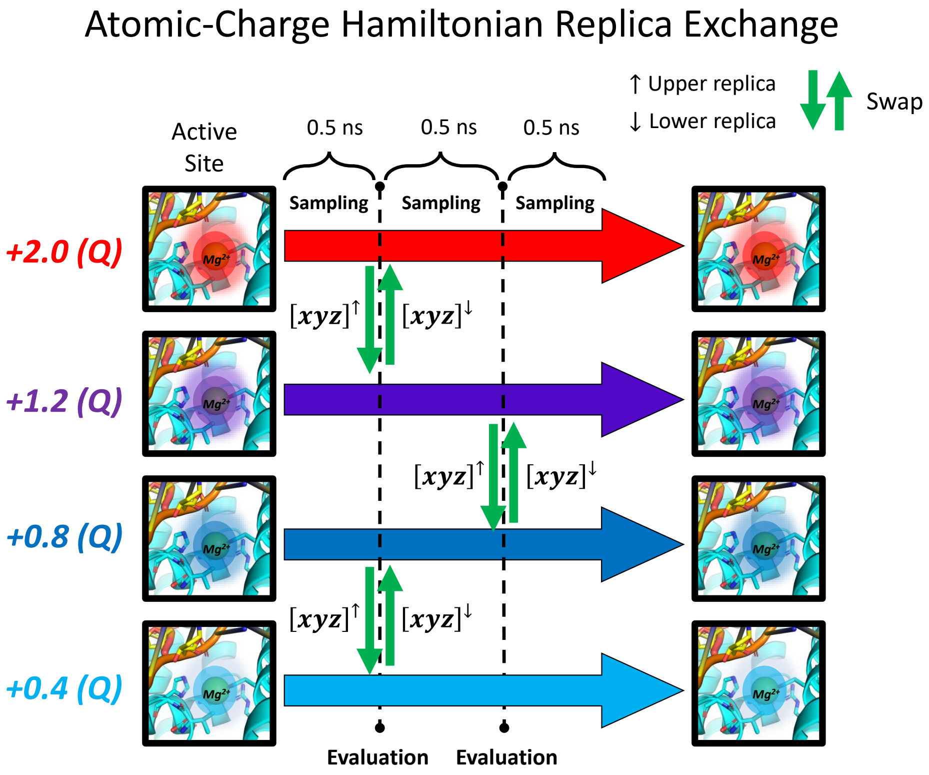


**Figure 2.13:** Illustration of the atomic charge hamiltonian replica exchange molecular dynamics (ACHREMD) protocol. In this system, a $Mg^{+2}$ ion has its charge modified across 4 replicas down to +0.4. Every number of given simulation steps for sampling, an evaluation decides whether exchanges between neighbouring windows happen or not. If an exchange is accepted, the configurations between upper and lower charge replicas are swapped and another round of sampling begins.

Chapter 3

# Machine Learning Transition State Analysis (MLTSA)

This chapter is partially extracted from our previous published applied method [1]. Here I collected the analytical model validation from the first paper and continued the investigation towards exploring MLTSA's capabilities and its validation on other dynamical models, such as the 2D models we created.

## 3.1 Introduction

**Machine learning for molecular dynamics analysis.** MD data is intrinsically high-dimensional, which makes it difficult for researchers to understand complex biological processes. Not only that, but also smaller details can be brushed off while looking at other major rearrangements during trajectory analysis. In addition, many long-timescale events such as ligand unbinding an be a combination of multiple small steps with many intermediates requiring smaller rearrangements, similar to a checkpoint. These smaller steps are also difficult to notice, fast-changing interactions can also be overlooked or they are simply too noisy to correlate with the main event taking place.

**Current challenges.** ML models are well-suited for learning to predict or classify atomic structures [24], moreover protein conformations and their different states [57], however, the non-linearity of some of these methods can make them hard to interpret. Although ML methods have proven useful in the interpretation of

MD data [33], there is still a need for further development, enabling strategies and validation as well as improving interpretability of the models. Recent efforts have made use of autoencoders and variational autoencoders for representation learning of molecules, however, their representation is not intrinsically related to the internal coordinates of the system and simple relations such as distances, angles or interactions cannot be derived easily [34, 58].

In order to contribute towards the ML-aided interpretability of MD, I have developed the **machine learning transition state analysis** (MLTSA) which has originally been used to predict the outcome of downhill simulations started at the TS to pinpoint relevant interactions near the TS [1, 2]. Although it has proven useful for ligand-unbinding, it is not limited to MD data only. To validate the methodology and explore its usage, I developed a **one-dimensional-based analytical model** to assess MLTSA's ability to pinpoint correlated features from time series. I tested two ML models, two MLP and GBDT classifiers tasked to predict the outcome.

Although our approach has shown to work for this one-dimensional analytical model [1], its feature relevance remains the same during the duration of the trajectories, i.e. time series. Thus, a **more complex potential surface validation model** was designed. I implemented a framework to both simulate two-dimensional trajectories in langevin dynamics and generate one-dimensional projected features from a given 2D potential surface.

**Spiral datasets** are a type of synthetic dataset often used in ML to illustrate and test the capabilities of the models. These datasets are often used to test model performance, especially for classification and pattern recognition tasks. They consist of points that are arranged in a spiral pattern in a two-dimensional space.

They are particularly interesting because they challenge algorithms to learn complex and nonlinear decision boundaries. These datasets can't be effectively separated using simple linear classifiers, making them a good benchmark for testing the capabilities of more advanced algorithms.They highlight the strengths and weaknesses of different algorithms and help researchers and practitioners make informed choices about which methods to use for different types of tasks.

There are generally two main types of spiral datasets, **single-armed** and **dual-armed**. In the first one the points form a single spiral arm that starts from the center and spirals outwards, which would end following only one path to one end point. In the latter one, the points form two spiral arms, usually winding around each other in opposite directions, this one is often considered more challenging because the two arms may intertwine at some points making the decision boundary more complex. Thiss allows us to get a validation framework representation closer to that of a real-world problem such as a ligand- unbinding event.

Thus, taking advantage of the higher complexity of the data, a **spiral potential surface** was used to generate a spiral dataset. Despite this being a more complex task, it was still reasonably easy for a time-series to predict the corresponding class/state by looking at which arm the class went through. Thus, with a bit of feature engineering, one-dimensional projections of the original 2D data were created as features for training, distributing the information among multiple features. All of this was done for a more detailed benchmark on the MLTSA analysis. This dataset allowed for assessing both the capability of the ML models and the MLTSA protocol to pinpoint relevant features and how their importance changes at different times throughout the trajectory.

A summary of the data, approaches, feature generation and evaluation in this work can be found in Table 3.1.

## 3.2 Applied Methodology

### 3.2.1 Analytical Datasets

In order to validate MLTSA's ability to pinpoint relevant CVs, two different analytical toy models were developed. The first one and simplest is based off a set of one-dimensional potentials (*1D Analytical Model*) that are used to simulate particle motion on, thus generating different 1D coordinates. The trajectories are started from the same point each time, resembling a TS, and its coordinates are stored as if they were coming from the same system and as a unique trajectory, emulating the different coordinates one would obtain in a complex system. Coordinates are then mixed with each other within a trajectory to distort them and make them correlated with each other. Then only one of the original potentials is used for labelling its outcome, using its original coordinate values. This is done to resemble a real-life situation where the deciding potential for an event might not be able to be directly sampled, but different variables that are highly correlated to it can be sampled. Another model based on two-dimensional potentials of different shapes was also created. This time only one *2D potential* is simulated starting at the same point for each trajectory, emulating a TS or saddle point. In this case the final coordinates used for prediction are lower-dimensional projections of the original 2D coordinates of the trajectory on different axes, i.e. one-dimensional coordinates projected from the original. This emulates a situation where only lower-dimensional projections (distances/angles/dihedrals/etc. for MD) are available from the experiments. Additionally, the same information is degenerated on multiple coordinates, thus making its importance through time change as the trajectory advances.

## 1D Analytical Model

In Fig.3.1 a visual summary of the whole methodology for generating curated the 1D analytical datasets can be found. From a series of potentials, features are generated in random linear combination of the different potential's coordinates, resulting in correlated and uncorrelated features to the labelled outcomes (IN/OUT).

**Defining the potentials.** The data in this model is first generated in one-dimensional potentials of two different shapes depending on the number of minima they have: Single-Well potentials with one minima at 0 (SW, $y_{SW}$); and Double-Well potentials (DW, $y_{DW}$) with two symmetrical minima and a saddle point placed at 0.5. The equation describing the shape of the potentials is

$$y = k_1 \left( x - \frac{1}{2} \right)^4 + k_2 \frac{1}{\sigma\sqrt{2\pi}} e^{\left( -\frac{1}{2} \left( \frac{x-\mu}{\sigma} \right)^2 \right)} \tag{3.1}$$

Where $k_1 = 100$, $\mu = \frac{1}{2}$ and $\omega = 0.01$. To generate a SW $k_2 = 0$, and for a DW $k_2 = 1$.

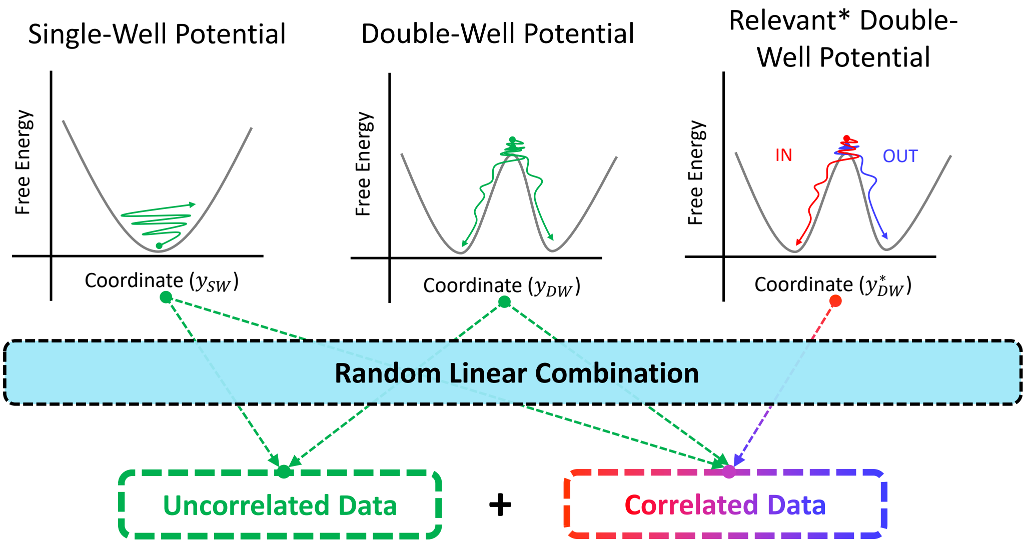


**Figure 3.1:** Illustration of the 1D analytical model dataset creation starting from the defined potential, trajectory generation and linear combination to generate a dataset with both correlated and uncorrelated data. Note that data in this context refers to the final usable features created coming from combinations of the coordinates $y$ sampled. The relevant potential ($y^*_{DW}$)is the one used to compute the labels for training.

**Trajectory generation.** With the shape of the potentials, the $y_{SW}$ and $y_{DW}$ coordinates are calculated along the potentials using the overdamped Langevin equation,

$$\frac{d\mathbf{x}(t)}{dt} = \frac{\mathbf{F}(\mathbf{x}(t))}{\gamma} + \sqrt{\frac{2k_BT}{\gamma}}\xi(t) \tag{3.2}$$

Where $\mathbf{x}(t)$ is the position of the particle at time $t$, $\mathbf{F}(\mathbf{x}(t))$ is the force on the particle at position $\mathbf{x}(t)$, $\gamma$ is the friction coefficient, $k_B$ is the Boltzmann constant, $T$ is the temperature, and $\xi(t)$ is a random noise term with zero mean and unit variance that models the thermal fluctuations in the system.

This equation describes the motion of a particle in a medium with friction, where the particle experiences both a deterministic force, given by $\mathbf{F}(\mathbf{x}(t))$, and a random thermal force, modeled by the noise term $\xi(t)$. The term $\sqrt{2k_BT/\gamma}$ represents the strength of the thermal force, and the friction coefficient $\gamma$ determines the rate at which the particle loses energy to the medium.

For the simulations in this work, I used a simplified 1D version where $2k_BT =$

1, the coefficient $\gamma = 0.01$ is constant and the force $\mathbf{F} = \mathbf{y}(\mathbf{x}(t))$, $\xi(t)$ is a number randomly sampled from a normal (Gaussian) distribution centered at 0 and the spread is 1.0.

Using this I can generate downhill simulations and calculate the coordinates started from $y = 0.5$ for both SW and DW trajectories for the desired number of steps, which depending on the friction and the timestep will be noisy for SW and end up in one of the two minima for DW.

**Dataset creation.** To create the datasets, I simulate downhill trajectories in parallel for each potential starting at $y = 0.5$ for a given number of SW and DW potentials. Taking a look at the final coordinate value of each of the DW potentials, after a reasonable number of simulation steps, one can classify them in two different states (see left panel Fig.3.2), left potential at $< 0.3$ and right potential at $> 0.8$. I call the lower value IN and the bigger value OUT to resemble that of the two states representing the unbinding event. The starting point (0.5) would be then representing the TS of the unbinding. In this manner, one can obtain labels for each simulated DW potential for each trajectory. However, I will only use one of the DW potentials, called **relevant DW** ($y^*_{SW}$), to label the system's outcome, while using all of the potentials to generate the features for the datasets. This way the data becomes complex and it is more challenging for the ML models to predict the outcome from early times as seen in Fig.3.2.

Using the trajectories generated I defined input features ($y_{feature}$) by combining the coordinates from two different potentials ($y_{pot1}$ and $y_{pot2}$) as

$$y_{feature} = \alpha y_{pot1} + (1 - \alpha) y_{pot2} \tag{3.3}$$

Where $\alpha$ is **the mixing coefficient** which decides the contribution of each of the $y_{pot}$ to the resulting feature $y_{feature}$. This linear combination allows one to have the knowledge of the correlation for the new feature to each of the original potentials, thus tracing back the correlation to the relevant DW ($y^*_{DW}$) potential. This becomes useful for validation purposes later on in the feature analysis when looking at the ML identified relevant input features. In this way, many input features can be generated from multiple potentials by randomly picking two of them and assigning a random mixing coefficient. Note the resulting data will have both correlated and uncorrelated data, resulting of a mixture of both SW and DW potentials, adding complexity to this model (see Fig.3.2). Additionally, this mixing can be done as well for the wanted degree of mixing $n$, using $n$ potentials to create a feature.

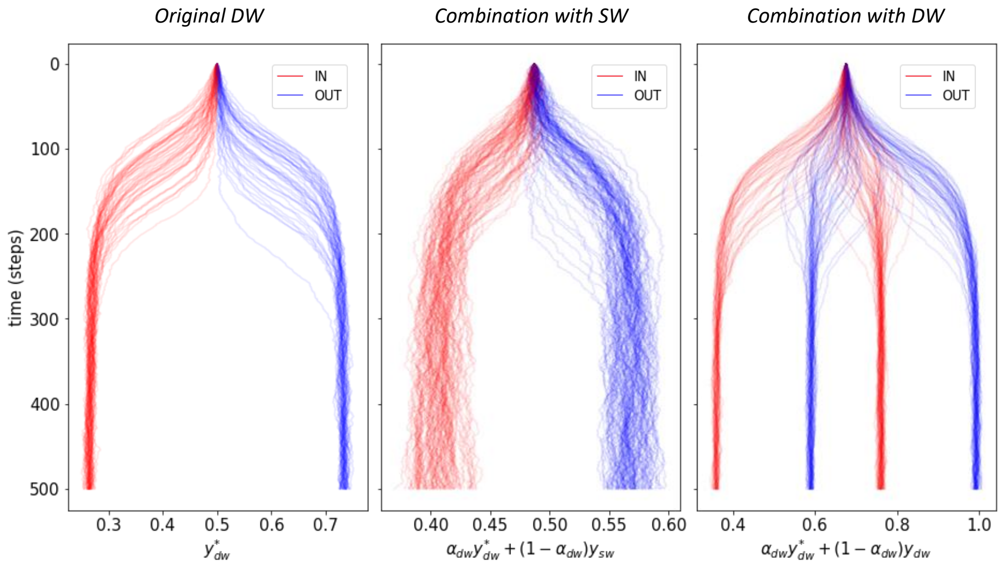


**Figure 3.2:** Illustration of an example of combinations with a relevant DW ($y^*_{dw}$). Left is the original $y^*_{dw}$, middle is the resulting combination with a SW ($y_{sw}$) and right is another combination with a non-relevant DW ($y_{dW}$).

In summary, one trajectory run with $n_{steps}$ in each potential is considered to be from the same sample/simulation and when mixed $n_{features}$ times, it forms one independent trajectory with $n_{features}$. Following this methodology, the dataset is created, having the the shape of $(n_{simulations}, n_{features}, n_{steps})$.

## Computational Details for the 1D Datasets

**Trajectory generation.** Trajectories were generated following the methodology in section 3.2.1, using the Overdamped Langevin Dynamics Equation (Eq. 3.2). 200 independent trajectories were generated for 500 steps on different potentials (24 SW and 1 DW). This set of data will be called *1-DW*. An additional 2nd dataset including 20 SW and 5 DW potentials (with only 1 decisive DW as mentioned in section 3.2.1) was also created and will be referred to as *multi-DW*.

**Dataset creation** Both *1-DW* and *multi-DW* sets were used to create two experimental datasets of the same name. These consist of 180 features newly generated from the obtained trajectories. Before that, a DW potential is chosen to be the relevant potential and its value at the end of the trajectory will determine the label or class it will have (similar to an IN/OUT). The features are generated by randomly selecting two potentials and combining them to generate a new coordinate/feature as a linear combination with a random mixing coefficient ( $0 < \alpha < 1$ ) per feature as described in section 3.2.1. For the features that are mixed the chosen DW potential coordinates, their correlation will directly be affected by the value of their $\alpha$, having no correlation for a value of 0 and a total correlation for 1. The mixing coefficients for both datasets can be found in the SI section. Thus, the two datasets with the features and labels are created for training.

## 2D-based Analytical model

The need for a more challenging example, which includes time-dependency on the feature importance and allows for a better assessment of the model performance, prompted the development of a 2D analytical model potential. Although the pipeline for this is highly customizable, the shape chosen for the experiments was that of the spiral, where particles have the highest energy at the chosen TS point and advance downhill quickly through the different arms (see Fig.3.3). This is used to resemble the downhill simulations of an actual TS structure from an unbinding profile, which can be noisy around the TS and it quickly goes through one direction or the other despite having very close values in one dimension (the X axis in this case), and moving only on the other dimension (large values on the Y).

**Defining the potential surface functions.** The potential surface is shaped with valleys and barriers so the downhill simulations run through the correct paths for each. Additionally, trajectories will start from the origin (0,0) and will be simulated to go down two paths for the Spiral Dataset as shown in Fig. 3.3, where the TS is the origin.

**Langevin Dynamics.** In two dimensions, the position of the particle can be represented as a vector $\mathbf{r} = (x, y)$, and the potential surface can be represented as a

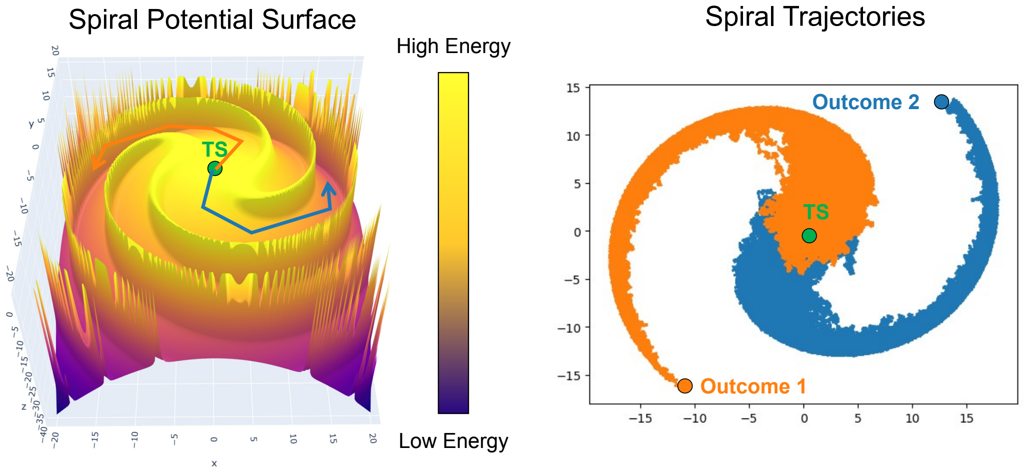


**Figure 3.3:** Illustration of the spiral potential surface and the generated trajectories from the started transition state point.

function $V(x,y)$. The Langevin equation for the particle's position is

$$m\frac{d^2\mathbf{r}}{dt^2} = -\nabla V(\mathbf{r}) - \gamma\frac{d\mathbf{r}}{dt} + \sqrt{2mk_BT\gamma}, \eta(t) \tag{3.4}$$

where $m$ is the mass of the particle, $\gamma$ is the friction coefficient, $k_B$ is the Boltzmann constant, $T$ is the temperature, and $\eta(t)$ is a Gaussian random force with zero mean and autocorrelation $\langle\eta(t)\eta(t')\rangle = \delta(t-t')$.

To simulate Langevin dynamics in a 2D potential surface, one can discretize the equation using the Euler-Maruyama method. One first defines a time step $\Delta t$, and then updates the position and velocity of the particle at each time step.

Then, at each time step, one would calculate the gradient of the potential using finite differences, update the velocity and position using the equations, and record the new position of the particle. One would repeat this process for a specified number of time steps to simulate the dynamics of the particle in the potential surface.

**Dataset Creation.** After labelling the outcomes of the Spiral trajectories, their $X,Y$ coordinates are transformed and rotated following the equation:

$$(X_{rotated} = cos(\theta) * X_{priginal}), (Y_{rotated} = sin(\theta) * Y_{original}) \tag{3.5}$$

Where $\theta$ is a given angle to be rotated to. After the rotation, as in the procedure in this work, only the coordinate values on the new $X$ axis is kept, thus making $X_{proj} = X_{rotated}$ being the input feature to use in the dataset. This new coordinate is the projected coordinate onto one dimension.

In the work portrayed here, this is done for 72 different $\theta$, thus enabling to capture the one-dimensional projection in all 360 degrees with a spacing of 5 degrees between them. Note this can be done differently for other approaches.

### Computational Details for 2D Dataset

**Trajectory Generation.** Trajectories are generated using langevin dynamics, starting at (0,0) as the highest energy point. Then in a ”downhill” manner, trajectories started there will end up in two of the available arms from the spiral.

**Labelling.** A clustering algorithm (DBSCAN) is used to classify in an automated fashion the final datapoints in order to label the data with two classes (Left and Right).

**Creating Features.** To generate features, a rotation of the data to a given angle is performed to project it in the $X$ axis. Then the transformed data’s $X$ axis is kept as an $X_{proj}$. This new projected coordinate is then used as an input feature. This is performed from 0 to 360 degrees, every given amount, depending on the wanted number of features. In this work, I used 72 features, which meant projecting every 5 degrees.

### 3.2.2 Trainings and models

The dataset is split into a **training set** (70% of all trajectories) and a **test set** (30% of all trajectories) for the ML training. In addition, I run additional independent simulations for a **validation set** to prevent any overfitting. To ensure significance, the trainings are replicated 100 times with a random seed for the initialization each time and a random splitting between the training/test/validation sets.

The models used in this work are the MLP (`MLPClassifier`) and GBDT (`GradientBoostingClassifier`) implementation by Scikit-learn [59]. The details on the models are:

- The **MLP** model was setup using 3 layers (input, hidden, output) with as many input nodes as features used in the first layer, 100 hidden nodes in the second layer using the ReLU [60] activation function and 1 output layer made of 1 node with a logistic activation function (0-IN/label1, 1-OUT/label2). The model was optimized using the Adam solver [61], with a learning rate of 0.001, iterating over data until convergence or upon reaching the maximum number of iterations (500 epochs). Convergence is determined by the tolerance and the number of epochs with no change in loss. When having 10 consecutive epochs with less than 0.0001 improvement on the loss, the training stops, and it is considered that the model has reached convergence. All datasets used the same parameters for training.

- The **GBDT** model was trained with 500 decision stumps as weak learners minimizing a logistic loss function, with a learning rate of 0.1. The Friedman Mean Squared Error (MSE) was used for to assess the split of the internal nodes, using a minimum of 2 samples to split, and 1 sample required at the leaf nodes. The maximum depth of the individual estimators was 3, without a limit on the maximum number of features to consider for the best split or the leaf nodes. Training was done using a validation fraction of 0.1 internally.

### 3.2.3 Summary

As shown in Table 3.1, both 1D and 2D analytical models have analogous processes. The 1D is based on SW and DW potentials, whereas the 2D has a Spiral shaped potential. They are both simulated starting at the highest energy state (TS) and go downhill towards one of two outcomes (IN/OUT or Left/Right). Features are generated afterwards, since using the plain coordinates would be too trivial. In the 1D analytical model, there can be correlated or uncorrelated data, whereas all data in the 2D is correlated to some degree. However, some correlated data can be more descriptive of the direction of the trajectory, thus being more ”important”.

**Table 3.1:** Summary of the analytical models explored using MLTSA for validation.

| Process | 1D Analytical Model | 2D Analytical Model |
|---|---|---|
| Potential | SW/DW | Spiral |
| Trajectory Generation | Langevin Dynamics | Langevin Dynamics |
| Feature Generation | Linear Mixing | Coordinate Projection |
| Analysis | AD/FI | AD/FI |
| Evaluation | Correlation Score ($\alpha$) | Jaccard Index |

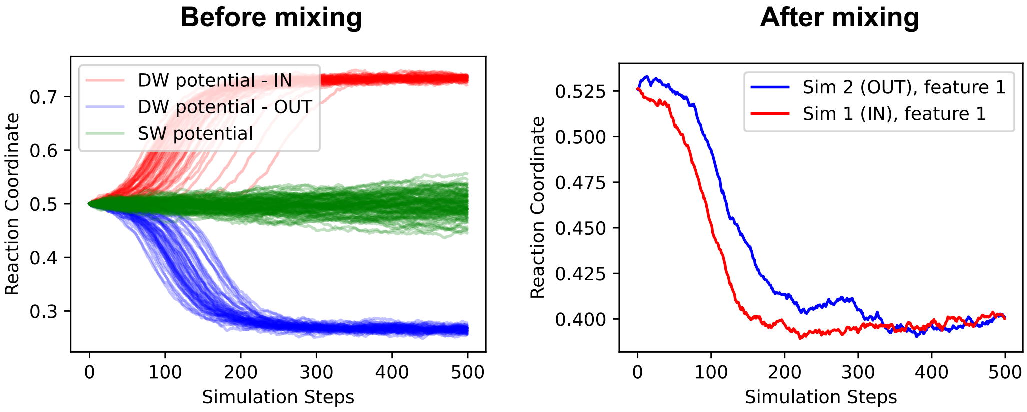


**Figure 3.4:** Left: simple coordinates generated on the SW and DW potentials.The left plot represents data generated on a SW and a DW, the trajectories labelled as IN are in red, whereas the ones labelled as OUT are portrayed in blue. Right: An example of a resulting feature from a linear combination between SW and DW for both IN and OUT trajectories.

# 3.3 1D Analytical Model Results and Discussion

## 3.3.1 Data generated

**Producing features.** As mentioned in section 3.2.1, the raw output (1D coordinates) from the one-dimensional potentials is too trivial for the ML models to predict their trajectory outcome, even at early/short times. Thus, an additional post-processing step has to be done to generate multiple features, with and without correlation to our decisive potential to be able to detect with the MLTSA if and how correlated they are. As seen in Fig. 3.4 (left plot), the decisive DW potential has two classes, one that tends to a lower threshold at the end of the trajectory, which will be labelled as ”IN” (red), and a higher threshold which will be labelled as ”OUT” (blue). However, SW potentials (one example in green) are very noisy towards the end, only having values oscillating around 0.5 as in $0.4 < x < 0.6$. All of these coordinates represent ”internal coordinates” one can obtain from a real MD simulation. These coordinates are transformed during the post-processing into a new input features through the linear combination mentioned in section 3.2.1. The **linear combination** done in this step of a SW trajectory coordinate with the decisive DW* trajectory coordinate, can generate, for example, features similar to feature 1 in Fig. 3.4 (right plot). Note that even in different individual trajectories (Sim.1/IN and Sim.2/OUT) the correlated feature produces the same trend for two different outcomes. This step was done for 200 trajectories on two sets of different combinations of SW and DW potentials (1-DW/multi-DW) and new input features (180) were generated randomly for the analysis, containing both correlated and uncorrelated data. However, when features are highly correlated, one can still visually discern which is the outcome easily. Thus, a training at different trajectory times has to be done to be able to asses the accuracy of the models and a visual inspection are necessary to determine the right time for the given problem.

## 3.3.2 Training through time

**Searching for the right time-frame.** ML training on the model potential-derived trajectories was performed with both MLP and GBDT ML methods. The MLP training was performed at different time frames and trajectory lengths, from the 0th time step to the 500th step in intervals ranging from 10 to 150 frames at a time to assess the accuracy through time (Fig. 3.5).

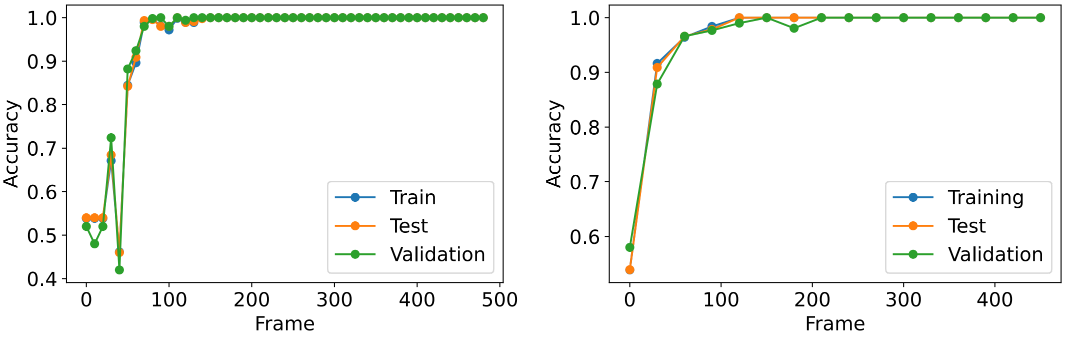


**Figure 3.5:** Accuracy of the Multi-Layer Perceptron (MLP) prediction for the dataset with 1 DW and 24 SW potentials at different starting times using 10 frame intervals (left) and 30 frame intervals (right).

As shown in Fig. 3.5, at times later than 100 steps/frames, the model quickly achieves 100% accuracy on both train and test sets, even in the validation in some cases. This happens when using any amount of values from 10 to 150.

**Relevance of the right time-frame.** It is important to note that during the exploration of these models for feature analysis purposes, a model (for both MLP and GBDT as well) that performs over >98% accuracy (on validation) is often rendered useless for identifying relevant features. This phenomenon involves the model having 0 accuracy drop (AD), or having its feature importances (FI) too small to find any relevant one, thus implying *everything* is relevant. We have observed this in biologically-relevant data as well [1]. I think this may happen for two main reasons. The first one is related to the data, the data itself may be too biased already towards the final outcome, i.e. having big values at a late time which means being OUT, and may already be giving away too much information. This, however, would not explain this happening in the analytical model, in which totally uncorrelated features should remain non-relevant, while the relevant ones should be picked up as relevant still. Something I have observed is that greatly increasing the number of features (up to more than a thousand) has a similar effect where the accuracy is high (>98%) but no feature analysis can be done and no single features become important for predicting. I believe this might be due to a similar problem where too much correlated data is present, but in this case the learning is split between all features. Note that when the number of generated features is increased, the number of features mixed with the relevant trajectory stay the same, while more ”distracting” uncorrelated features are generated. Thus, the model relies in all features by a small margin and altering only one at a time has no effect in the accuracy for the MLP, while the feature importance in the GBDT becomes so divided that no features are highlighted. We noticed this happening for biologically-relevant data as well [1].

On the other hand, when using models that are underfitted ($< 70\%$ accuracy), the results are not reliable and both relevant and non-relevant features get picked up by both MLP and GBDT, probably due to stochasticity and wrong learning. On the same note, an inaccurate model is not reliable for feature analysis, since the problem has not been learned.

Therefore, using a suitable time range is of the essence. In this work, I found a suitable time consisting of the 30th-60th simulation steps for each trajectory. This is a time frame where it is hard to guess the outcome even for humans (See Fig. 3.2) using the raw data. In this range, the trained ML methods found the classification problem accurately solvable, but not too trivial. As shown in Fig. 3.5 this time range stays below the 98% accuracy for validation.

**Table 3.2:** Table containing the average accuracies (for training, test and validation) and number of epochs used for training of GBDT and MLP methods over the 100 independent replicates of our procedure, for both types of datasets (1 DW and the 5 DW potentials) tested.

| | **Training Acc.** $< \% >$ | **Test Acc.** $< \% >$ | **Validation Acc.** $< \% >$ | $< Epochs >$ |
|---|---|---|---|---|
| *GBDT (1-DW)* | 100.00 | 99.72 | 91.45 | 500 |
| *MLP (1-DW)* | 94.83 | 94.73 | 93.04 | 204 |
| *GBDT (multi-DW)* | 100.00 | 99.80 | 91.64 | 500 |
| *MLP (multi-DW)* | 91.85 | 91.83 | 89.32 | 311 |

**Accuracy at the sweet spot.** I replicated the complete process 100 times by generating 200 new independent simulations for each replica and performing the training afterwards. Training accuracies for both ML models at 1DW and 5DW potentials can be found in Table 3.2. The MLP achieved an average test accuracy of over 94% and an average validation accuracy of over 93%, whereas the GBDT achieved over 99% on the test set and 91% on the validation set, suggesting a better generalization for the *1-DW set*. However, there is a slight drop in validation accuracy for the MLP when dealing with the *multi-DW set* compared to the *1-DW set*, whereas the GBDT had a similar accuracy in both cases. This could suggest that it is more suitable for complex problems, however, no hyperparameter optimization was done for the models, thus no assumptions can be drawn from this alone. One also has to consider that the GBDT model takes, on average, 2x the training time of that of the MLP model. In addition, increasing complexity on the analytical model shows an overfitting and bad generalization on the GBDT (See A.1).

### 3.3.3 Feature Analysis

**Validating the feature selection.** To identify the selected DW potential and its highest correlated features from the dataset, I calculated the AD (MLTSA as in Fig. 2.10 and described in section 2.2.4) using the trained MLP and compared this approach to the FI using GBDT. Results of both feature analysis methods are found in Fig. 3 for the 1DW dataset and in Fig. S6 for the 5 DW potentials dataset.

For the *1-DW set*, the highest correlated features (colored depending on the correlation level, color bar in Fig. 3.6.a) were correctly identified by both MLP and GBDT models. For GBDT, only the top three features show a high FI value (labels added to datapoints in Fig. 3.6), whereas the rest of the correlated features ranging from $\alpha$ 34% up to  60% do not show a significant FI value. In addition, despite three features (# 48, #89 and #136) having 40.34%, 34.80% and 35.48% mixing coefficients, respectively, GBDT did not capture their correlation, showing values very close to 0. For the MLP, the top three distances are similarly captured as in the FI with the highest ADs. Importantly, all correlated features have a non-zero AD, showing that they are correctly identified.

Using the dataset with increased complexity consisting of 5 DW potentials producing 15 correlated features, I observed a similar performance of the two ML methods (Fig. 3.6). GBDT correctly captured and ranked the top three features (#8, #25 and #35). However, most other important features scored a FI value very close to 0. Out of 15 correlated features, GBDT did not identify 12 of them with high FI, whereas the MLP captured all of them. However, the MLP's AD did not rank the top four features in the correct order, scoring the 3rd most correlated feature with the biggest AD.

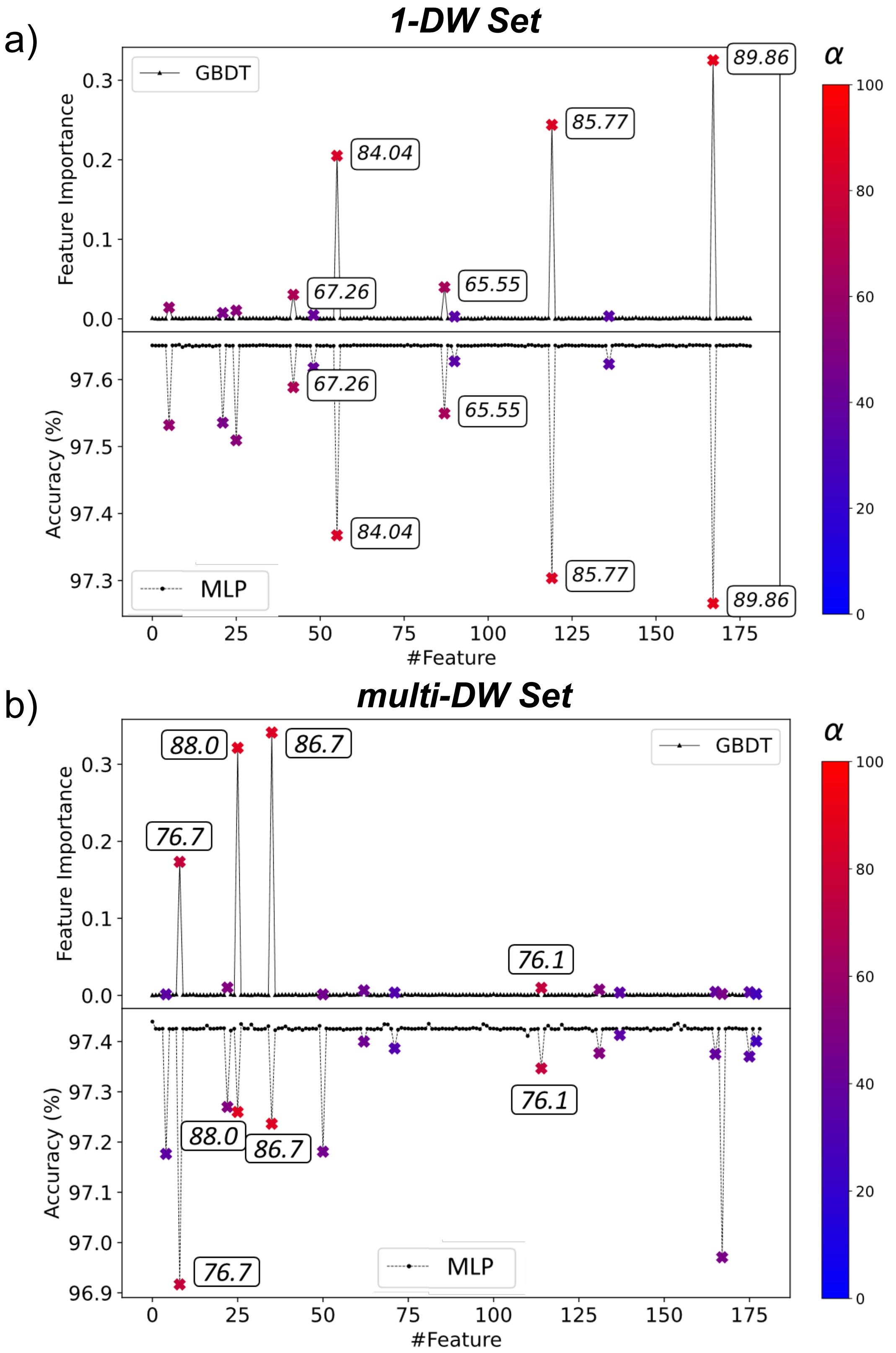


**Figure 3.6:** Comparison between GBDT (top) and MLTSA with NN (bottom) feature analysis methods for the *1-DW* (left) and *multi-DW* (right) datasets. Correlated features are marked from blue (0%) to red (100%) depending on the mixing coefficient, $\alpha$ (x symbols, color scale on the right, five highest mixing coefficients also displayed for the datapoints). Uncorrelated features (small black symbols) are at 0 FI for GBDT and show no loss of accuracy for MLTSA with MLPs. Correlated features all show a significant AD for the MLP, while only the top correlated features have high FI using GBDT.

Considering both analytical models, I found that whereas GBDT has a higher specificity to rank the top correlated features in the correct order, MLP has a higher sensitivity and captures all correlated features but cannot necessarily identify the highest ranked ones quantitatively using the AD as the measure. I have observed this behaviour through the different datasets and tests. Therefore, a combination of the two ML methods can further help identify the most important features and its most advised. In more complex systems, this performance might not be directly generalizable due to the simple linear correlation of the CVs of this model. MLP might outperform at this task for more complex relationships as seen with a higher degree of mixing coefficients in Fig. A.1, however, GBDT results show that it fails at ranking correctly the most correlated feature while MLP ranks it correctly and highlights more relevant features again. In this case the higher degree of mixing, i.e. increasing the number of potentials involved, makes the data noisier and less correlated in the end at it is not reliable enough for validating selecting features from much more complex data. The lack of current toy models to validate does not allow for testing this further. In order to overcome this, a more complex model consisting of a 2D surface potential was created.

## 3.4 2D Analytical Model

### 3.4.1 Dataset generation

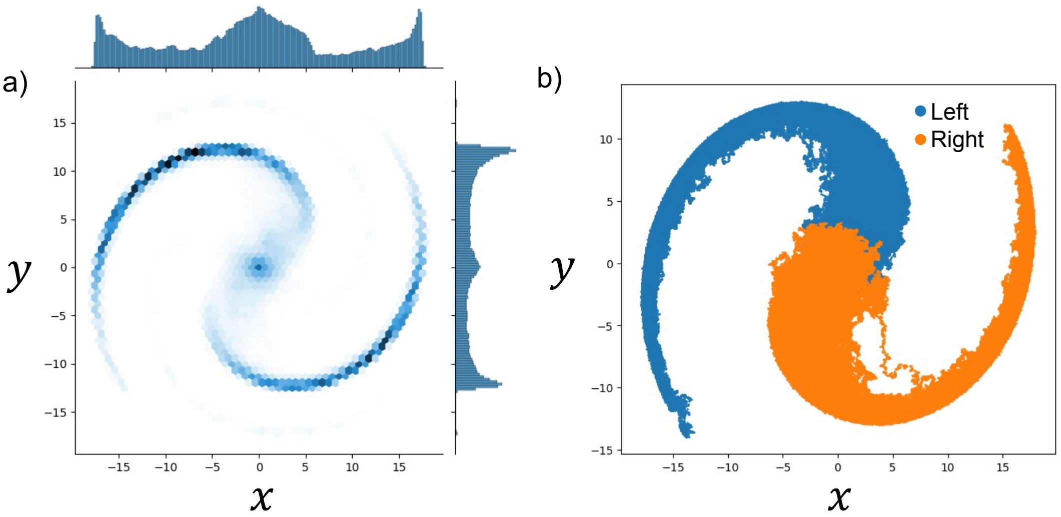


**Figure 3.7:** a) Plotted distribution of the values for the trajectories generated on the spiral potential. b) Labelled classes for the downhill spiral trajectories.

In this dataset, trajectories start from the coordinate origin (0,0) and quickly converge towards one of the arms of the spiral. Even though this spiral can have more than two arms (see Fig.), for the purpose of this work and to follow the context within ligand-unbinding, two final outcomes will be assumed similar to a two-state based process (see Fig. 3.7), in this case being left (IN) and right (OUT). Once the trajectories are generated and labelled as two different classes, in a similar fashion to the 1D dataset (see section 3.2.1 and 3.2.1), new input features are generated from the original data. In this case, the original $X,Y$ features are projected onto a new angle-based $X$ axis causing a rotation as in Fig. 3.8. After that, only the $X_{proj}$ is used as a new input feature. This is done for 72 different angles (every 5 degrees). Unlike the 1D analytical model dataset where the relevance of a feature is the same throughout the whole simulation and it is independent of the time, the 2D has two most describing features (angles) for a period of time that can already separate the data in two classes. These most describing features change through time, which adds more complexity and the ability to asses if MLTSA is able to capture this changes through time i.e. looking at a portion of time shows the relevance of that exact moment which leads up to the final outcome.

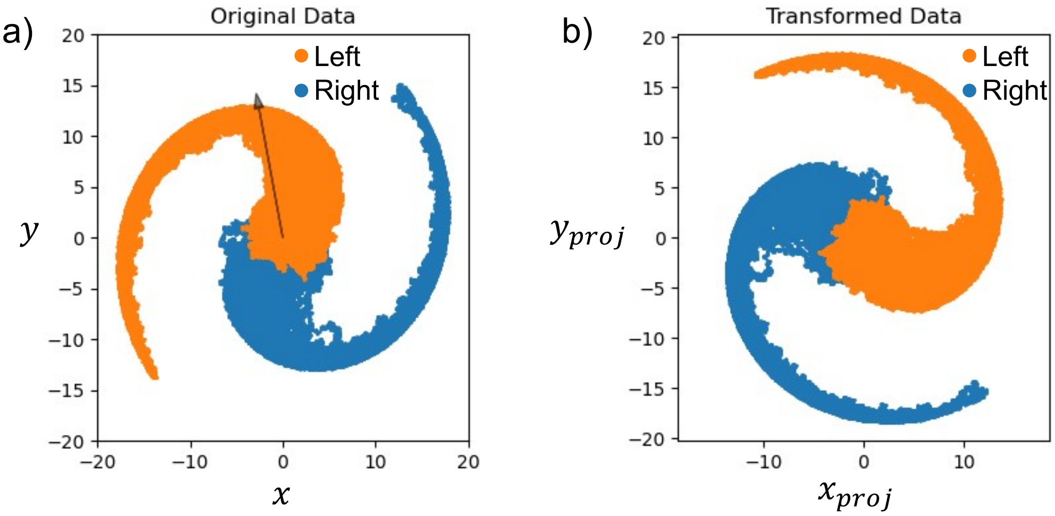


**Figure 3.8:** a) Generated coordinates $(X,Y)$ from the 2D spiral potential. b) New projected coordinates $(X_{proj}, Y_{proj})$ after the angle-projection transformation

## 3.4.2 Search for the right time

In an analogous way to the 1D analytical model, the 2D model also relies on searching for appropriate time ranges where the classification task is not trivial, yet achievable. For that purpose, a preliminary training of 10 independent replicas with both MLP and GBDT at different time-frames throughout the simulations, allowed us to highlight the underfitted and overfitted regions (Fig. 3.9) and pinpoint the regions available for training.

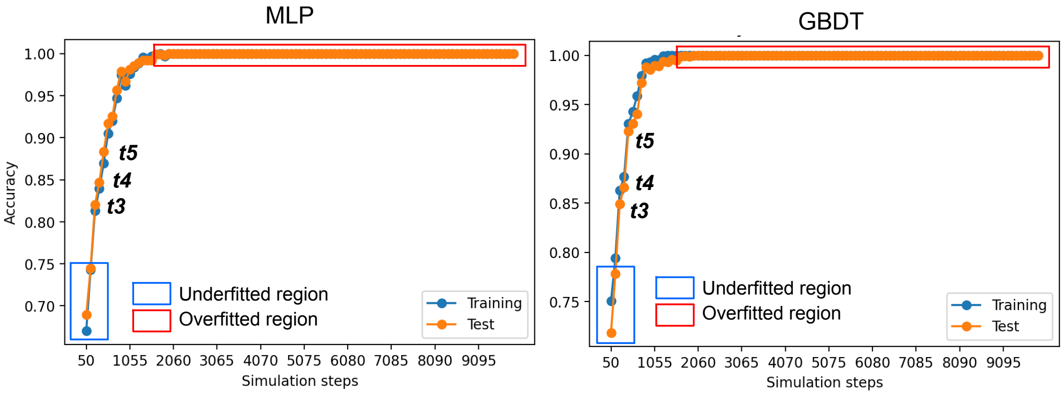


**Figure 3.9:** MLP (left) and GBDT(right) accuracies at different times (every 50 frames). The regions where the model is considered to be overfitted and it can no longer be used for feature analysis is highlighted in red, whereas the underfitted one, where the model is unreliable is highlighted in blue.

**Similar performance.** Overall, both MLP and GBDT performance have a similar accuracy on the same time-frames. In this setting it is still important to find a suitable time frame for both. Thus, the following experiments were conducted in 3 time-frames to check if the most relevant features change with time. The time-frames are the 3rd, 4th and 5th datapoints from Fig. 3.9 which correspond to the 150th-200th (t3), 200th-250th (t4) and 250th-300th (t5) frames.

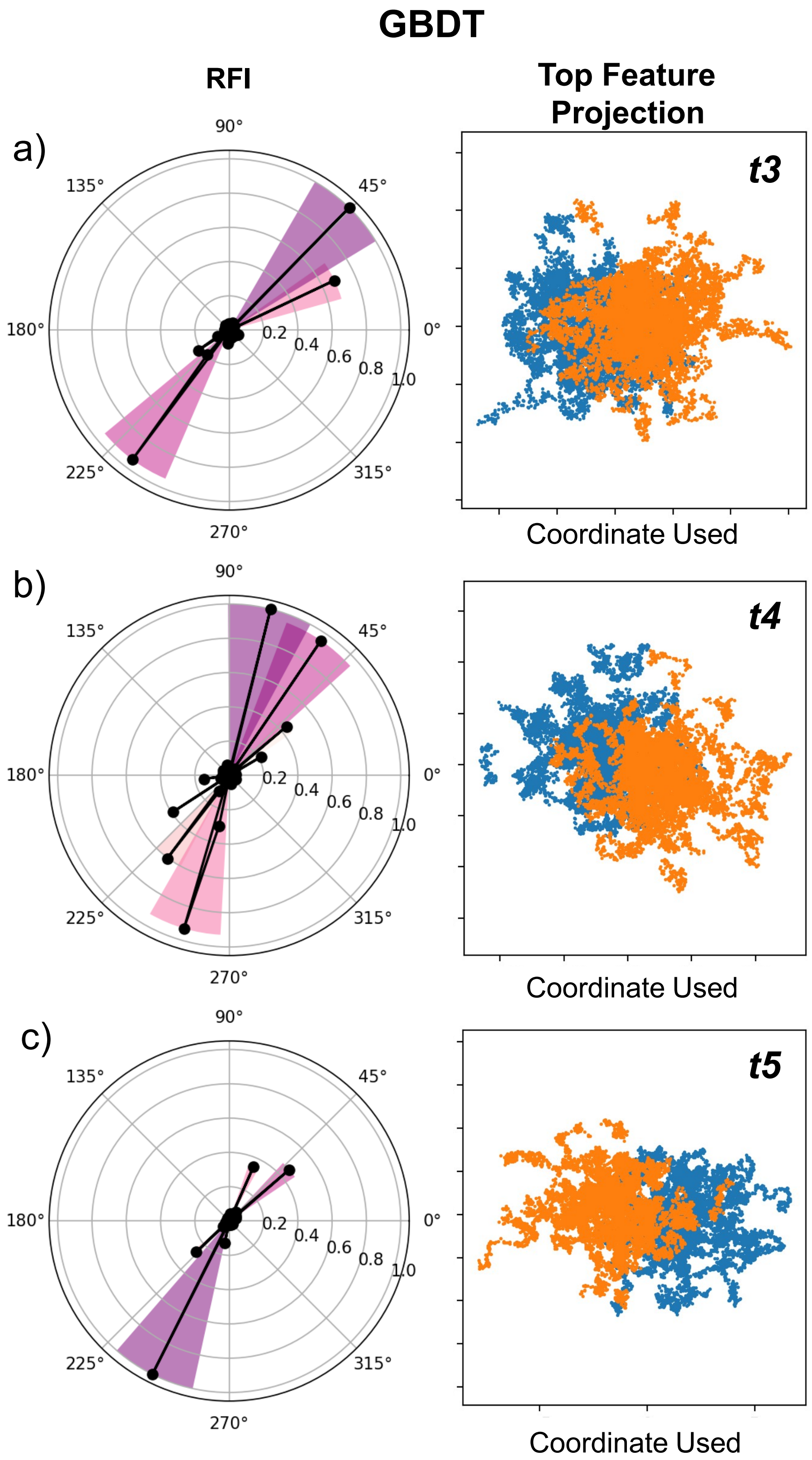


**Figure 3.10:** Feature Analysis results for GBDT using FI. The polar plots represent the relevance of the angle projections (as input features) using their relative FI (RFI) from 0 importance to 1 most-important. In shades, the top 5 relevant features, the higher the value the darker the shade. On the right, a plot of the data for Left (blue) and Right (orange) labels on the most relevant projection for the model. The plots correspond to the time-frames t3 (a), t4 (b) and t5 (c). Note that the X coordinate marked as ”Coordinate Used” is the value of the represented projection and Y is not used in training, but for visualization purposes.

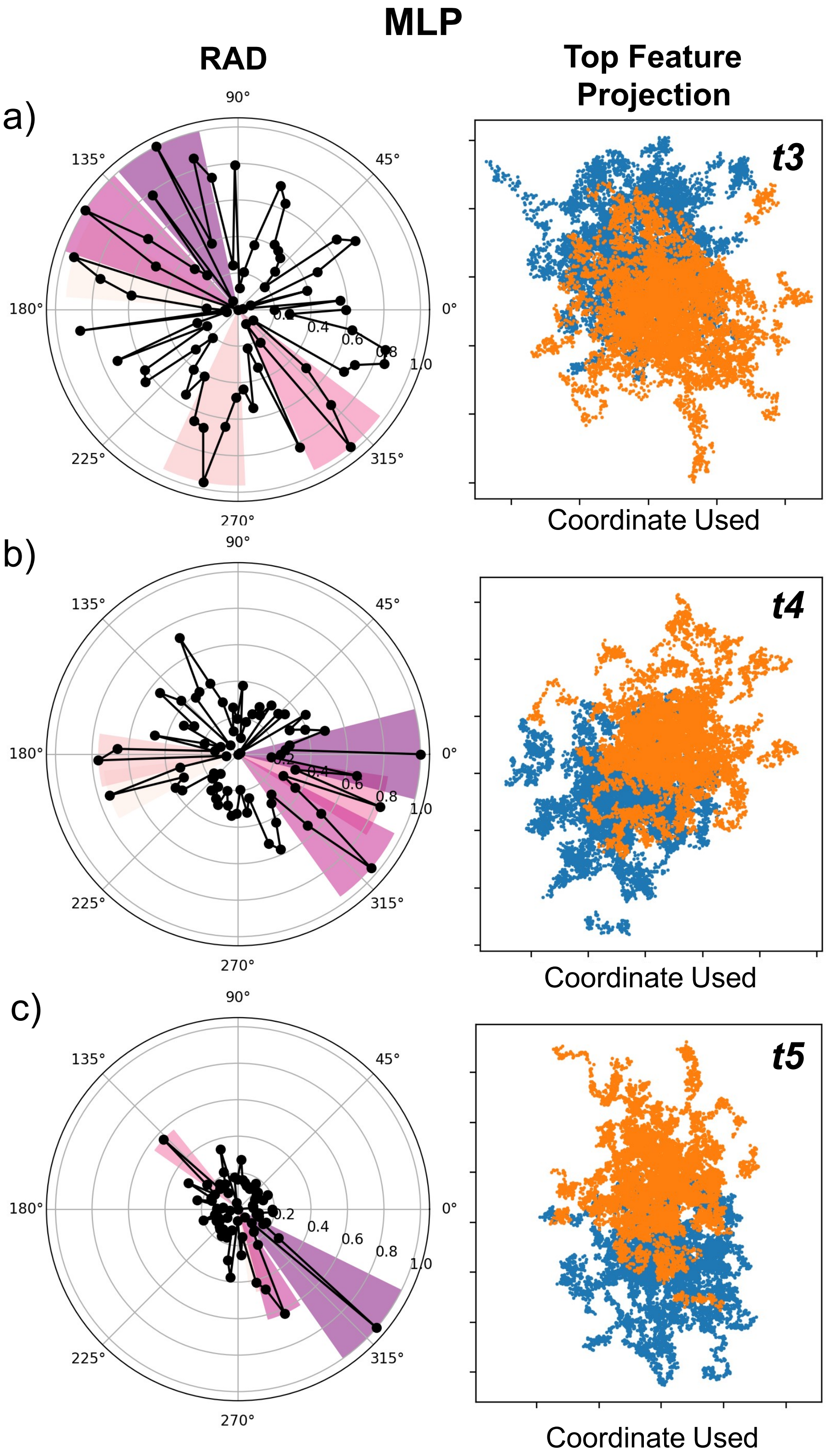


**Figure 3.11:** Feature Analysis results for MLP using AD. The polar plots represent the relevance of the angle projections (as input features) using their relative AD (RAD), re-normalized from 0 (least important) to 1 (most-important). In shades, the top 5 relevant features, the higher the value the darker the shade. On the right, a plot of the data for Left (blue) and Right (orange) labels on the most relevant projection for the model. The plots correspond to the time-frames t3 (a), t4 (b) and t5 (c).Note that the coordinate used is only the X in the represented projection.

### 3.4.3 Feature Analysis

The resulting feature analysis for both GBDT and MLP at t3, t4 and t5 is portrayed in Figs. 3.10 and 3.11. Note that only the $X_{proj}$ is kept for training, so the most relevant feature thus, is the one that can best separate the two classes with the most ease when looking only at that value, i.e. simply a cutoff on the distribution. That is why the Jaccard Index becomes valuable for evaluating the performance on both models.

For both models, the most relevant features are different in each time-frame, which is expected. However, there is a big difference between the pinpointed features from GBDT and MLP. GBDT is able to pick up top features clearly whereas MLP has a very noisy importance polar plot, especially at earlier times like t3 Fig. 3.11.a. On the contrary, GBDT has way less relevant features and they are highlighted more clearly through all times (See Fig. 3.10. In addition, when looking at the projected top features for both and having in mind that the X coordinate is the only used for training, GDBT excels at finding the input features which separate the data in two the most, as seen in Fig. 3.10.

**Table 3.3:** Resulting Jaccard Indices for the top features selected for each model.

| Time-Frame | MLP-Top Feat<br>Jaccard Index | GBDT-Top Feat<br>Jaccard Index |
|---|---|---|
| *t3* | 0.581 | 0.584 |
| *t4* | 0.535 | 0.627 |
| *t5* | 0.626 | 0.662 |

**Who performs best?** GBDT seems to be able to capture a feature close to the most descriptive one, together with the features opposite to the best projected angle, which are equally descriptive in an inverse relationship. This is noticeable looking at the overlayed plots directly from the data in Fig. 3.12. Keep in mind the features translate in order directly to projection angles, from 0 to 360. When comparing the Jaccard Index for the top features of both MLP and GBDT, however, MLP's guesses have a slightly lower value, which means that the feature is better at separating both distributions. Which of the two model is more suitable for this remains is still unclear, however GBDT has a more robust and clear feature selection, which favours it towards real-life applications. Note that the sprial dataset is closer to a complex problem such as ligand-unbinding since the importance of different features through time changes and quick oscillations on the same direction between small values still lead to different outcomes. However, once more, the usage of both models for analysis is advised, as proven with the 1D analytical model results and with the protein data applications [1, 2].

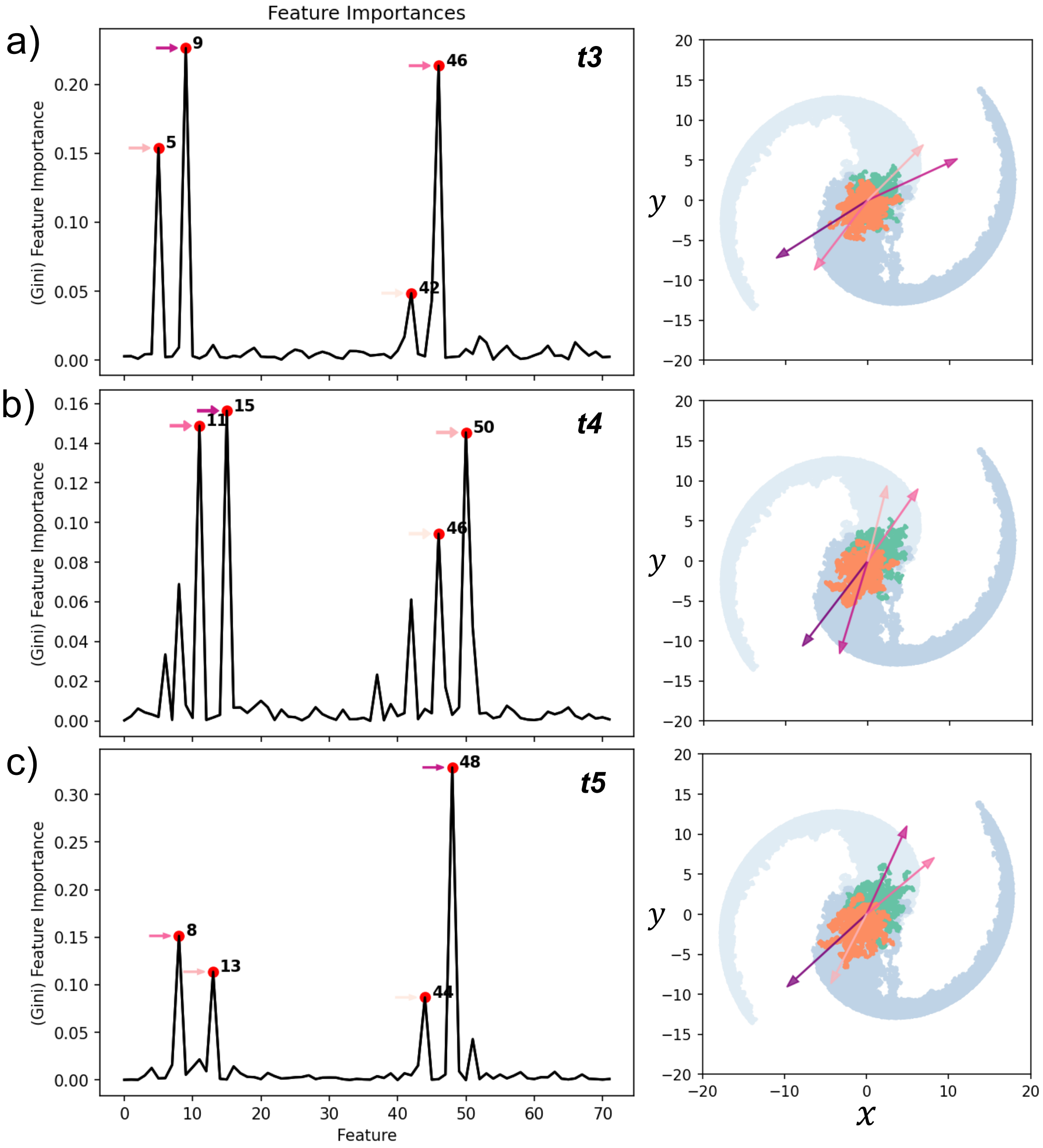


**Figure 3.12:** Results of the MLTSA on the selected time-frames for the Spiral Dataset. These show the GBDT FI next to the original data and the most relevant features (in shades of purple) for the t3 (a), t4 (b) and t5 (c).

## 3.5 LSTM attempts

Once the capabilities and limitations of the GBDT and MLP for the spiral dataset have been explored, the scope of the project turned towards a more complex neural network architecture such as the **long-short-term memory** (LSTM) neural network. LSTM is a powerful approach due to its unique capabilities in handling sequential data and capturing temporal dependencies. The sequential nature of the spiral dataset makes this model well-suited for capturing patterns and trends. Also, the **temporal dependency** from the spiral dataset, where the current datapoint's position is influenced by previous data points, and the complex patterns it can contain are a good match for the LSTM memory span and generalization capabilities. The inherent memory and feature extraction mechanisms of an LSTM makes it well-suited for tackling the challenges posed by spiral datasets, ultimately leading to improved performance in tasks such as prediction, classification, or pattern recognition. Experimenting with this model would benefit the field by bringing insights on the MLTSA's capabilities while being able to assess the capabilities of LSTMs on spiral datasets.

### 3.5.1 1D Analysis

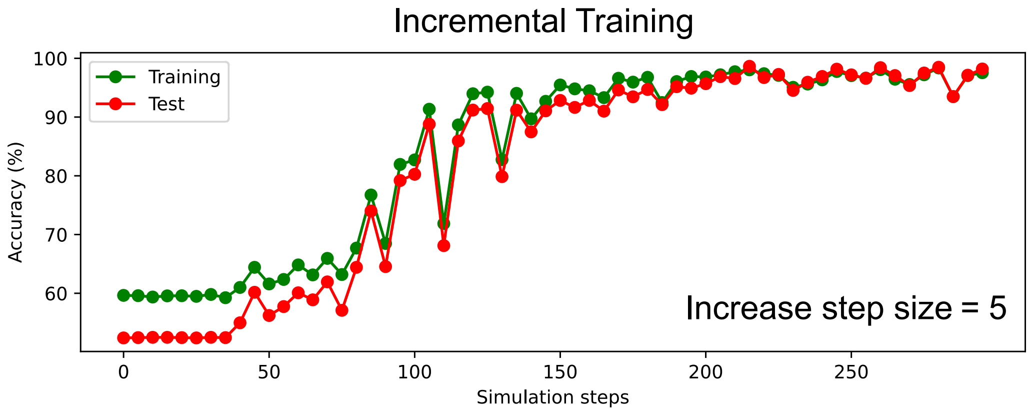


**Figure 3.13:** Training and testing accuracy for the LSTM Model when increasing the size by 5 each time until 300 simulation steps.

**Incremental Training** . Taking advantage of the LSTM's architecture, I explored the accuracy of the model training at different times, increasing by a step size of 5 each time up to 300 simulation steps.Note that in this case the test set is a validation set with completely new run simulations. The results of this incremental training can be found in Fig. 3.13. The model struggles to predict the outcome at early times, with very low accuracy, starting at 55% and staying under 80% up to having data from the 0th to the 100th step. This is a major difference with simpler models like GBDT and MLP, where with little data from the 30th to the 60th step the accuracy was already high. However, as seen in Fig. 3.5, the simpler models are very sensitive to the amount of data used, and more importantly, the task is hard at times earlier than 100 simulation steps. All of these facts suggest that the behaviour of this model is different than that of the simpler models, and its training strategy has to be different.

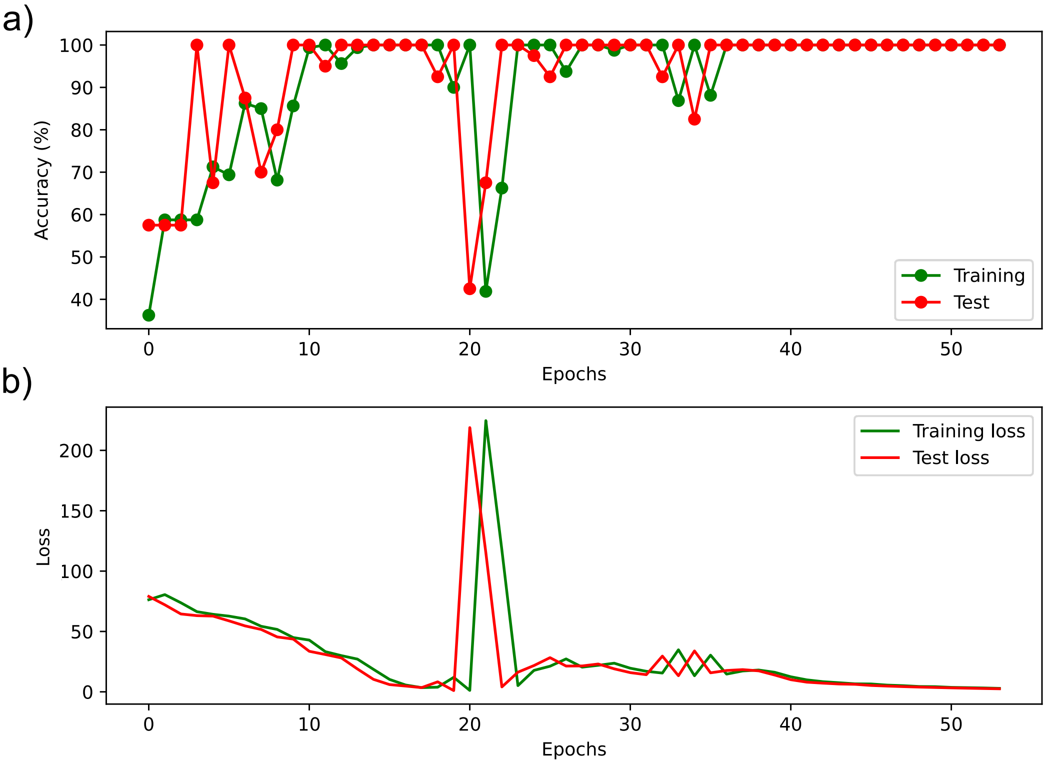


**Figure 3.14:** Evolution of a) testing and training accuracy through epochs and b) training and testing loss through epochs.

**Learning the trajectory.** The model was further trained on all of the data from beginning to end (500 steps). The evolution of the accuracy and loss through the epochs can be found in Fig. 3.14. The model quickly reached convergence in less than 100 epochs and was able to perform 100% accuracy on the training data and

100% accuracy on the test data (new simulations). Although the learning speed was not expected, having all of the data available makes it easy for the model to learn to discern one class from the other. However during the training epochs, one steep decrease/increase in accuracy/loss at around 20 epochs shows us that some steep barriers can be found in the error landscape, and that possibly a too big learning rate was being used. Different learning rates should be explored in future works.

**Feature Analysis.** As seen in Fig. 3.15, unlike the simpler models (MLP/GBDT), when the LSTM is trained with the full trajectory it is able to pinpoint relevant features. Although not all picked up features are relevant, its performance is impressive.

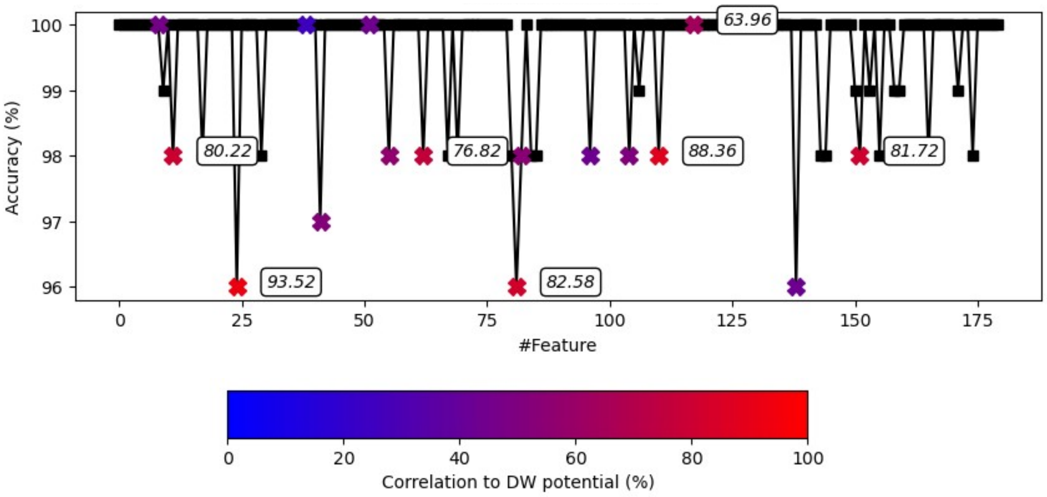


**Figure 3.15:** AD plot of the LSTM model trained on full trajectories. Features correlated are marked with an X, the color corresponds to the level of correlation. Features above 60% correlation have their score written.

**Training at the right time?** When the model is trained for the same amount of simulation steps (30 to 60), it takes much longer to train, and it is hard to converge, but it achieves 87% accuracy on validation. Results for the training attempts are shown in Fig. A.2, the AD is also shown, which captures very few features. However, this could mean the model is not converged at all, and further investigation is needed with this more complex architecture.

## 3.6 Conclusions and Future Work

The experiments performed in this work have shown that MLTSA is useful for selecting relevant features at given time-frames. In addition the 1D analytical model has validated they both are able to select correlated features with ease.

The 1D analytical model has proven useful for assessing time-series classification tasks, and its easy implementation and data generation make it suitable as a validation test for other approaches.

On the same tone, the 2D analytical model, the Spiral implementation in particular, has proven challenging enough for both ML models. Moreover, it has aided in exploring the usage of MLTSA for time-changing importances, a much more closely related case to that of protein data. It has proven a good exercise for validation, which is advised not only for similar approaches, but for other classification tasks and time-series approaches, particularly modifying the number of arms, thus increasing complexity and labels. This is already available in our code implementation for anyone to test.

From the performed experiments in both 2D and 1D models and their results, it seems both models (MLP and GBDT) are good for testing other types of forecast and prediction, as well as classifications. Additionally, the ML models performances keep suggesting that a combined approach works best for finding relevant features in complex time series data.The LSTM exploration is still preliminary and more tests are needed to asses its usage for feature analysis in time-series. There is an improvement over the simpler methods, where there is no need to find the right time-frame to train at, which is an improvement over the fully automation of the protocol.

Regardless, there is till plenty of room for improvement and exploration in this field, other potential implementations are yet to be explored, as well as other ML models or deep learning architectures, such as transformers. Further experimentation is advised as well as improving upon what's being built.

Chapter 4

# Understanding Kinases: Unbinding CDK2 Inhibitors

This chapter is mostly from our first published work, containing the methodology for our iterative **unbinding protocol** as well as the **MLTSA** protocol for understanding unbinding paths from inhibitors of **Cyclin-dependent kinase 2** (CDK2). This was the preliminary application of MLTSA and it validated its capability to detect relevant residues for the unbinding paths of three different ligands. After this, the follow-up publications optimized the protocol to get the most information out of the analysis, however, this was the first biologically relevant system it was applied to. Hence, it is more methodology focused and it doesn't delve too deep in the MLTSA results.

## 4.1 Introduction

**Drug-Target residence time and its relevance.** In drug design, long residence time is now considered just as relevant as having a strong binding affinity [62]. The residence time refers to the duration that the ligand stays bound in the binding pocket, and it is closely related to the overall unbinding process rate. Obtaining information about the high energy transition states and free energy barriers associated with this process is difficult [10, 63]. Even if a drug has a high binding affinity, a short residence time can significantly decrease its efficacy [9]. Recent studies have shown that in some targets, the kinetics of drug-receptor binding may be more important than affinity for drug efficacy [10]. The drug-target dissociation process can also be complex and involve several steps and pathways. As a result, drug candidates with high affinity but low residence time may be discarded in the drug discovery process [64, 65].

**Experimental determination is challenging.** Drug discovery faces the challenge of predicting kinetics of ligand-protein interactions with a fast and reliable method [66]. However, before experimental determination of ligand kinetics, ligands first need to be synthesized, which can be expensive and time-consuming, even for a moderate number of compounds. To obtain kinetics of ligand-receptor unbinding, various experimental methods have been used, such as radioligand binding assays, fluorescence methods, chromatography, isothermal titration calorimetry (ITC), surface plasmon resonance (SPR) spectroscopy, and nuclear magnetic resonance (NMR) spectroscopy [65, 67]. However, these methods can suffer from interference (especially fluorescence), lack of accuracy for short residence times, and high cost/hazard in the case of radioligands [68]. Among these methods, SPR is the most widely used to measure rate constants associated with the unbinding of ligand-receptor ($k_{on}$ and $k_{off}$). This method is label-free; however, the attachment of the protein to the probe may influence the activity of the protein, due to conformational changes [68]. To overcome these difficulties, various computational techniques have been proposed as alternatives to estimate the kinetics of unbinding events, providing a screening approach [69, 70].

**Computational Strategies.** Molecular dynamics (MD) is a powerful computational tool used to study biological processes like protein-ligand interactions at an atomistic level. [3] Unbiased MD simulations have been successfully used in drug discovery, either with multiple short simulations or specialized computer architecture. However, due to limited timescales, obtaining sufficient statistical sampling for accurate calculation of kinetic and thermodynamic properties can be challenging. Drug-protein unbinding processes occur on long timescales, ranging from milliseconds to hours, making it difficult to simulate some drugs with half-lives of hours like Aclidinium, Deoxyconformycin, or Tiotropium [71]. Enhanced sampling methods are therefore required to accelerate simulations and sample rare events [72].

**Attempts at ligand unbinding biasing CVs.** Enhanced sampling techniques have been developed to accelerate simulations and sample rare events, in order to predict free energy barriers and uncover biological kinetics [73, 74]. These methods include free-energy perturbation, metadynamics (MetaD), temperature-accelerated MD (TAMD), steered MD (SMD), and more [12, 54, 75–80]. A key factor in these methods is the identification of a collective variable (CV) that represents a physical pathway for the calculation of the free energy profile [81]. However, finding appropriate CVs can be challenging, with few practical ways to build them [54, 55, 82].These techniques have been previously used for ligand unbinding, such as using MetaD to predict the unbinding of p38 MAP kinase bound to type II inhibitors [83], and using a combination of MetaD and QM/MM simulations for more accurate prediction of kinetics [84]. Steered Molecular Dynamics (SMD) was used to calculate the residence times of Sunitinib and Sorafenib in complex with the human endothelial growth factor receptor 2, and to calculate the unbinding free energy profile for TAK-632 and PLX4720 bound to B-RAF [84, 85]. However, the predicted free energy barriers for unbinding were significantly lower than experimental data in both cases. In order to produce accurate free-energy profiles using biased simulations with important degrees of freedom, one needs to define an ideal set of collective variables (CVs) that map the full path of the reaction coordinate [86, 87]. Usually, the vectors that describe this manifold are selected based on a priori chemical/physical intuition, which may neglect essential interactions occurring during the unbinding process and significantly affect the free energy calculation. Additionally, resolved structures may not always reflect appropriate conformers for ligand binding.Note that in pursuit of defining this ideal set of CVs to map the full trajectory, the unbinding protocol pretends to automate a part of it, while the MLTSA analysis refines the existing one and aids in discovering new ones.

**Our approach for obtaining unbinding profiles.** In this work, a novel enhanced sampling method to obtain accurate free energy barriers for ligand-protein unbinding and identify key molecular features determining the unbinding kinetics is proposed. The proposed iterative method assigns CVs during the unbinding trajectory, and then uses these CVs as the driving force to pull the ligand out from the pocket and perform the sampling for accurate free energy calculations. Unlike existing methods, such as $\tau - RAMD$, there is no need to a priori select CVs or decide what to bias; these naturally arise from the unbinding trajectories that take the flexibility and dynamics of the system into consideration.

The CVs extracted from the trajectories sufficiently describe a full pathway for the unbinding process, and are subsequently optimized in the space of the identified CVs to obtain a minimum free energy profile using the finite temperature string method [41]. While different unbinding trajectories may lead to slightly different variations due to multiple local minima along the paths, the main transition state ensembles are typically captured by all of these paths similarly after convergence to the minimum free energy pathway [19, 49] The results show little variation in the unbinding free energy barriers using different starting pathways for free energy calculations.

**Machine Learning methods to understand unbinding processes.** In addition to determining unbinding rates, the work aimed to identify key molecular descriptors that provide guidance for further drug design based on improved residence times. A systematic approach to identify key low-dimensional sets of internal coordinates using machine learning (ML) approaches was also proposed. Machine learning methods have been widely successful in multidimensional data-driven problems, and they are also applied to biomolecular simulations to determine key CVs [88–90]. In this work, I developed a novel approach that makes use of the obtained string unbinding path and taking advantage of that, its transition state (TS) ensemble. I explored two different ML methods in this study: Neural Networks (NN) [91], which provide efficient training on complex high-dimensional data, and Gradient Boosting Decision Trees (GBDT) [92], which allow straightforward evaluation of feature importances (FI) [93]. I generated unbiased ”downhill” trajectories initiated at the TS and used these to train a ML model that predicts the fate of binding or unbinding.

In order to demonstrate the accuracy and effectiveness of our approach on complex biomolecular systems, the free energy barriers for two ligands were calculated. The ligands bind to Cyclin Dependent Kinase 2 (CDK2) with PDB IDs of 3sw4 (18K) and 4fkw (62K) (as shown in Fig. 4.1) [94]. CDK2 is a vital regulator in the growth of eukaryotic cells, and the deregulation of CDK2 has been linked to unscheduled cell proliferation, leading to the progression and aggressiveness of cancer [95, 96]. Therefore, CDK2 is an appealing target for treating specific tumors of particular genotypes [97]. Various molecules, including AT759 [97], AG-024322 [98], Dinaciclib [99], Roniciclib [100], and Milciclib [101], are currently being clinically evaluated as CDK2 inhibitors for cancer treatment. In addition, CDK2 is an excellent benchmark system due to its small size and well-documented kinetic data for binding different molecules [94]. There were no previous computational studies on the kinetics of ligand unbinding on CDK2 prior to its publication date, although up to this thesis one study has been published [102]. This study is based on the MM-GBSA approach, and studies different inhibitors than our paper.

## 4.2 Computational Details

### 4.2.1 Simulating CDK2

All MD simulations were carried out in NAMD 2.12 [103], using the AMBER ff14SB force field for the protein [104], and using the general Amber force field (GAFF) for the ligands[105].

Our unbinding method is illustrated algorithmically in Fig. 2.2. More details on how the protocol is applied can be found at section 2.2.1. An explorational unbiased MD simulation of at least 20 ns was performed to identify the initial interactions between the protein and the ligand in the bound state. These initial simulations allowed to define the first set of CVs describing all distances between the heavy atoms of the ligand and the heavy atoms of the protein smaller than $d_{in} = 3.5Å$, the interaction cut-off. The identified interactions will generate a single one-dimensional CV as the sum of these M distances, $d_i$, and will be used for iteratively biasing the simulations to observe an unbinding trajectory.

At every iteration, the bias is defined as a harmonic restraint, such as 2.3, where

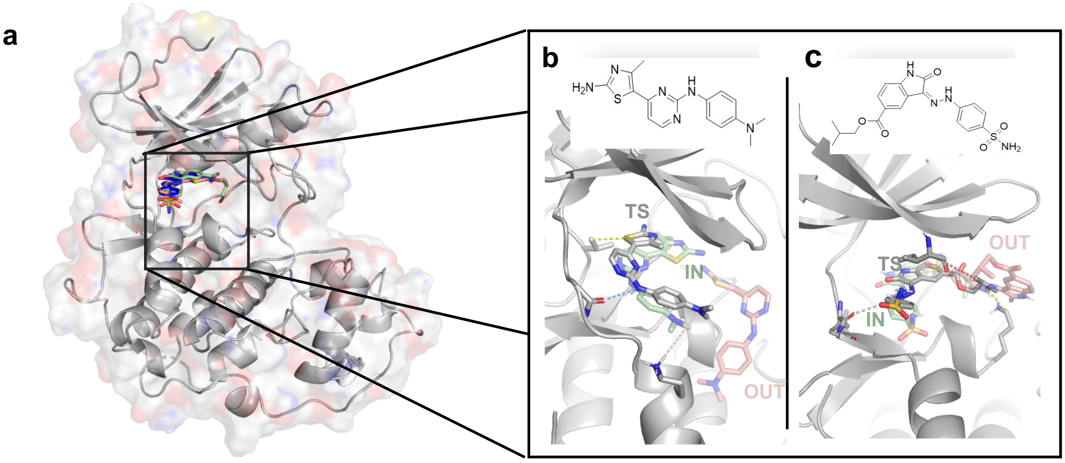


**Figure 4.1:** Illustration of the simulation system, a CDK2 complex bound to two different ligands: b thiazolyl-pyrimidine derivative (18K) and c carboxylate oxindole derivative (62K), originated from PDB structures 3sw4 and 4fkw, respectively. Structural details of the ATP pockets are shown for the two systems (bottom), with the ligands in the bound (green sticks), unbound (red sticks), and transition states (grey sticks). Dashed lines depict key interactions.

$D = D_0 + (Md_{tar})$. Here, one aims to reach the target value D for the 1D CV starting from the initial value at the beginning of the nth iteration $D_0$. $d_{tar}$ is the incremental factor, set to 1 Å, representing the average increase one aims to achieve per distance for the next iteration. The targeted D value will be reached progressively within the next 10 ns long MD simulation for every iteration. The force constant, $k$, was set to $20kcal/mol/Å^2$.

At the end of each iteration, the biased trajectory was analyzed, and novel interactions were identified, within $d_{in}$ of the ligand, that are present for more than half of the total simulation time (i.e., 5 ns). These novel interactions are then added to the list of interactions that define the main CV for the next iteration. Additionally, it also re-evaluates existing interactions. If a distance during the last 5 ns of the trajectory exceeds $d_{out} = 6$ Å or its variance exceeds $dvar = 1Å$, then the distance is removed from the main CV in the next iteration. This exclusion factor will ensure that once a protein-ligand atom pair distance has exceeded $d_{out}$, and therefore there is no significant interaction between these atoms, this interaction is no longer biased. Similarly, loosely interacting atom pairs have higher distance fluctuations, and thus the corresponding weak interaction does not need to be included in the bias.

To reduce the number of interactions between the ligand and the protein and to remove redundancies, one combines atoms that are part of an equivalent group where a rotational degree of freedom can interconvert the atoms from one to the other (for example, benzene ring or carboxylic groups). Here, I considered the centre of mass of that functional group and not the individual atoms.

The iterative process will end when no more distances are present in the main CV from the last iteration n, thus there are no more stable interactions between the ligand and the protein, suggesting that the ligand is outside the binding pocket.

### 4.2.2 Free Energy Calculations

Once the ligand is outside of the binding pocket, to determine the minimum free energy path for the unbinding trajectory, the finite-temperature string method was used [41]. The initial path and the full set of distances (CVs) are taken from the obtained unbinding trajectory [41, 106]. The CV values were extracted for each interatomic distance along the initial unbinding path to construct the minimum free energy unbinding pathway iteratively, building a string of 100 windows in the coordinate space. For each window and each CV, a positional restrain equidistantly along the initial fitted string was applied, using a force constant of 20 $kcal/mol/Å^2$. Biased simulations were performed using these restraints for a total time of 5 ns per window. From the obtained set of trajectories, a high-order (8) polynomial fitting is applied using the average values for each collective coordinate to build the subsequent set of refined CV positions. The procedure is carried out iteratively until the convergence of the free energy profiles and the unbinding pathway. This is verified by ensuring that the maximal change of each CV between subsequent iterations is below 7% (or 0.3 Å) from the previous iteration. By adding multiple overlapping biasing potentials along the dissociation pathways which are parametrized via the identified CVs, the string simulations can sufficiently sample the high dimensional path describing the full unbinding trajectory in detail. The combination of the novel identified CVs with the finite temperature string method allow to fully describe the pathway and recover the free energy profile thus contributing towards an almost fully automated protocol for ligand unbinding. Finally, to obtain the corresponding potential of mean force (PMF), the simulations were unbiased using the binless implementation [41] of the weighted histogram analysis method (WHAM) [50]. See section 2.2.1 for a more detailed description.

Note that the method does not aim to calculate binding free energies or $k_{on}$ rates. These would require simulations of a completely dissociated ligand and protein system, for which the string method is not an efficient algorithm. To this aim, routinely used efficient and accurate FEP76,77 calculations can be combined with our method to determine binding free energies and $k_{off}$ rates, respectively, from which the $k_{on}$ rates can be derived.

### 4.2.3 MLTSA

We developed a Machine Learning Transition State Analysis (MLTSA) method to identify novel refined descriptors that determine the fate of a trajectory from the TS, which is applicable to unbinding simulations, but also suitable for other applications as a low-dimensional feature selection method for highly complex processes where a TS region is identified. In this case, the novel molecular interactions between the drug molecule and the protein for unbinding provide key signatures that determine the unbinding kinetics.

I trained the MLP to analyze the model datasets of the downhill trajectories and predict their possible outcome from early on data. The training was performed using the Scikit-learn library [59]. I trained a simple model with an MLP Classifier architecture, using three main layers (input, hidden, and output) with as many input nodes as input features depending on the system of study, fully connected to a hidden layer with 100 hidden neurons and ending in an output layer with one output node each for IN or OUT classifications. The model was optimized using the Adam solver [36] and using the ReLu [107] function as an activation function for the hidden layer. The training was done with a learning rate of 0.001, iterating over data until convergence or upon reaching the maximum number of iterations (500 epochs). Convergence is determined by the tolerance and the number of epochs with no change in loss. When there are 10 consecutive epochs with less than 0.0001 improvement on the given loss, the training stops, and convergence is reached. The same parameters were used for both the analytical model and CDK2 data.

I also tested the GBDT model using the Scikit-learn library as a comparison to the MLP approach. This method provides feature importances (FI) that enable the ranking and identification of relevant features. I trained 500 decision stumps as weak learners for GBDT minimizing a logistic loss function, with a learning rate of 0.1. The criterion for the quality of the splits was the Friedman Mean Squared Error (MSE), with a minimum of 2 samples to split an internal node, and a minimum of 1 sample to be at a leaf node. The maximum depth of the individual regression estimators was 3, without a limit on the maximum number of features to consider as the best split, without maximum on leaf nodes and using a validation fraction of 0.1. The same parameters were used for both the analytical model system and the CDK2 simulations.

For the application of the MLTSA on CDK2, first I identified the approximate TS location by selecting the last simulation frames from the five windows with the highest energy near the TS point of the obtained PMF. From each of these five starting coordinates, I then ran 50 independent unbiased MD simulations, each 5 ns long. I classified and labelled these short 'downhill' trajectories by considering a combination of two key distances (Table B.1), to identify which simulations finish either in a ligand bound position (IN) or in a ligand unbound position (OUT). I then selected the starting structure (i.e., the TS) that provides the closest to a 1:1 ratio of IN and OUT events amongst these trajectories, and I ran 200 additional 5 ns-long unbiased MD simulations with this starting point. I considered all interatomic distances (heavy atoms only) between the ligand and the protein within 6 Å at the TS starting position and determined the values of these distances along downhill trajectories. These constitute a dataset of distances for each simulation trajectory, and I aimed to select the most important features from these with our MLTSA method.

The number of epochs and convergence of the loss function for each model can be found in Tables B.3, B.2 and Fig. B.3. Thus, using the frames coming from the multiple short unbiased MD simulation trajectories starting from the TS, I provided a dataset of distances extracted along the trajectory, as well as the future outcome of the IN or OUT events as the desired answer/classification. I performed the ML training at several different time ranges of the trajectories (Fig. B.2), to observe the predicted accuracy at different time ranges along the simulations. From all the available trajectories for each system I reserve a part for further validation to avoid the overfitting of the model. The rest is used for training, with all frames from the trajectories concatenated and randomly mixed, then split in different fractions as training (0.7) and test (0.3) sets. The trained model is additionally verified to have a similar prediction accuracy on the unseen trajectories.

Using the trained model, one can assess which features are the most important for the model to predict whether the simulation is classified as bound (IN) or unbound (OUT). To do so, we apply our own feature reduction approach (FR), in which every single distance (i.e., feature) is excluded one-by-one from the analysis, and I calculate the drop in accuracy compared to the full set of distances present. Differently from the standard approach [53], where the real value of each excluded feature is replaced with a zero, here one replaces the value for each excluded feature with the global mean of that selected feature across the simulations, thus cancelling the variance of the aforementioned feature. This approach is more in line with the input values explored by the trained model, whereas a 0, specially in physical quantities might be really exceptional and make the model perform in unexpected ways.

## 4.3 Results and Discussion

### 4.3.1 Unbinding CDK2 Inhibitors

For each system, I performed three independent simulation replicas starting from the respective equilibrated system. For each replica, I performed the initial unbiased MD simulation, followed by our unbinding trajectory procedure and subsequently calculated the minimum free energy path and the corresponding free energy profile using the finite temperature string method (Fig. 4.4).

Fig. 4.2 shows a representative result of the unbinding process for selected interactions. The first distance (blue line) is identified from the initial unbiased bound simulation as being shorter than 3.5 Å. Later during the biased unbinding process at

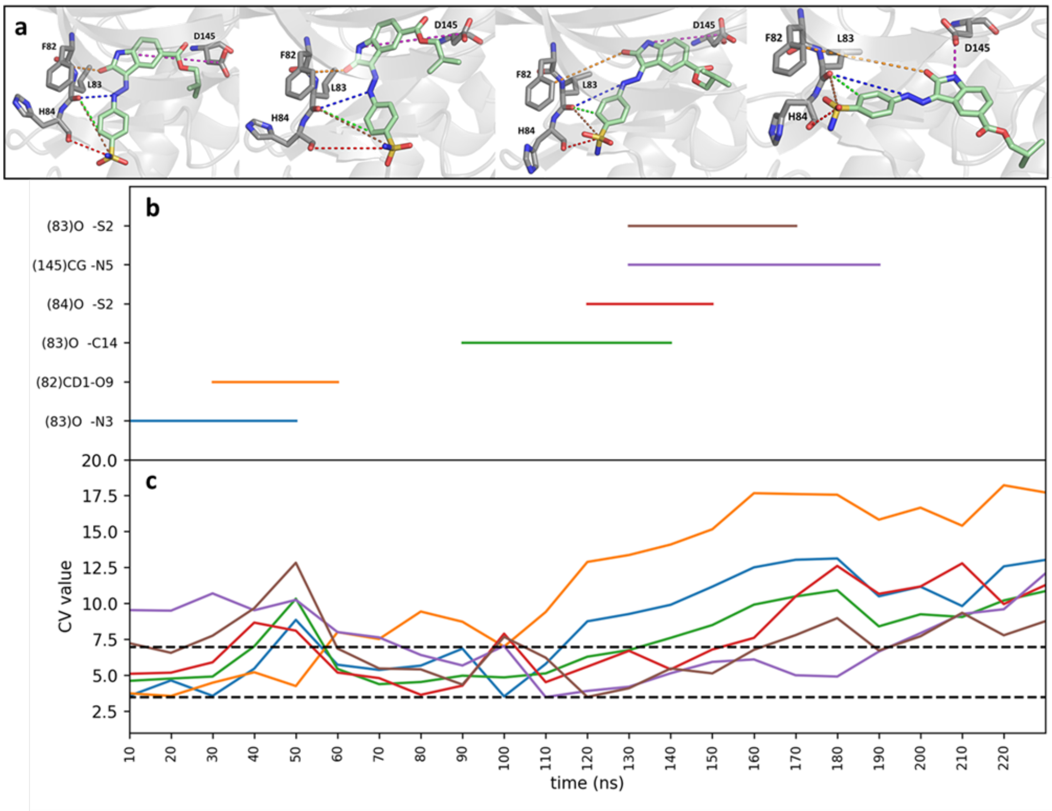


**Figure 4.2:** a) Unbinding trajectory of ligand 62K represented as selected snapshots along the trajectory at 0, 70, 141, and 219 ns from left to right, respectively. Representative distances used for the bias are shown as colored dashed lines. Some of the representative distances included in the CV along the unbinding trajectory are shown in (b), the corresponding distance values to these are plotted in (c). The lower dashed line at 3.5 Å is the cut-off value below which an interaction is included in the main CV, the upper cut-off at 6 Å is the value above which the distance is excluded from the CV.

30 ns a new interaction is found (orange line) and at 90, 120 and 130 ns more distances are included in the main CV (green, red, purple, and brown). Additionally, interactions are progressively being removed as they are breaking (above 6 Å).

Overall, while the identified CVs in different replicas vary, a few common key CVs are present in all unbinding trajectories within all replicas (Fig. 4.3). Even if the actual unbinding pathways have differences amongst the replicas, as seen by looking at the distances found along the paths, they are all expected to pass through the same TS ensemble and show generally the same mechanism. This can also be confirmed from the consistent free energy profiles (See Fig. B.7, also for the 60K/4FKU system).

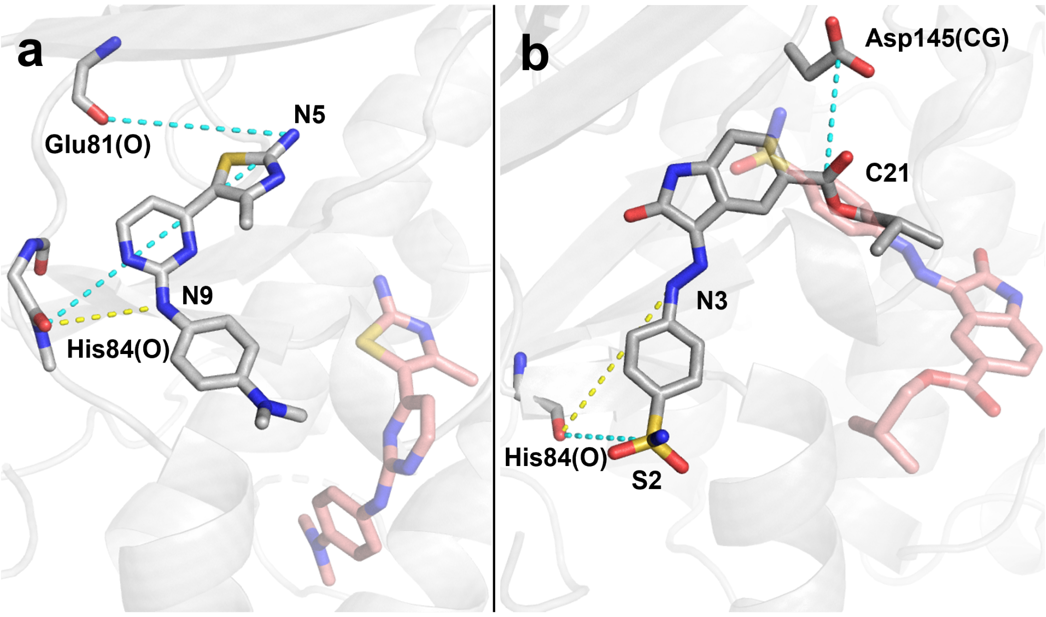


**Figure 4.3:** CVs obtained from the unbinding of 18K (a), and 62K (b); representative distances shown in dashed lines (yellow: interaction from the initial structure, cyan: interaction found during the unbinding trajectory), red sticks represent the coordinate of the ligand when it is outside the pocket. These distances appear in each of the three replicas for each system.

**Table 4.1:** Ligand binding kinetic and thermodynamic values of 3sw4 and 4fkw systems from Dunbar et al. [94] and calculated results obtained from the simulations. $\Delta G_{calc}$ was calculated using the Eyring-Polanyi equation: $k = k_B T/h$ $exp(-\Delta G/k_B T)$ at 298 K. [108]

| **PDB** | **Ligand** | $K_D(M)$ | $k_{on}$ | $koff$ | $\Delta G_{exp}$ | $\Delta G_{calc}$ |
|---|---|---|---|---|---|---|
| | | | **[M-1s-1]** | **[s-1]** | **(kcal/mol)** | **(kcal/mol)** |
| **3sw4** | *18K* | 9.61E-07 | 1.00E+05 | 0.0823 | 18.93(±0.17) | 16.29(±0.21) |
| **4fkw** | *62K* | 4.73E-08 | 6.49E+04 | 0.00261 | 20.97(±0.05) | 20.27(±1.06) |

Additionally, I also performed the unbinding calculations for a third ligand, 60K, that is analogous to 62K (Fig. B.8). Interestingly, I identified that all three replicate string pathways originating from three distinct unbinding simulations present a rotation of the hydrazineyl N=C bond, leading to a cis(Z)-trans(E) isomerisation of the ligand near the TS (Figs. B.9 and B.10). This is due to, on one hand, the initial strong forces in the string simulations that could be avoided in the future, and, on the other hand, to force field inaccuracies with a too low energy of the transform and too low barrier for the related dihedral angle rotation as determined by DFT calculations (Fig. B.11). When compared with 62K, which does not exhibit this behavior in any of the three replicas, one can observe a lower energy for the 60K trans state, that enables it to avoid the TS bottleneck. Correspondingly, all three distinct replicas result in a consistently too low unbinding free energy barrier when compared with experiment (Fig. B.7).

### 4.3.2 Recovering the Free Energy Profiles for the Unbinding

The energy barrier extracted from the PMF of the simulations agrees closely with the experimental $k_{off}$ rates and are very well reproducible within the same system (Table 4.1 and Fig. B.7). The shape of the free energy profiles is also consistent amongst the replicas, however the exact shape depends on the CVs identified in that replica (Fig. B.7 and Table B.4). Generally, a higher number of CVs results in a broader TS region (e.g., Fig. B.7, ligand 62K). In addition, results for the third ligand, 60K is also presented, demonstrating a consistent underestimation of the free energy barrier due to the discontinuity of the dihedral angle along the minimum free energy paths [109].

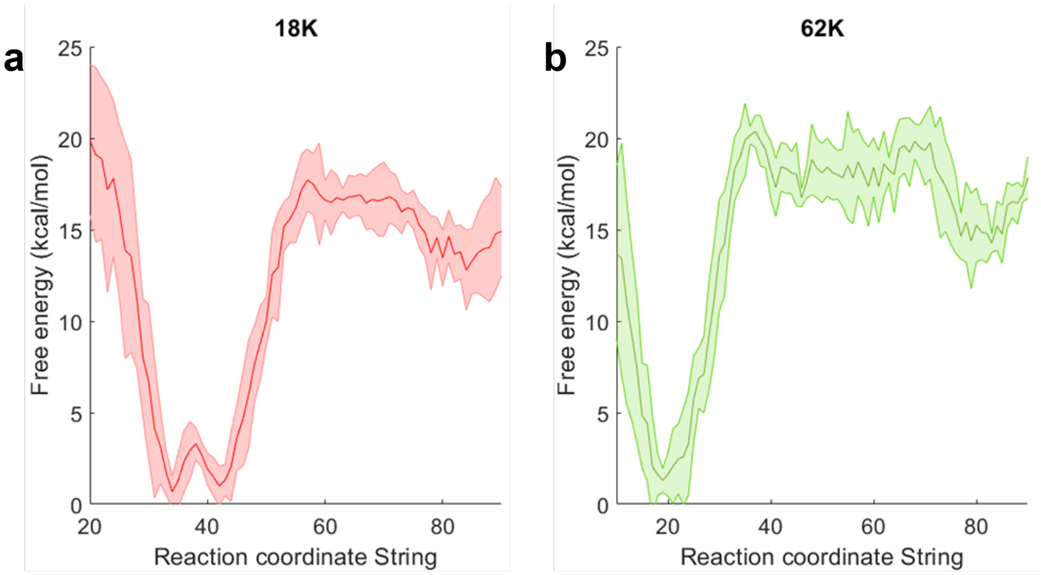


**Figure 4.4:** PMF of the unbinding path for 18K (a) and 62K (b). The free energy profile is obtained from a representative replica, the standard error, shown as a shaded area, was obtained by dividing the full dataset into 4 subgroups and recovering the PMF.

Importantly, comparing the same ligand within the three different replicas in all systems provide very similar free energy barriers, expressed with a low standard error. The energy barriers consistently reproduce the high energy barriers also seen experimentally thanks to the introduction of numerous key CVs that are not only taken from the initial ligand-bound conformation but, instead, introduced along the unbinding paths (Fig. 4.2).

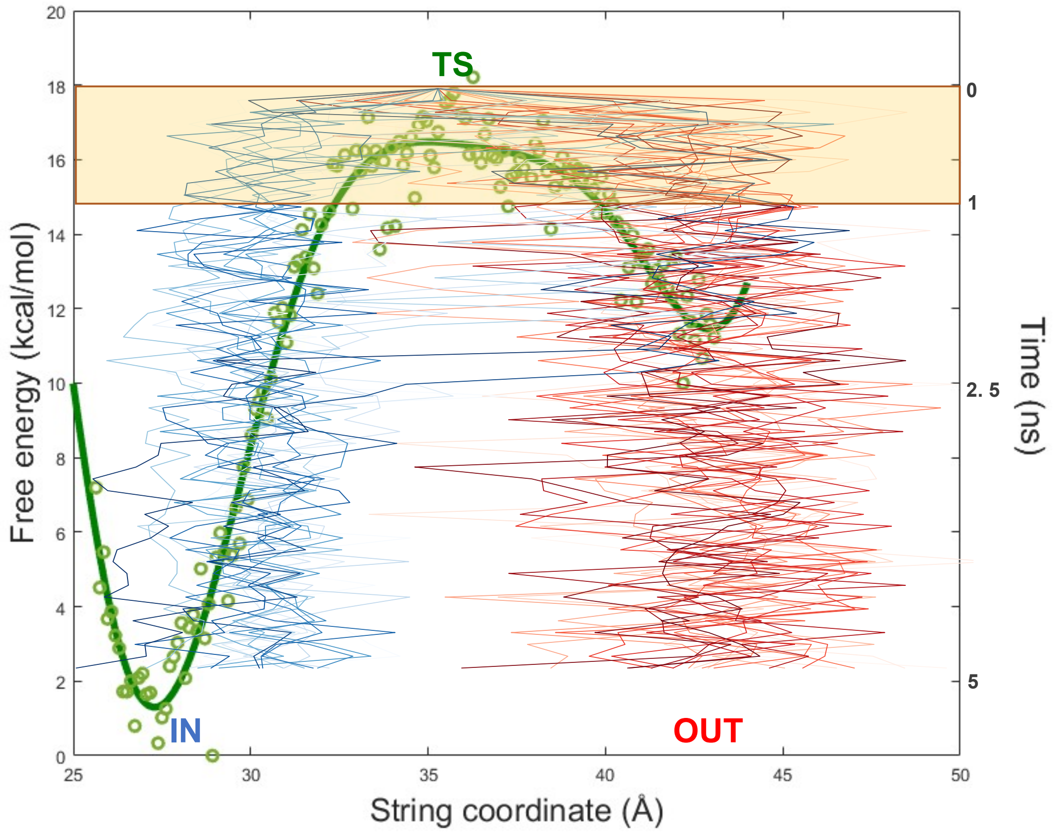


**Figure 4.5:** Representation of the PMF of ligand 62K along the string coordinate and the path of multiple downhill trajectories started at the TS (in green) for further analysis. Note that this string coordinate is the sum of the different distances included in the biased CV. From the TS coordinate as a starting point, a set of simulations leading to both an IN position (blue) and an OUT position (red) are represented as lines. The green dots illustrate the free energy profile datapoints obtained from the WHAM calculation using the string window as string coordinate. The green line represents the fitting obtained from the green dots. The yellow shade represents the simulation time portion used for analysis during our machine learning-based approach.

Only one main barrier is observed, corresponding to the breaking of the drugs with the His84 H-bonding contact (Fig. 4.4) [94], suggesting that the different replicas do indeed share the same TS ensemble, despite the slightly varying pathways and identified CVs along the path. This H-bond was reported as a key interaction in many ligands in complex with CDK2/CDK5 [110, 111]. This interaction was included in the initial unbiased simulation in the bound systems at the beginning of the unbinding procedure. However, during the unbinding trajectories, once this important H-bond between His84 and the ligand is broken, new interactions are formed, for varying time scales. For 18K, in all the three replicas, H-bonds are formed with the exocyclic amino group of the ligand (N5) and the backbone oxygen of Glu81 and subsequently with the backbone oxygen of His84. 62K presents a sulphonamide terminal group, which, during the trajectory, interacts with Val163 and His84 of CDK2.

### 4.3.3 Analyzing and Understanding the Unbinding paths

To analyze which distances are the most important at the TS region, I implemented our MLTSA method. Starting with two datasets of 139 (62K) and 148 (18K) independent downhill trajectories for each system, and initial set of CVs of over 170, I obtained key distances for each system that are major determinants for the prediction of whether a molecule ends up in the bound or unbound states (Fig. 4.5). By training with trajectory data from up to 0.3 ns of each downhill simulation, the model can predict with high accuracy the IN or OUT outcome of the trajectories, more specifically: 80.11% for 18K and 93.83% for 62K. To confirm the effectiveness of the ML training, I compared the ML prediction accuracy with using optimal thresholds of the main string CVs (Fig. 4.5) to determine the outcome at 5 ns of downhill simulations (Figs. B.4, B.5 and B.6). Importantly, the ML model predicts the outcome more accurately at early times (before 0.3 ns), than using the best possible prediction via the string reaction coordinate: with above around 80% to 94% accuracy versus 55-to 61%, respectively for the ML and the main CV (Figs. B.4, B.5 and B.6).

Using the trained model, I then performed a feature reduction analysis to identify which CV features affect the overall prediction ability of the ML model the most. For both molecules I was able to select the most important structural features (Fig. 4.6), that lead to the significant reduction of the prediction accuracy, when

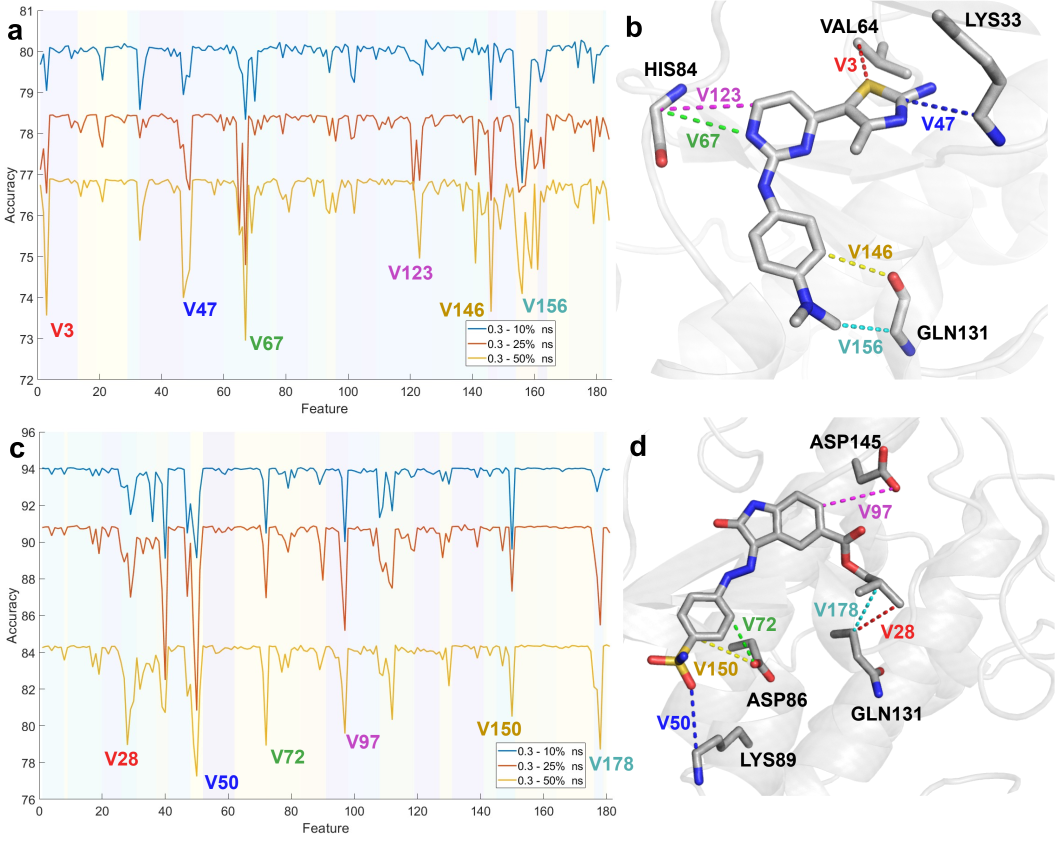


**Figure 4.6:** Identification of the key distances (Feature Reduction) from the largest accuracy drop using the last 50% (yellow), 25% (red), and 10% (blue) of the frames up to the first 0.3 ns of the simulations for a: 18K and c: 62K. The different shades in the background group the different features according to the atom of the ligand involved. Features presenting significant decrease in accuracy are labelled and portrayed as a 3D representation on the right side of each plot: b:18K and d:62K.

such features were eliminated (these were kept as a constant value and fed to the ML, see section 2.2.5 for details), while other features did not affect the overall accuracy of the predictions.

I also compared the validity of the feature reduction approach with GBDT to identify FIs from the GBDT model. The results obtained show broad similarity with the main MLTSA approach (Figs. B.12- B.13) and they both outperform the baseline approach without ML. This suggests that alternative ML models may also be used successfully and further validate the results.

While some of the highlighted interactions with the ligands were already identified for the bound state (such as His84) other interactions were previously suggested (Asp86 and Asp145) and new ones have been identified by the MLTSA (Lys89, Gln131), and suggested as possibly relevant. Note that the highlighted residues in this work are tightly bound to the TS crossing rather than the bound state, and thus, their interactions are meaningful in such context. For example, 62K-Lys89 interaction makes sense when exiting the pocket due to its negatively charged end, whereas Gln131 has more weight in 18K's case than in 62K's due to it's positively charged end. These interactions appear relevant for the ligand's unbinding mechanism, and as such should be studied by other means and further validated as crucial. Doing so would provide more ground for drug designers to improve on potentially improving CDK2's inhibitors with longer residence times, by tuning down these polar interactions.

The MLTSA is significantly less computationally intensive than either the unbinding simulations or the string calculations. The trainings were optimized up to $2min$ per training on 24 CPUs. The short downhill trajectories ( $5ns$) can be trivially parallelized, which constitute the main cost of the MLTSA analysis ($< 4h$ on a standard GPU). The ML training and accuracy drop calculations have a negligible cost compared to these, therefore MLTSA could be a quick and effective approach to understand key CVs at the TS.

## 4.4 Conclusions and Further Work

Optimizing ligand unbinding kinetics is a very challenging problem for small molecule drug discovery and design, that can lead to the development of drugs with superior efficacy. To tackle this, we have developed a new method, which allows us to calculate the free energy barrier for the ligand unbinding process, therefore providing quantitative information about the residence time of a specific ligand. Our method involves an exploration step, where a ligand unbinding path is determined together with key collective variables that describe this path. Subsequently, I performed accurate free energy calculations using the complete set of identified interactions as CVs along the unbinding path via the finite temperature string method. This provides us with the free energy barriers and an ensemble of structures at the transition state of the ligand unbinding process. The novelty of the method lies in the combination of automated iterative addition and removal of the collective variables determining an unbinding trajectory, which allows us to discover novel interactions not available a priori, based on the interactions from the bound structure. I found that while the unbinding trajectories show different paths between different replicas for the same system, our method nevertheless identifies the key interactions important during the unbinding process and provides consistent free energy barriers. The combination of generating an initial path and identifying the important CVs for the unbinding process with the string method for accurate free energy calculations using high dimensional reaction coordinates provide an efficient way to obtain quantitative kinetics of ligand unbinding.

I tested this method using a well-studied cancer drug target, CDK2, using two drug molecules with measured kinetic profiles. I obtained energy barriers in agreement with experiments using our method, which demonstrates the fundamental importance of determining a well-selected, high-dimensional set of CVs for the correct description of the process and kinetics results. Although I previously showed studies in which a bad selection of CVs leads to poor results, in this case, no results for a poor CV on CDK2 unbining have been done or published previously, however kinase ligand unbinding is challenging on its own.

To aid the kinetics-based design of novel compounds, we also developed a novel method, MLTSA, that allows us to identify the most important features involved at the TS of the unbinding. I generated multiple trajectories initiated at the TS, which either terminated in the bound state or in the unbound state. I then trained an MLP to predict the outcome of the trajectories by using a set of CVs and data drawn from the initial segment of the trajectories only. By doing so, I were able to demonstrate that the ML was able to predict the trajectory outcomes with much higher accuracy than using the original set of CVs used for the free energy calculations. A feature importance analysis was further employed to then identify the key CVs and the corresponding structural features that determined the fate of the trajectories, which therefore are the most important descriptors of the TS.

The average training time using a single core was around 3.5 minutes/model to converge, whereas the GBDT training took about 5 minutes/model. Thus, it is suggested that a joint approach with both models which may complement one another, could be used to identify relevant CVs. Nonetheless, future studies with non-linear correlated time series can further help to explore the performances of these and other ML methods. Importantly, analogous analysis can be performed for various complex processes, including ones with multiple states as possible outcomes.

In addition to binding rates, I also aimed to identify specific molecular features and interactions with the target protein that allows us to design kinetic properties of the ligand. Using our ML methods, I identified multiple interactions between the protein and specific parts of the ligands that were of major importance for the trajectories to cross the TS. Important protein-ligand interactions at the TS-bound poses for CDK2 correspond to functional groups of the distal ends of the ligands. Besides His84, a known key residue for interaction with multiple CDK2/4 inhibitors, here I also identified additional common interactions within CDK2 across the ligands, for example between Lys89 and the sulfonamide groups or between Asp145 and the carboxylic group and the ester group for 62K, respectively. Importantly, to perform this analysis, one requires the approximate knowledge of the TS structures as well as the MLTSA approach generating a set of downhill unbiased trajectories from these starting points. Our algorithms enable us to uncover novel design objectives for a kinetics-based lead optimization process.

## Chapter 5

# Understanding GPCRs: Unbinding Muscarinic Antagonists from the hMR3 Receptor

This chapter is the work from our latest paper [2], where we applied the unbinding protocol and the MLTSA approach to a much more complex system, a transmembrane protein with long-action inhibitors, the human Muscarinic receptor 3 (hMR3). This work is much more data-driven and results focused than the previous. The methodology was applied very similarly to the previous chapter, and in this one we were more confident using the MLTSA to focus the research. It is an unpublished paper, but it is available as a preprint in biorXiv.

## 5.1 Introduction

Muscarinic receptors (MR) are a five-membered subtype group of transmembrane receptors, which form an important part of the parasympathetic nervous system. They are activated by neurotransmitters such as acetylcholine and muscarine [112], and transmit extracellular signals to the cell interior, which makes them attractive drug targets [113].

The sequence identity between the five MR isoforms is low, except between the transmembrane regions [114, 115]. This region contains seven alpha helix substructures, which anchor the protein in the outer membrane of the cell [116]. On the cytoplasmic side, the receptor is bound to a GTP-binding protein, which is responsible for the subsequent signal transduction. Therefore, MRs are part of the G-protein coupled receptor (GPCR) superfamily.

Downstream signaling can be spontaneously induced when MRs bind to GTP-binding protein, even in the absence of the corresponding agonist [117]. Activation, as well as the downstream signaling can be suppressed when suitable antagonists are bound to MRs. This can be exploited pharmacologically, and several important muscarinic antagonists were developed and used for instance, as bronchodilators in the treatment of asthma or chronic obstructive pulmonary disease (COPD) [118–121].

Human MRs (hMRs) are expressed in a variety of tissue in the human body, therefore a drug with low selectivity may cause severe complications and side effects [122]. While the hMR3 isoform — which controls the tension of the smooth muscle tissue in the bronchial tubes — is the actual target of bronchodilators, the off-target binding to the highly homologous transmembrane region of hMR2 is responsible for serious side effects, especially in the cardiovascular system [122–125]. Due to the high homology between the two isoforms, the binding affinity of most muscarinic antagonists is very similar. For example, the pKi value of the pharmacologically widely used tiotropium for hMR2 is 10.7 and for hMR3 11.0 [122]. Nevertheless, tiotropium shows a high selectivity because the dissociation rate from hMR2 is significantly higher compared to that from hMR3 by about one order of magnitude [122, 126, 127]. As a consequence, the residence time of tiotropium in the hMR3 isoform is very long and the binding was considered to be kinetically irreversible [122, 128].

As in general, the drug unbinding process is a rare event, it is highly challenging to study it experimentally and the detailed mechanism is still mostly unknown. However, there are several computational studies available that attempt to approach this problem via molecular dynamics (MD) simulations [3, 129].

Simulations on the beta-2 adrenergic receptor using RAMD found two different types of pathways for the unbinding of the beta blocker carazolol. One of them along the long axis directly into the extracellular space and one laterally into the membrane [130]. Recently, it was shown that the path leading directly into the membrane is probably an artefact caused by the force constants of the biasing potentials being too high [131]. For the same receptor, binding paths for several antagonists and agonists could be identified by conventional MD [132]. A free energy profile (FEP) was also presented, which is characterized by two barriers. The first barrier describes the process of docking of the ligand from the solution to the tunnel entrance of the receptor (the extracellular vestibule). The second barrier is on the way of the ligand from the extracellular vestibule to the orthosteric binding site.

Later works using metadynamics and Markov State Models (MSMs) found the resting state in the extracellular vestibule to be very shallow and a significant barrier for the desolvation process could not be found [63]. It is now largely consensus in the available literature that the rate determining step is indeed on the way from the vestibule to the binding site [133, 134].

Previous studies on the unbinding path of the hMR2 receptor and its agonist iperoxo have also shown that the process encompasses two steps. In these unbinding processes the rate limiting step was found to correspond to the ligand exiting from the orthosteric binding site to extracellular vestibule [135, 136]. Two different exiting pathways are suggested, the first one (and more favorable) involves the rotation of the ligand and its exit through the extracellular vestibule, while the second one is characterized by the rearrangement of the extracellular loop 2 (ECL2) limiting the ligand from fully entering the solvated state, i.e. leaving the protein completely. Free energy profiles for the unbinding were estimated using metadynamics, however, calculations of the free energy barrier or unbinding rates proved to be challenging due to force field inaccuracies [136]. Given the homology between hMR2 and hMR3 similar limitations are expected to arise, which have been considered for this study.

In this work, I applied our recently developed unbinding algorithm [1] to hMR3, to investigate the dissociation of tiotropium (**1**) and two structurally similar ligands, N-methylscopolamin (**2**) and homatropine methylbromide (**3**) (Fig. 5.1).

The obtained unbinding pathways were refined using an adaptation of the finite temperature string method [137] (Please see 2.2.1 for details). Finally, the transition state (TS) of the tiotropium unbinding was detailed and analyzed with the aid of machine learning (ML) to identify prominent interaction pairs of the ligand and the receptor at different levels. Additionally, I also revealed key conformational changes of the protein that define the downhill trajectory outcomes.

## 5.2 Computational Details

### 5.2.1 Building hMR3

**Starting Structure.** The starting coordinates for hMR3 were obtained using a rat MR3 crystallographic structure, PDB ID 4U14 [138], with a resolution of 3.57 Å and with tiotropium bound in the orthosteric site. The structural model was truncated to the transmembrane helices and the extracellular loops, which are highly conserved between human and rat (91.85% homology) and contain the necessary and sufficient domains for ligand unbinding [114].

**Parameterization and System Building.** The protein was inserted into a membrane using the membrane builder [139–141] of the CHARMM-GUI web server [142–144]. and then solvated in water [145] with 150 mM KCl. The membrane consists of $POPC : DMPC : PYPE : DMPE$ in the ratio of $1 : 2 : 3 : 4$, chosen on the basis of earlier studies of hMR3 and on tracheal membrane tissue [146].

The ligands (Fig.5.1) were geometry optimized at the B3LYP/6 31G** level of theory [147] applying the ORCA 4.1 software suite [148–150]. With the optimized structures, force field parameters for the ligand were defined using the CHARMM-GUI ligand reader [151].

The all-atom CHARMM36m force field was used for the protein [152–154] and the lipids [155, 156], and the TIP3P model [145] for the water. Simulations were carried out with the NAMD software package [103] using input generated by the CHARMM-Input generator [157]. The cutoff for non-bonded interaction was kept at 12 Å, the switch distance at 10 Å. Electrostatic interactions were handled by a particle-mesh Ewald solver with a grid spacing of 1 Å. The temperature was

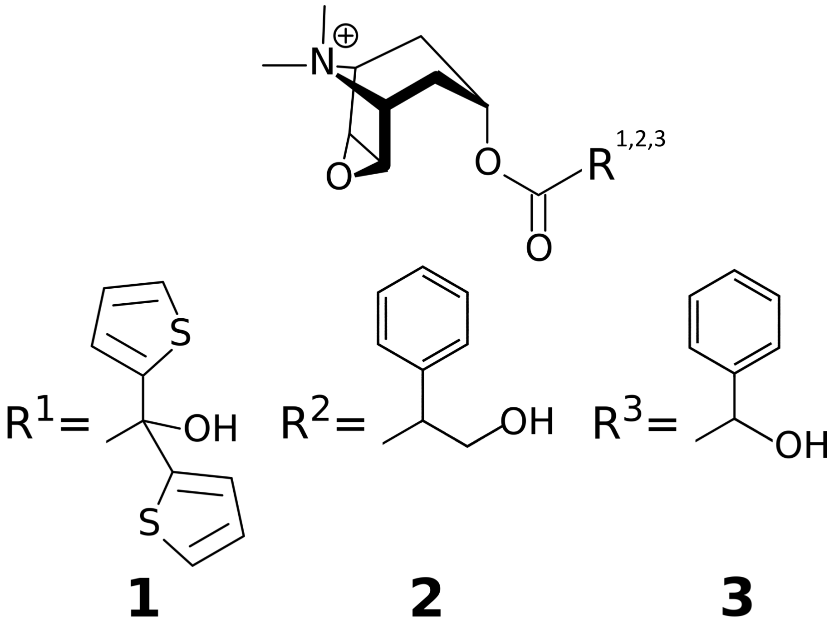


**Figure 5.1:** 2D molecular representation of the pattern structure (top) shared between all three ligands (bottom) and their distinct radicals (*R*). Numbers correspond to tiotropium (**1**), N-methylscopolamin (**2**), and homatropine methylbromide (**3**).

kept at 310.15 K using Langevin dynamics. Pressure was kept at 1.013 bar by Nosé-Hoover Langevin piston pressure control [158, 159]. The structures were first energy minimized according to the CHARMM-GUI scheme and subsequently equilibrated for 50 ns.

### 5.2.2 MD Simulations of hMR3 Inhibitors

**Unbinding Simulations.** The unbinding procedure was followed as described in section 2.2.1. After the equilibration, a 20 ns production run without any restraints was performed. During this production run all interacting pairs of heavy atoms – one in the ligand and one in the protein – were identified. Thereby, a pair is defined as “interacting” if the distance between the atoms is below 3.5 Å for more than 50% of the simulation time. Based on the sum of these interacting distances, a collective variable (CV) is defined and restrained harmonically [1] (See section 2.2.1). During an iterative process, subsequent simulations of 10 ns use this biasing CV with a force constant of 10 kcal $mol^{-1}\AA^{-2}$. The constraint position (i.e., the length) of the CV is monotonically increased. In the next iteration, new interaction sites are identified in the same way as before and these are added to the CV. Interactions are discarded and removed from the CV, if the distance between the atoms is larger than 11 Å. A shorter cutoff distance results in the ligand falling back into the original binding position after a few iterations. This is possibly due to the long range interactions still acting on the ligand and prompting its return back to the bound state. The procedure is repeated until the ligand is displaced out of the receptor.

The unbinding simulations were run for 25 iterations, adding up to a total of 240 ns simulation length. Thereby, a total of 52, 50, and 44 interacting protein-ligand distances were identified by our unbinding method along the paths for ligands **1**, **2**, and **3**, respectively.

**Refinement of the path using the string method.** The unbinding path was used as a starting point for the subsequent refinement using the finite temperature string method [48] as described in section 2.2.1. Since the string iterations are computationally expensive and the convergence is slow due to the high-dimensional nature of the system, only 20 iterations were calculated.

**Approximation of the TS region.** To approximate a TS structure from the string windows, I identified a set of structures from the string windows, which are very similar in the unbinding paths of all three investigated ligands ( comparison in Fig. 5.3). I selected five string windows as starting points around the window with these distinct structures for ligand **1** and performed 50 independent unbiased (downhill) MD simulations with 5 ns lengths each. Thereby, I was able to identify the structure that provided the closest 1:1 ratio of a binding (IN) or unbinding (OUT) events, which I considered to be the TS of the unbinding process.

### 5.2.3 MLTSA Protocol

To aid in the identification of the main CVs driving the system across the TS and to pinpoint novel descriptors that determine the fate of a binding/unbinding events, I used our MLTSA analysis (section 2.2.5). In this approach, a ML model is trained to predict the outcome of downhill simulations with data close to the TS. Subsequently, the trained models are used to discover the key TS-defining features of the system.

**Creating the datasets.** Using ligand **1**'s identified TS structure as the starting point, multiple 5 ns long unbiased simulations were run. 149 downhill trajectories were classified and labelled by considering a linear combination of 52 distances to identify which simulations arrived to an IN or an OUT state. A minority of additional trajectories not reaching clearly either the IN or the OUT states after 5 ns were discarded. To train ML models, several sets (see Fig. 5.2) of features containing different distances (CVs) along the simulation frames were created, following the methodologies described in section 2.2.2:

- To assess **intra-protein interactions**, a first dataset (*XYZ-PCA set*) included the Cartesian coordinates of all protein atoms ( 6*K*, not including hydrogens). To reduce the dimensionality, PCA analysis was applied as described in section 2.2.2 and only the top 100 components were used as input features. Please see section 2.2.2 and C.2.1 for further details on the methodology applied. However this dataset is complex to interpret.

- To enable more **interpretable localized features**, additional datasets containing ligand-protein distances were created. The first such set (*3Å set*) contained all interatomic distances between the ligand and the protein within 3 Å of the ligand at the starting TS position, excluding hydrogens. The second dataset of this type (*6Å set*) was created in a similar fashion to the previous one, but with a cutoff of 6 Å instead. These datasets followed the protein-ligand shell methodology described in section 2.2.2. These datasets enabled for a better understading of the relevance of features near the active site of the protein.

- For the fourth dataset (*3Å+ECL2/TM5 set*), the same data within 3 Å of the ligand was used, with the addition of the interatomic ligand-protein distances of the extracellular loop 2 (ECL2) and the transmembrane region 5 (TM5), including residues from I222 to T231. These datasets were created to assess the importance of the different loops, when compared to the active site residues themselves.

- An additional fifth dataset, to assess **overall ligand-protein contributions**, was also created (*allres set*), which considers all residues and includes the closest distance between the residue and the ligand at each simulation frame (see section 2.2.2 for clarification). This allowed for a more broad view of what parts of the protein are relevant and it was easier to implement than single-atom distances, which would have been computationally expensive to screen. This dataset was also later amended with the closest 8 water molecules, their distances to the ligand (*allres+wat set*) was included to enable the assessment of the role of water molecules.

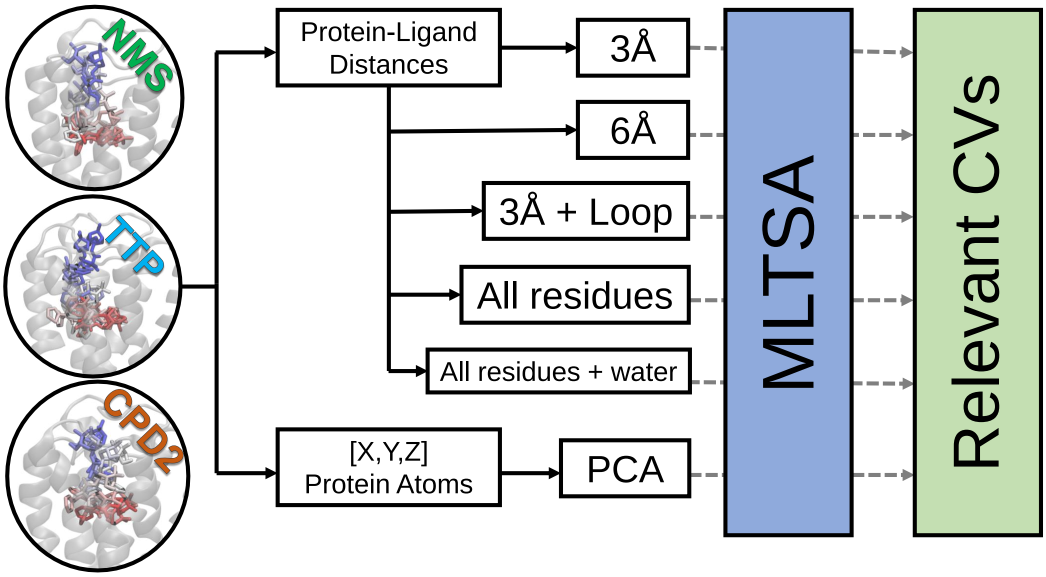


**Figure 5.2:** Diagram showing the unbindings of ligands **1** (cyan, TTP) **2** (green, NMS) and **3** (orange, CPD2) of the different datasets derived from **1**'s downhill trajectories.

A visual summary of all datasets created can be found in Fig. 5.2.

**Machine Learning models and training.** I used two different ML models: a Multi-Layer Perceptron (MLP) neural network classifier [160], and a Gradient Boosting Decision Tree (GBDT) classifier [161]. Both models were trained to predict the outcome (IN/OUT) of the simulations from early on data at the time range from 0.05 ns to 0.1 ns, totaling 2500 frames per simulation. I trained 100 independent

MLP and GBDT models randomly assigning the 149 simulations into training data (70%) and validation data (30%). Details on the trainings and hyperparameters can be found at section C.3.

**Feature Analysis.** Following the methodology in section 2.2.4, I used the Gini feature importance [162] to evaluate the relevance of the features from the GBDT models, averaged across the 100 trainings to calculate their relative feature importance (RFI). To identify key features in MLP models, I removed the variance from each feature one-by-one [1] (with the algorithm in section 1) and assessed the accuracy drop when predicting outcomes with the altered dataset on the trained models. If the accuracy of the prediction is greatly reduced when a feature is altered, the feature is considered important for the description of the TS. I identified the overall top features averaging the relative accuracy drop (RAD) from all 100 trainings on all datasets used (Figs. 5.6 to 5.7).

## 5.3 Results and Discussion

### 5.3.1 Unbinding the Bronchodilators

**Bound state in the orthosteric site.** In all three ligands, the initial ligand positions in the unbinding simulations are close to the starting bound pose: the charged end of the molecule is nestled in an aromatic cavity, which is formed by the residues W503, Y148, Y506, and Y529. The tyrosines form a cap around the ligand. Simultaneously, the S151 residue coordinates the epoxide group via a hydrogen bond and the negatively charged residue D147 neutralizes the positive charge of the ligand. At the opposite end of the molecule, the N507 residue stabilizes the molecule by a hydrogen bond with the OH group. The same binding mode was also described in recent works [128, 133, 138, 163].

**Departing from the binding site.** As illustrated in Fig. 5.3 and 5.4, the first movement from the binding state (Fig. 5.4.**A**) is a rotation of the charged end of the molecule. Thereby, the hydrogen bond of the epoxide group with S151 is broken and the ligand slightly gains flexibility. Apart from that, the ligand's position in the binding site remains nearly unchanged (Fig. 5.4.**B**). This first movement is most

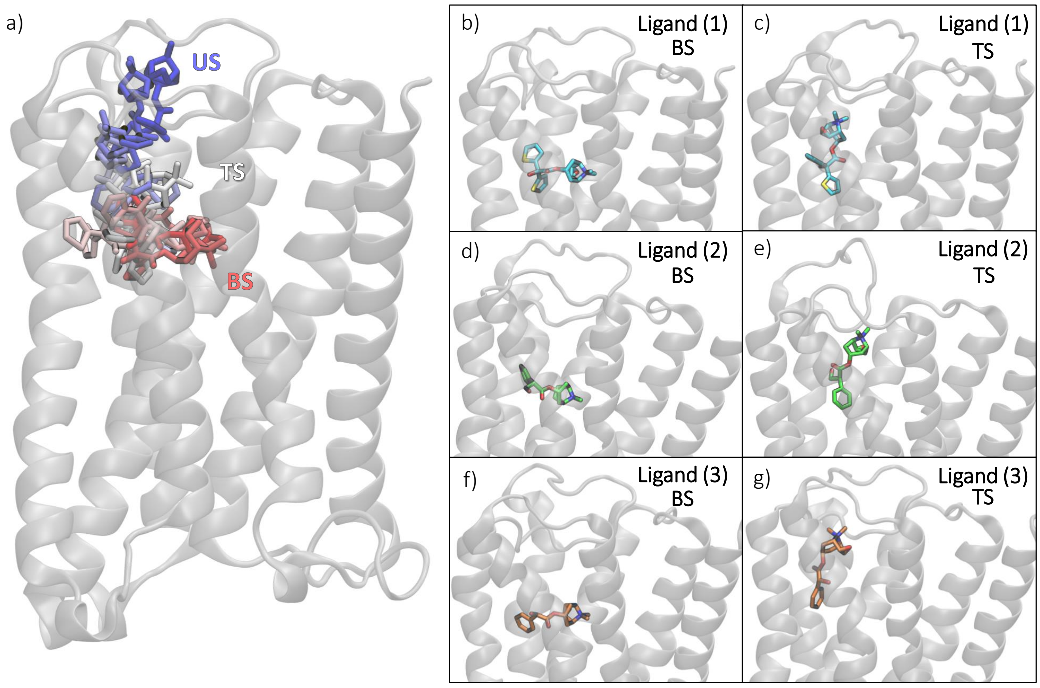


**Figure 5.3:** **a**: Overlay of ligand **1**'s (tiotropium) structures during the unbinding path from bound state (BS) to unbound state (US) passing through a transition state bottleneck (TS) similar in all three ligands. BS and TS comparisons for ligands **1** (b and c), ligand **2** (d and e) and ligand **3** (f and g)

pronounced for ligand **1**, which follows a helical motion along its longitudinal axis and thus it detaches itself from the aromatic cavity.

This shift is present, but less pronounced for ligand **2** (Fig.5.4.**A**). Subsequently, ligand **2** breaks through the tyrosine-formed ceiling via a path associated with significantly more dislocation of the residues Y148, Y506, and Y529.

In the path of ligand **3**, the entire molecule does not shift, instead mainly the end of the ligand with the thiophene ring moves (Fig. C.1). This allows the charged end to slip outwards the aromatic cage in a rolling motion.

**Through the bottleneck.** The new position after the shift, allows the molecules to rotate their charged end by 90° towards the direction of the receptor tunnel's exit (counterclockwise), without exerting a lot of tension on the tyrosine residues forming the aromatic cap. During this rotation, all three paths pass through a state (Fig. 5.4.**D**), which is highly similar in all unbinding trajectories. Interestingly, this

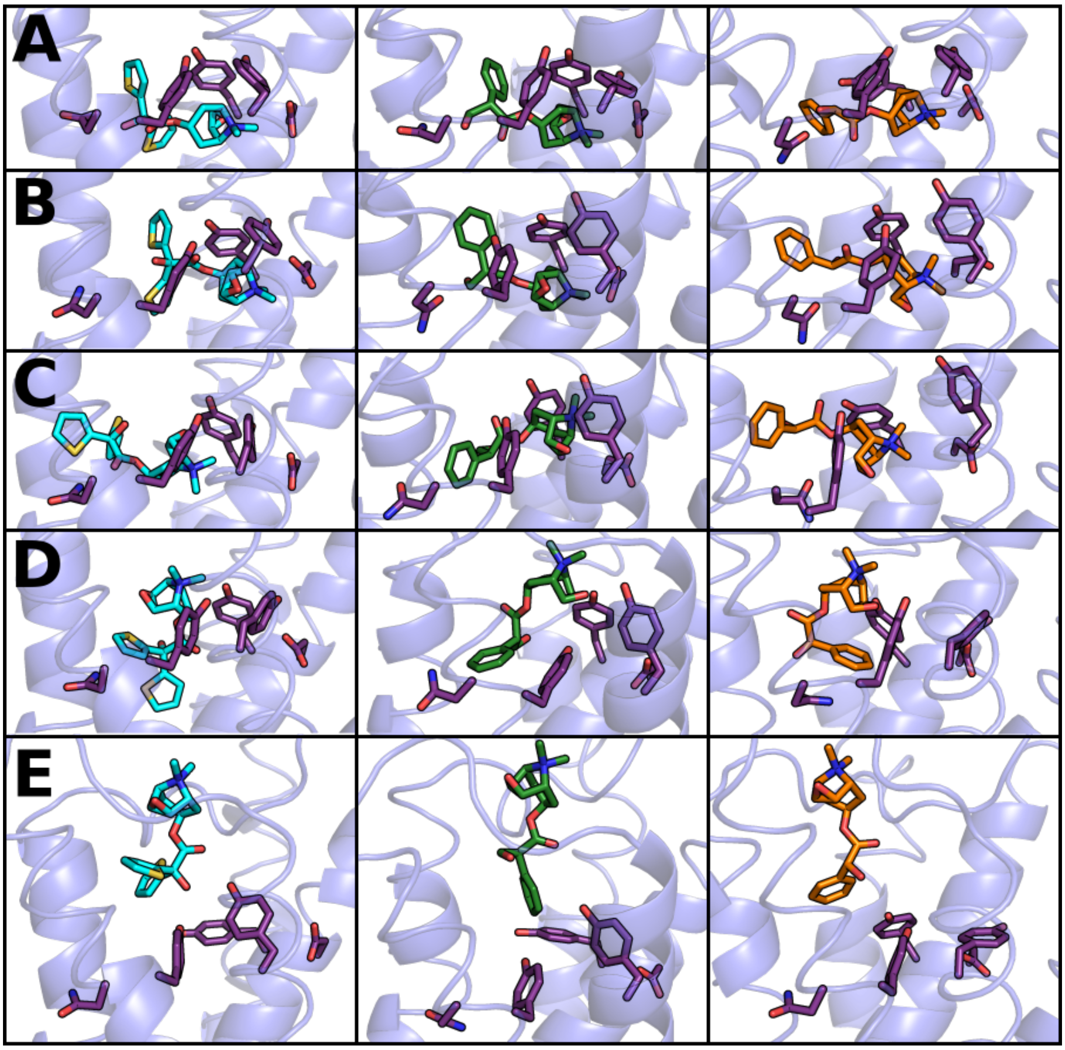


**Figure 5.4:** Key frames of the unbinding path of ligands **1** (left, cyan), **2** (middle, green) and **3** (right, orange) in sticks. hMR3 is represented as cartoon, residues forming the aromatic cage of the orthosteric binding site are also represented as sticks (in purple). Frames start from being bound in **A**, to different points in the unbinding process for each ligand, finishing with the ligand on its way to the extracellular vestibule in frame **E**, this point is considered to be the TS of the whole process.

rotation was observed to proceed clockwise for the iperoxo ligand unbinding path in hMR2 [136]. This movement positioned the charged end of the molecule pointing towards the membrane in these previous simulations, and not to the extracellular vestibule observed by us. I subsequently found by unbiased simulations starting from this structure that either the ligand is led back into the binding site, or it moves along the exit tunnel towards the extracellular vestibule (Fig. 5.4.**E**). Therefore, this position can be identified as a TS of the unbinding from the orthosteric site. The pathway is also similar to the previously reported forced dissociation with acetylcholine as well as tiotropium (**1**) on hMR3, and with a slightly tilted orientation on hMR2 [133].

**To the extracellular vestibule.** For all ligands the total length of the simulations was not sufficiently long to observe the complete unbinding of the ligand, rather the ligand remains in the extracellular vestibule but outside the orthosteric binding site. However, in line with the consensus literature, it is estimated that the final unbinding step from the extracellular vestibule has a significantly lower barrier, therefore it does not likely contribute to the off rate [63, 133].

### 5.3.2 Understanding Triotropium's Unbinding

**Downhill trajectories from TS structures.** I evaluated starting structures from 5 string windows for ligand **1** near the bottleneck conformations that all three shared (Fig. 5.4). The structure closest to the TS position led to 85 and 64 downhill trajectories of 5 ns reaching the IN and OUT states, respectively (Fig. 5.5). To explore the time range where the TS is probed, I performed initial ML trainings to identify the region where the ML method can accurately, but not with full confidence, predict the final outcomes from as early an timeframe as possible. I found that this was already possible from the 0.05 to 0.1 ns timeframe. Trainings at different times can be found in Fig. C.3, final accuracies for all datasets in Table C.1.

**Assessing contributions from protein conformational changes.** To consider changes in the protein structure affecting the unbinding, I analyzed the protein Cartesian coordinates via their top 100 PCA components (Fig. C.2). I was able to predict the outcome very accurately, obtaining average test accuracies of 100% (MLP) and 93% (GBDT). Out of the 100 components, the first two PCA components were important both for RFI and for RAD. Additionally, PCA23 and PCA59 were important for RFI (see section C). The main PCA component represents large-scale movements from the TM2, TM3, and TM6 to TM7 helices, including some ECL1 residues (Fig. C.4 and C.5). The residues that contributed the most are from the middle of TM6, close to the ligand. The second main PCA component (top RAD feature) represents motions from the rest of the protein, mostly from TM4 to

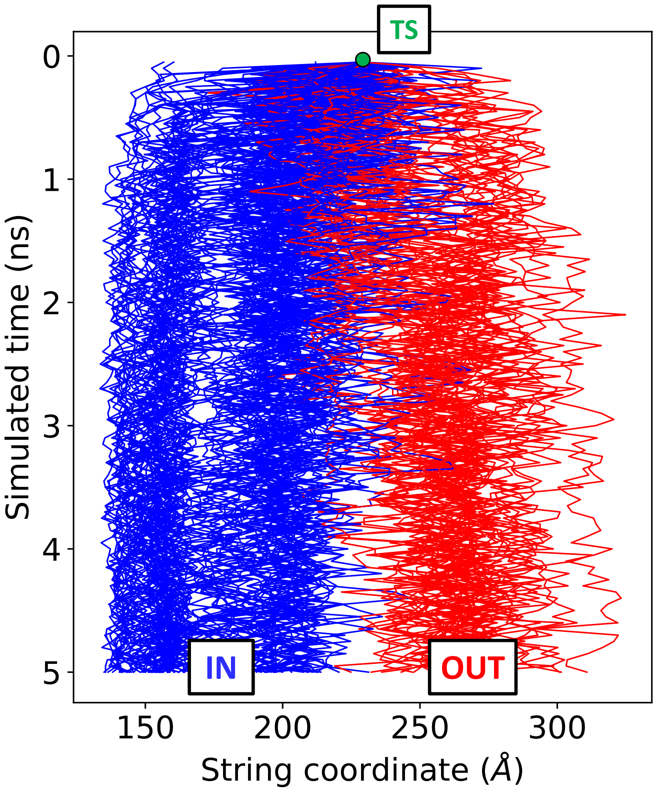


**Figure 5.5:** String coordinate value evolution during the downhill simulations started from triotropium's TS (value in green). Simulations ending in IN are blue and red for the ones ending in OUT.

TM5, with the ECL2 loop being especially relevant. The largest contributions come from residues (W206, Q207, I222 and Q223) that belong to ECL2/TM5 junction, some from TM4 that are close to the ligand (I194 and V193). However, due to the broad distribution present in the PCA components, their interpretability is limited. Therefore, I focused next on feature sets that are precisely localized and able to assess specific ligand–protein atomic distances instead.

**Key feature identification from the 3Å dataset.** I created a high-resolution dataset, which contained atomic distances between the ligand and protein residues within 3 Å of the TS structure of the ligand. Using the 3Å dataset, I achieved a prediction accuracy of 78% with MLP and 77% with GBDT, and obtained consistently similar key features by RAD and RFI (Fig. 5.6). Both models (MLP and GBDT) agreed on the importance of four out of six top residues: D147, W199, T231 and Y529.

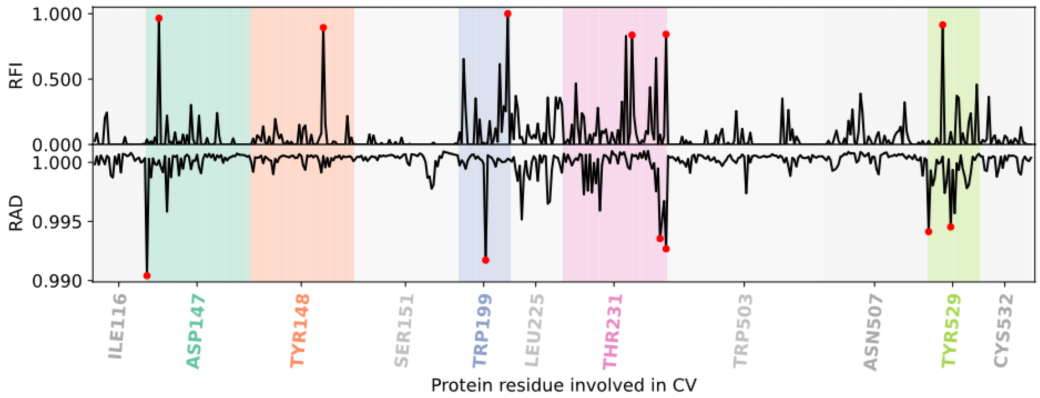


**Figure 5.6:** Relative feature importance (RFI, top) and relative accuracy drop (RAD, bottom) shown for every interatomic distance between ligand 1 and hMR3 in the 3Å dataset. Distances are ordered and clustered by residue number. Residues with the top six distances (red symbols) are highlighted.

Three of these key residues were previously known to play important roles in the unbinding process. D147, as mentioned earlier, interacts with the charged amine moiety in the bound form. Similarly, Y529 is part of the aromatic cage around the ligand. Additionally, the aromatic substructures of the ligand are known to interact with a hydrophobic region close to W199. Mutational studies show an accelerated dissociation (lower $K_i$ beyond the expected) for Y529A and reduction of the half-life for both W199A and D147A, further suggesting their involvement [128].

Interestingly, T231 was not previously reported and validated as relevant for ligand interactions in the bound state. Even though there are no experimental studies, it was previously identified computationally to form relevant contacts during the forced dissociation of tiotropium [133].

**Contribution of the extracellular vestibule within 6 Å.** To assess the contributions from more distant atoms beyond 3 Å, I also analyzed results from a dataset that includes 5000 interatomic distances within a range of 6 Å from the ligand at the TS. In this dataset, I analyzed both individual feature importances, and average importance values for each residue (Fig. 5.7). Accordingly, D147 and T231 are again part of the top 6 key residues both measured by RAD or RFI. Newly identified key distances include I222, and T234, which were not part of the previous dataset, as well as additional heavy atom distances from L225 and N507. L225 was previously reported relevant for the binding/unbinding kinetics in hMR2/hMR3 experimental studies, but insufficient alone to explain the difference between both receptors. N507 is a previously validated relevant interaction that accelerated the dissociation of tiotropium when mutated to Ala (N507A) [128, 133].

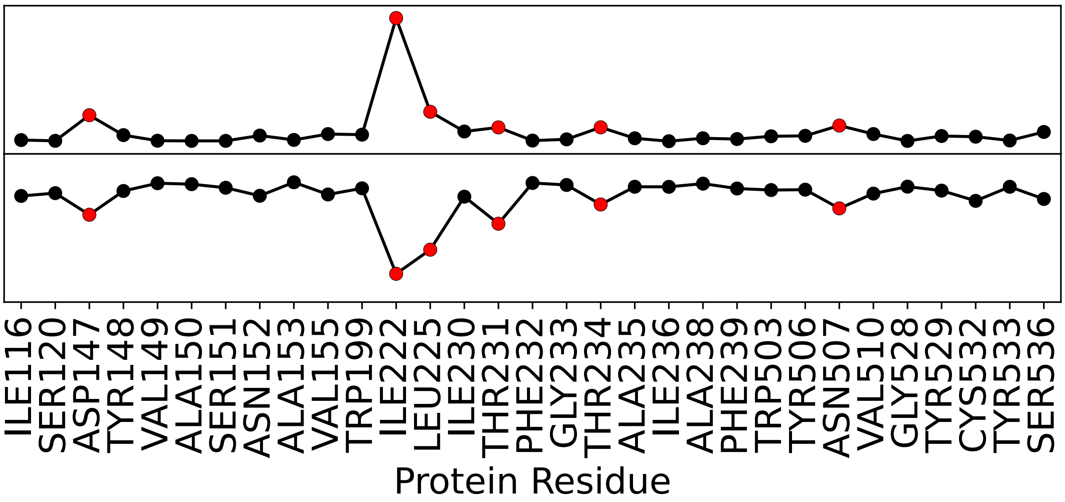


**Figure 5.7:** Average RAD (from MLP) and RFI (from GBDT) of the interatomic distances of the ligand 1 per protein residue for the 6Å dataset. In red, the top 6 residues detected by both approaches. Note the Y scale goes from 0 to 1.

Interestingly, the residue with the most relevant interactions is I222 and it was not described previously. Together with L225 and T231, I222 forms a hydrophobic cluster on the extracellular vestibule (Fig. 5.9 and 5.10a, b and i). The fact that the most relevant residues (I222, L225, T231, T234) are close together in an extra-cellular loop (ECL2) may be indicative of the importance of this loop for the unbinding. When aligning hMR2 and hMR3 protein sequences, most of the sequence is identical, but the region prior to T231 (ECL2/TM5) has a high genetic variability (Fig. 5.11). Interestingly, preceding I222, there is another variation in the sequence for ECL2: F221 in hMR3 is substituted with Y177 in hMR2. Moreover, this residue is a potential phosphorylation/modulation site for hMR2 [164, 165], thus thought to be not only an important region for allosteric regulation, but it could alter the observed unbinding kinetics depending on the phosphorylation state of hMR2. This suggests that the residues between I222 and T231 may be relevant to the significantly different behavior observed between hMR2 and hMR3 in terms of residence times [122]. Hence a third dataset (3Å+ECL2/TM5 Loop) was created containing all the residues prior to T231, which range from I222 to T231.

**Exploring the role of the ECL2/TM5 junction.** In the presence of distances from this region (Fig. 5.11), the top features belong mostly to the ECL2/TM5 junction, except for W199 when using RAD. In hMR2, L225 corresponds to a Phe residue (Fig. 5.11), which is bulkier. Interestingly, this change was previously reported to remove a pocket in hMR2, which is present in hMR3 [133]. The negatively charged E227 is replaced by a neutral Asn in hMR2. Remarkably, both ML models found E227 important, despite its longer distance (Fig. 5.10.g). This residue has been mutated to Ala (E227A) previously, resulting in a slight decrease in the half-life of tiotropium, **1**, from 24.5 h to 20.1 h. The RFI, however, found an additional key distance involving F224 as one of the most relevant distances. When mutated to F224A, the half-life of **1** is reduced by  50% to 13.8h [128].

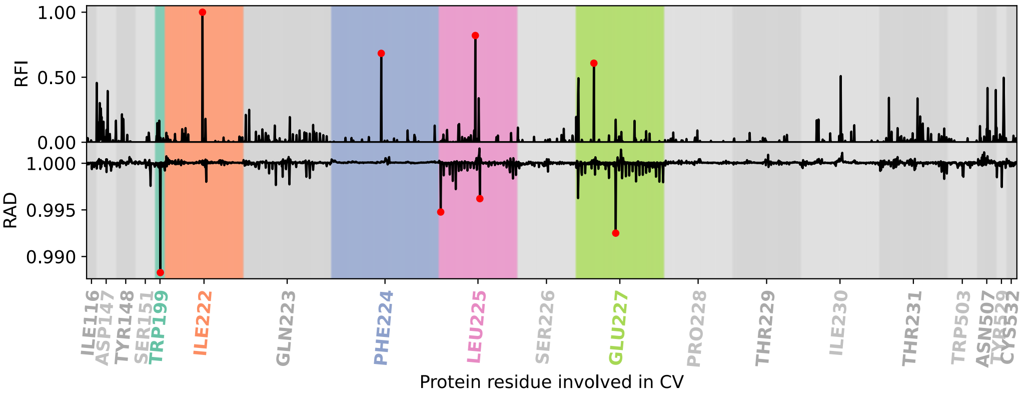


**Figure 5.8:** Relative feature importance (RFI) (from GBDT model) and relative accuracy drop (RAD) (from MLP model) values for each interatomic ligand-protein distance per residue in the ligand **1**'s 3Å+ECL2/TM5 dataset. Marked in red are the top distances for each model. Highlighted, the most important residues for the ML models.

Additional tests with distances from an alternative loop, ECL3, were also added to the 3Å dataset and analyzed (Fig. C.6) for comparison. These demonstrate no significant contributions from this region, thus validating the unique role of the ECL2/TM5 junction.

**Structural spotlights of tiotropium involved in unbinding.** The results point to key atomic contributions from only a few selected atoms of tiotropium (Fig. 5.9.c).

The most prominent moiety corresponds to the methyl groups (C and C8 atoms) that are bonded to the charged amine. Of key relevance is also the S1 sulfur atom from only one of the two thiophene rings, showing key interactions with W199, I222 and T231 (Fig. 5.9 panels c, b and i, respectively). Finally, the O2 atom from the carbonyl oxygen of the ester group is also important, as identified in interactions with W199, L225 and Y529 (Fig. 5.9 panels c, a and f, respectively). In agreement with the results, previous studies have shown that the tiotropium analogues with the closest $K_i$ values have a pattern containing all three groups: an amine cap, the carbonyl group in between, and two aromatic rings (thiophene or not) at the end [128].

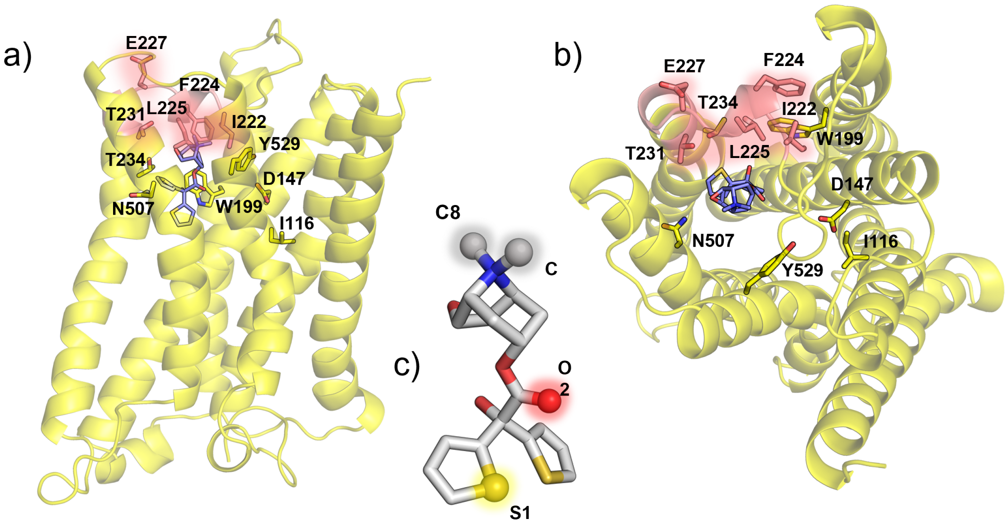


**Figure 5.9:** Front (a) and top (b) view of the M3 receptor at the TS, in sticks the most relevant residues for the unbinding process found by the ML models. In salmon, the residues belonging to the ECL2 loop, which is found to be the most relevant region. c) Ligand 1's structural representation with the most relevant atoms found by the MLTSA, highlighted, and annotated.

**Overall residue-ligand contributions.** To assess all the residues in the protein, I decreased the resolution of the feature space, and evaluated only features defined via the closest distances between each residue and the ligand (*allres dataset*). This allows us to evaluate all residues, including the ones far from the ligand, which can nevertheless have key impact on the simulation outcome. The resulting training from this dataset yielded 79% for GBDT and 77% for MLP on their test set. T234, highlighted in the previous results as a key residue in the 6Å set as well, is the most

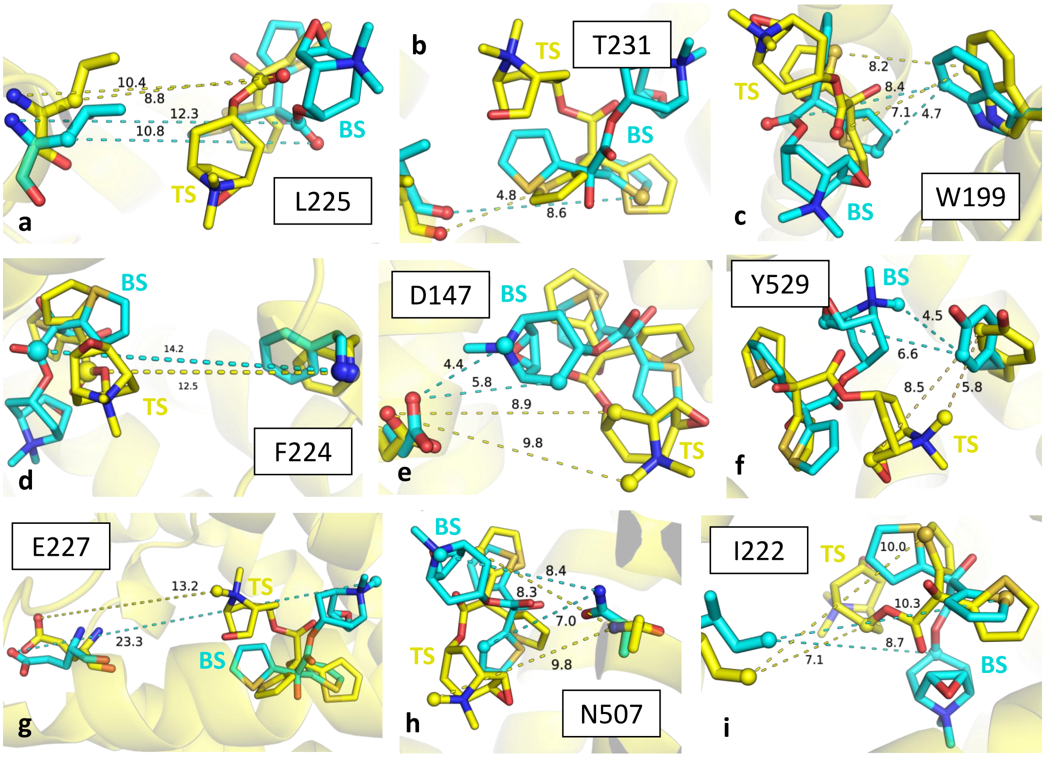


**Figure 5.10:** a) to i) are the top nine residues represented as sticks with their protein-ligand (hMR3-ligand 1) distances consistently found to be most important throughout the MLTSA analysis across all datasets. In yellow, the ligand-protein complex at the TS and their distances, in cyan the ones corresponding to the complex at the BS. Represented as spheres, the atoms that the interatomic distances represented correspond to.

important feature for RAD, and second most important for RFI, validating its key role (Fig. 5.12.a).

A more distant residue that shows key importance is L482, ranked 1st for RFI and 5th for RAD. This distant residue is at the N-terminal end of the TM6, located very near the kinase domain of ICL3, at the interface of membrane and the intracellular matrix (Fig. 5.12.c-d). This could signal changes in the ligand-bound state to the ICL3, which is not modelled in my simulation system. Accordingly, this region is located between two main binding regions of hMR3 for activation and regulation [166, 167]. Pyrophosphatase-2 (PPase 2A), a transmembrane enzyme which targets the C terminal region of the ICL3, the "KRKR" motif in ("ITKRKRMSLIKEKKAAQ"), is thought to be involved in hMR3 dephosphorylation [168]. Additionally, the muscarinic receptor signaling regulator, SET, a PPase 2A inhibitor, also binds to the same motif [169]. Furthermore, it was also suggested that protein kinase G II (PKG-II) activates hMR3 via a cGMP-dependent phosphorylation at S481 ("MSLIKEKK" motif) [166, 170]. Therefore this region is thought to be a putative phosphorylation site just preceding L482 [166]. Interestingly, ligand-dependent phosphorylation of S481 was also connected to enhanced dimerization and/or oligomerization [171]. This has been suggested previously in conjunction with homologous GPCRs [172–174], pointing to a general signaling mechanism in this family of proteins [175, 176]. Homo or hetero dimerization of kinase domains is often observed functional requirement along with phosphorylation when activating signaling pathways in general [40, 177]. L482, however, is the first residue in the simulation model after the missing kinase domain, hence the precise role of signal transduction from the orthosteric site to the ICL3 kinase domain remains to be explored in more detail.

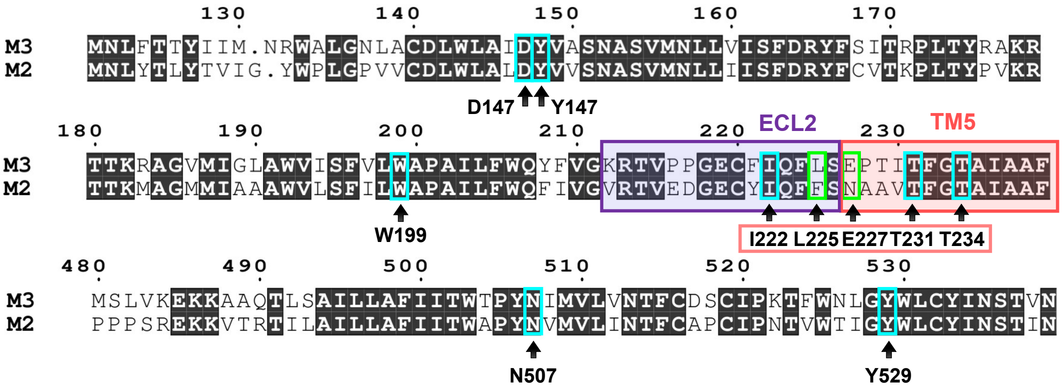

130
140
150
160
170
M3 MNLFTTYIIM.NRWALGNLACDLWLAIDYVASNASVMNLLVISFDRYFSITRPLTYRAKR
M2 MNLYTLYTVIG.YWPLGPVVCDLWLALDYVVSNASVMNLLIISFDRYFCVTKPLTYPVKR
D147
Y147
180
190
200
210
220
ECL2
230
TM5
M3 TTKRAGVMIGLAWVISFVLWAPAILFWQYFVGKRTVPPGECFIQFLSEPTITFGTAIAAF
M2 TTKMAGMMIAAAWVLSFILWAPAILFWQFIVGVRTVEDGECYIQFFSNAAVTFGTAIAAF
W199
I222 L225 E227 T231 T234
480
490
500
510
520
530
M3 MSLVKEKKAAQTLSAILLAFIITWTPYNIMVLVNTFCDSCIPKTFWNLGYWLCYINSTVN
M2 PPPSREKKVTRTILAILLAFIITWAPYNVMVLINTFCAPCIPNTVWTIGYWLCYINSTIN
N507
Y529

**Figure 5.11:** Protein sequence alignment of hMR2 and hMR3 for selected regions involved in the unbinding process. Key residues identified by MLTSA are distinguished as conserved (red) or non-conserved (green) between the two receptors. The ECL2/TM5 region is also highlighted (purple and salmon).

Other key residues also include distant locations that are near the ends of helical domains, similarly to L482: K93, K212, T514, D517, and N561. Some of these residues were identified as important by mutational studies, such as K212V and D517A, that decrease the tiotropium residence time in hMR3. Near T514 and D517, C519 was also previously identified as a key residue for RFI in PCA components 23 and 59.

The ECL2 loop remains key in this dataset as well, besides T234, F224 and S226 are also highlighted (Fig. 5.12.a). This is validated by the F224A construct, as mentioned earlier, where the half-life of **1** is halved [128]. In summary, RAD and RFI show a consistent picture, pointing to the key relevance of the ECL2/TM5 junction, in agreement with the previous results.

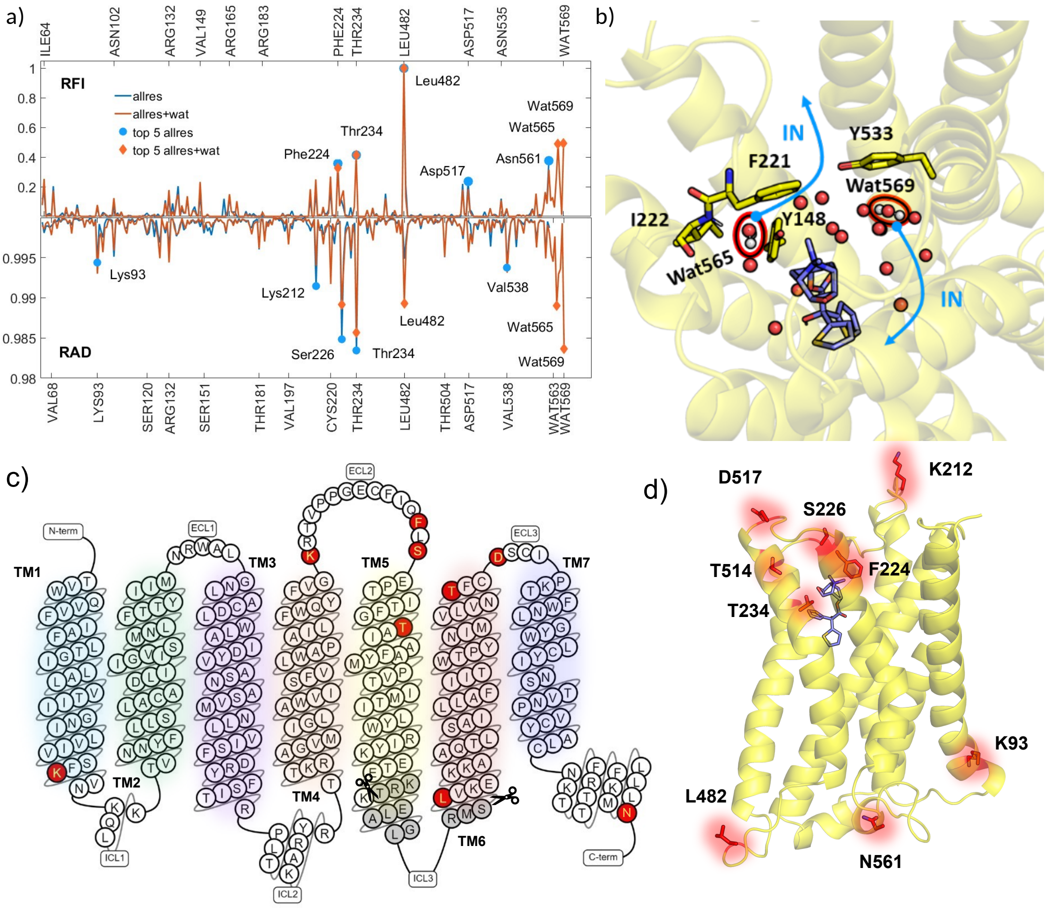


**Figure 5.12:** a) RFI and RAD for the allres (blue) and allres+wat (orange) datasets, highlighted are the top 5 residues for each approach (blue circle and orange diamond, respectively). b) TS snapshot showing the top two water molecules as well as nearby residues as stick in the allres+wat dataset. The blue arrows highlight the displacement of the water molecules upon re-entering of ligand 1 in the binding site. c) Diagram representation of the sequence of hMR3 portraying the different secondary structure motifs. In red, the top residues found decisive for the outcome by our MLTSA. In grey, the residues (kinase domain) not included in the simulation system. d) Top important residues from MLTSA highlighted in the 3D representation of hMR3, mostly corresponding to the ECL2/TM5 junction and the different ends of the alpha helices throughout the receptor.

**Water plays a role in the unbinding path.** Solvent molecules are known to play a crucial role in ligand unbinding kinetics [12, 178–181]. By both enabling the favorable electrostatic environment and orchestrating movements via hydrogen bonding, water molecules play a role that is often difficult to elucidate. To explore the role of water during the unbinding process, I included the 8 closest water-ligand distances together with the allres dataset as additional features. I found a modest increase in both MLP and GBDT prediction accuracy ( 81% and 79%, respectively). With these additional set of features, both RAD and RFI ranked the same two water molecules (Fig. 5.12.a-b, labelled 565 and 569) within the top 5 features. L482 remains ranked 1st for RFI, and it is 5th for RAD. Both approaches consistently find L482, T234, F224 and S226 most relevant together with water molecules (Fig. 5.12.c-d). This finding suggests that movements of water molecules in the pocket are also decisive to ligand unbinding in addition to the residues highlighted previously.

Water 565 is located near the ECL2 residues F221, I222, and forms H-bonds with Y148 and the backbone of I222, part of the ECL2/TM5 junction I highlighted throughout this work. Upon analyzing the most likely distances for IN and OUT trajectories, I observed that this water gets displaced in most of the trajectories as the ligand enters the orthosteric site. On the other hand, water 569 is on the other side of the ligand, closer to Tyr533, as well as Tyr529, which also forms the tyrosine cage. It only partially forms H-bonds with other water molecules, and it is located near a hydrophobic region of the pocket. While for OUT trajectories this position is not likely to change significantly, for IN trajectories the water moves deeper into the binding pocket as the ligand moves down into the orthosteric site.

## 5.4 Conclusions and Further Work

I generated and obtained consistent unbinding paths from hM3R for three ligands: tiotropium (**1**) and its analogues, **2** and **3**. All three ligands showed similar unbinding paths, including a first rotation of the charged end and a movement of the aromatic rings of the ligand, followed by a dislocation of the tyrosines forming the aromatic cage, finishing with a 90° angle rotation corresponding to the bottleneck while moving towards the receptor tunnel. Therefore, all ligands show a well-defined similar TS position and leave the orthosteric site in a highly homologous mechanism. The main barrier contribution in the unbinding process is known to be related to the ligand leaving the orthosteric site [128, 132, 133], therefore I did not follow up the subsequent full exit out of the vestibule. The obtained paths are also in agreement with previous studies — using much less resources, in an almost fully automated approach — on the unbinding paths of tiotropium for both hMR3 and hMR2, where the ligand exits in a similar way [128, 133]. Addtionally this approach provided more insights due to the ML analysis.

I further validated the TS structures by generating unbiased downhill simulations, which allowed us to further analyze the main events driving the unbinding at the TS. The first Cartesian coordinate-based (*XYZ-PCA*) dataset showed a remarkably good accuracy at predicting the outcome of the simulation at very early times. This first analysis suggested relevance of the ECL2 loop and the residues at the ends of the transmembrane helices but proved hard to interpret. A more local but high-resolution (*3Å*) dataset, which included the relevant protein binding pocket – ligand atomic distances at the TS structure, matched experimentally relevant residues such as D147, Y148, Y529 and pointed to T231 which is part of the ECL2/TM5 junction. An increased dataset (*6Å*) continued to point towards ECL2/TM5 junction contributions being the most relevant. I further tested the relevance of this region by augmenting the previous 3A dataset with these residues (*3Å+ECL2/TM5*). This further justified the key role of the ECL2/TM5 junction. On the other hand, adding e.g., ECL3 residues to the 3A dataset instead did not yield relevant distances from the ECL3 region. This further validated the relevance of the highlighted residues from ECL2/TM5, which also show differences in the protein sequence compared with hMR2 (L225/F181 and E227/N192 substitutions), highlighting potential role in the residence time differences between the two receptors.

Several residues identified by the MLTSA were previously experimentally mutated, further validating their importance in residence time. The available mutations show the largest influence for F224A, Y529A and N507A in the unbinding kinetics, while D147A, W199A, E227A, K212V and D517A impact it to a lesser extent. Additional residues I identified here as highly relevant remain yet to be experimentally probed for their role in ligand unbinding kinetics, such as L482, together with the preceding S481, as well as T234 remain to be further studied. Other identified residues that could play a role are: C220, I222, L225, S226, T231.

The results point to the structural importance of key ligand groups and consistently found specific atoms in the amine end, the carboxyl group, and the tiophene rings, to be highly relevant. All three pharmacophore groups match other variants of tiotropium that have a charged end, middle carboxyl group and an aromatic ring at the end, either one or two [128]. This analysis can therefore provide useful information to propose pharmacophores in future drug design studies for kinetics-based ligand optimization.

To account for all residue interactions with the ligand, a dataset with coarser interaction features (*allres*) was also used. This confirmed the importance of the ECL2/TM5 junction, and furthermore pointed to residues at helical ends. Additionally, when the closest ligand-water distances are added to the previous set (*allres+wat set*), two water molecules also appear at the top. The results suggest an important role of these molecules, whereby their movement is highly correlated to the ligand entering the orthosteric binding pocket. Importantly, L482 remains to be a top-ranked feature, near a phosphorylation site (S481 for PKG II) [166, 170] and between two specific binding regions for signaling and activating proteins (SET and PPase 2) [168, 169]. Interestingly, S481 phosphorylation was linked to enhanced dimerization in an allosteric mechanism upon antagonist binding [171], proposed to be a general mechanism in GCPR signal transduction [172–174]. This suggests that the conformational changes of the ECL2/TM5 junction at the TS crossing transduce a signal across the membrane to the intracellular ICL3 kinase domain of the receptor as the ligand exits or binds the orthosteric site. Our MLTSA analysis appears to capture and identify allosteric effects, opening up potential avenues in various other systems and processes as well [182, 183], beyond ligand unbinding. Nevertheless, the allosteric signal transduction remains to be studied in more detail, to aid the understanding of the function and mechanism of this biomedically-relevant receptor family.

## Chapter 6

# Atomic Charge Hamiltonian Replica Exchange

This work is currently under preparation for submission to a journal. It is at the late research stage, we are still aiming to do some more work which has not been included here, but the main findings are already mentioned in this chapter.

## 6.1 Introduction

Metal Ion coordination is crucial for the functioning of enzymes, more importantly for phosphate-related enzymes, where it is crucial for both recognition, regulation and catalysis for the active site geometries. Electrostatic interactions are also some of the major contributors to molecular interactions in biology for recognition, selectivity, dissociations, etc [39].

However, all of the major rearrangements needed to unveil these processes are also observed at quite long time scales. One example of a family of proteins that requires to undergo major changes to achieve its catalytically competent conformation is ribonuclease A (RNase A). This protein needs to undergo a complex series of changes to properly position its catalytic residues for RNA cleavage [184].The major changes that occur in the active site of RNase. A during catalysis involve the precise positioning and orientation of the catalytic residues (His12, His119, and Lys41) to enable the general acid-base catalytic mechanism. Additionally, conformational changes in surrounding residues help stabilize the substrate and transition state, allowing the enzyme to efficiently catalyze the RNA cleavage reaction through electrostatic interactions and hydrogen bonding [185]. MD is able to take a look at this interactions at the atomic level, however the time scales needed are much more than the ones available.

In order to aid in sampling the long timescales, enhanced sampling methods are being developed. Replica exchange MD (REMD) allowed for lowering the timescales by flattening the potential surface and observe rare events [54], however, although different temperatures accelerate the overall sampling, some other efforts are needed to focus on specific interactions. Hamiltonian replica exchange MD (HREMD) allowed for the alteration of other simulation parameters, such as interacting potentials [56]. Due to the relevance of metal ions and their role in biology, I developed a new flavour of HREMD, named atomic charge hamiltonian replica exchange MD (ACHREMD) which takes advantage of directly lowering the atomic charge of specific atoms to flatten the electrostatic potentials and allow for unbinding or dissociation events to take place much quicker. Major active site rearrangements are also relevant, which take long timescales for specifically complex systems, which is something the ACHREMD can excel at.

## 6.2 Computational Details

### 6.2.1 Toy Model trajectories

To be able to asses the possibility of unbiasing the data obtained from all replicas, a Markov Chain Monte Carlo (MCMC) sampling model potential was used, with a temperature ladder made of 30 different windows geometrically spaced. This model is created to imitate the systems described by the MD and validate the unbiasing in an easier example. The different replicas were run for a total of 105 “temperature” replica exchange attempts, having a local Monte Carlo (MC) step size of 102 before attempting an exchange. The model potential (see 6.2 for true potential shape) was described by the following equation

$$U(x) = G\sin\left(\frac{5}{1.195}x + \sin^{-1}(-1)\right) + \frac{Gx}{10} + 5.8 \tag{6.1}$$

Where $U(x)$ is the energy that is assigned to the $x$ coordinate and $G$ acting as a "temperature" term that changes from 0.1 to 5 between all replicas. The Metropolis-Hastings (MH) sampling technique was the criteria followed to evaluate exchanging attempts between replicas.

### 6.2.2 Hamiltonian Replica Exchange on Atomic Charges

In our implementation, two sets of relevant atoms are drawn from both molecules i.e. ligand and receptor, or anion and cation. These will have their topology's atomic charges tampered through the different replicas. Assuming the ligand's relevant atomic charges are positive and the receptor's negative, they will be tampered with by gradually decreasing the charge in the first group while correspondingly increasing the second group's charges to prevent breaking detail balance. For instance, an overall decrease of 0.5 in the first group, will increase the second groups charge by 0.5. For the MD trajectories, a high frequency of exchanges was achieved by running a small local sampling trajectory step, which ensured an equal sampling between charges. At every exchange attempt, the energy of the last frame was recalculated at every other charge configuration using NAMD [103]. Using the Metropolis-Hastings criteria, the acceptance/rejection of the configuration swaps was calculated and a new local sampling trajectory step was started. The energies of the trajectories at other configurations were also recalculated using the namdenergy plugin from VMD [103] which I used in the data analysis for the unbiasing.

### 6.2.3 Test Systems

#### NaCl Complex

In order to first evaluate the method on an MD simulation I implemented the ACHREMD on a system consisting of a Na+ atom as the positive ion and a Cl- atom acting as a counterion, with both charges being changed accordingly to maintain charge equilibrium throughout the different replicas. The NaCl complex was solvated in explicit water, in a box of 10 angstroms containing 1698 water molecules. The simulations were run using the NAMD software under the CHARMM36m [152] force field. As a baseline, I ran a non-biased MD simulation of 1.32 µs from the bound state to reconstruct the free energy profile and compare to classical MD, by extracting the distance between the ions throughout the trajectory and using them as a Reaction Coordinate (RC). For the ACHREMD procedure I spaced the replicas over 9 charge windows ranging from +1 to +0.2, with a distance between replicas of -0.1 on the Na+ charge, tampering the Cl- charge with +0.1 as a result. The value of the windows for the NaCl system can be found on Fig. 6.3 panel c. I run the simulations with a time step of 2 fs and recorded the coordinates every 1 ps making an exchange attempt every 10 ps (local sampling step) for 32500 exchanges which sums up 325 ns per replica (totalling 2.95 µs of simulated time). The same data resolution was used for the non-biased simulation.

## EDTA-Mg Complex

For the purpose of simulating a more complex system with higher barriers and more degrees of freedom while being similar to a ligand-binding system yet still simple to describe, I applied the ACHREMD to EDTA bound to a $Mg^{2+}$ ion. The bound state was obtained from an EDTA-Fe complex found in the 1ZLQ PDB structure which was later swapped from $Fe^{+2}$ to $Mg^{+2}$. This system was solvated in explicit water as well, with a box of 20 angstroms and 10395 water molecules. Simulations were run with NAMD and the CHARMM36m force field as well. In this case the charge replicas were linearly spaced from +2.0 to +0.4 over 31 replicas. The charge on the Mg+2 was reduced across replicas while the difference was being distributed equally towards the ligand's (EDTA) N and O initially bound to the ion (2 N and 4 O from the carboxylates). A non-biased simulation starting from the bound state was run for 6.18 µs as a baseline comparison. The replica exchange simulations were run for exchanges (5000) every picosecond, totalling 150ns of total simulation time. The reaction coordinate in this case was the log of the sum of distances between the 4 carboxylate atoms (C1, C4, C5 and C6), the 2 N (N1 and N2) atoms and the Mg ion. Simulations were run at a resolution of 2 fs per step, 0.6 ps of local sampling were run before attempting a replica exchange. I recorded the coordinates of the system along with the RC every 0.06 ps. Since there is a long range of values for the RC to take, but the most relevant ones are at the beginning, the logarithm of the

RC will be used.

### 6.2.4 CRISPR Cas1/Cas2 Complex

After exploring the available high-resolution Cas1-Cas2 integration complex PDB structures, no single model contained a full integration complex with all 4 Cas1 monomers, Cas2, the protospacer double-stranded DNA, and the target host DNA all unzipped and ready for catalysis. The available PDB crystal structures have either no metal ions resolved (PDB: 4QDL), a zipped DNA strand bound and no metal ions resolved in the active site (PDB:5VVJ, 5VVK ), the metal ions are just to stabilize the DNA itself (PDB:5XVP, 5VVL), or the DNA strands don't meet at the active site (PDB: 5XVN, 5XVO). Additionally, no structures contain both the protein complex with all 4 Cas1 monomers and the cas2, and the protospacer DNA strands, plus the target DNA all unzipped as reactants. Thus, an effort for combining several structures and coming up with a suitable model (see Fig. 6.1) was done. This required a chimeric model, comprised of different subunits and structures, which is a combination of PDBs 5VVJ and 5VVL. Due to the difficulties in bending the target DNA they could not be connected among the two active sites.

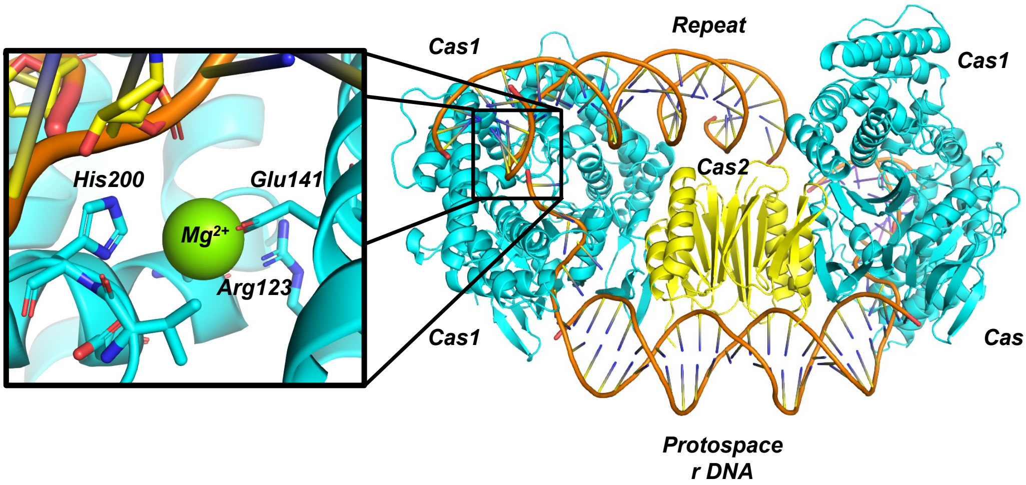


**Figure 6.1:** Chimeric model and active site of the proposed model for the CRISPR Cas1/Cas2 Complex

Taking into account the previously mentioned sampling problem, for a system this big (around 350000 atoms), the chances of witnessing a rare event, or even a big conformational change are low. By lowering the value of the divalent metal charge from +2 to values around 0, one lowers the energetic barrier around the residues involved in coordinating the substrate. This allows for more frequent crossing between high energy barriers and explore other conformations. By looking at the most stable conformations, one can find a more desirable arrangement of the active site than the one started with. After adding dummy K+ atoms for countering the ion charge lowered on the $Mg^{2+}$ and thus preserve detail balance among all replicas. The framework had 72 windows ranging from values of +2.00 (upper limit) to +0.4 (lower limit). The ACHREMD was run for more than 30 exchange attempts, sampling for 0.5 ns per step and totalling an approximate of 1.2 µs in an attempt to sample alternate active site geometries.

## 6.3 Results and Discussion

### 6.3.1 Toy Model Validation

After running the replica exchange for the toy model, the PMF can be recovered from every single window alone and all windows give back the correct values. Fig 6.2 shows the recovered profile from using data from all the different replicas. During a replica exchange or parallel tempering the original methodology calls for disregarding the data that is not run in the original conditions, i.e., the base temperature or unbiased state, and the exchanges are only useful for getting new configurations in the unbiased state and sample in an unbiased way. This is very inefficient specially for a big number of temperatures or charges, in this case. Now that I showed this is possible using free energy estimators such as WHAM/DHAM or MBAR, we will move to try it in a more complex system.

### 6.3.2 NaCl Dissociation

After running the NaCl system during several steps, for both a biased an unbiased state, it is advised to check first that there is overlapping between replicas. Fig. 6.3.a shows the histograms of the energies found during the simulated trajectories for each replica in each charge/state. A shift in the mean of each distribution is easily noticeable since the electrostatic potential is directly affected. There is good overlap, which ensures for a good exchange probability for sampling. Additionally,

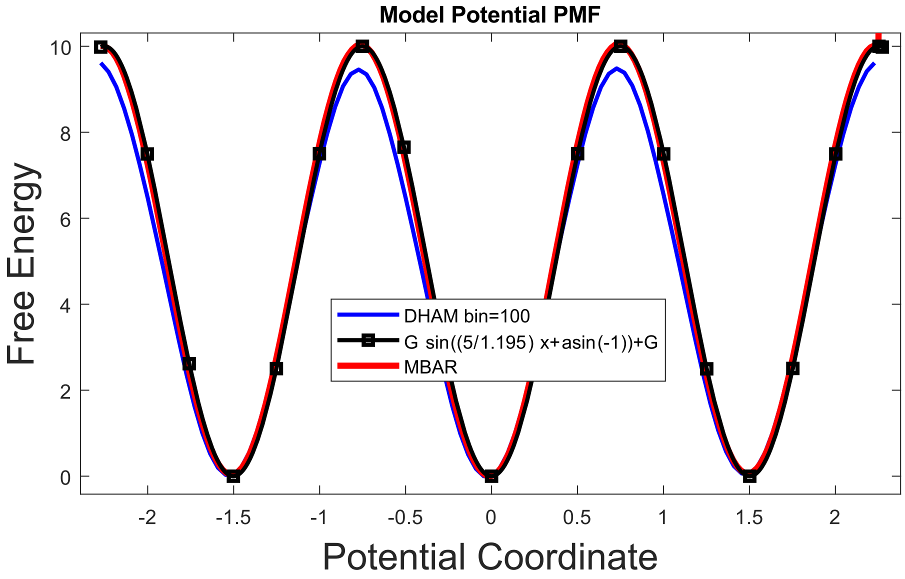


**Figure 6.2:** Free energy profile recovered using all replicas in the temperature ladder of the toy model potential. The original theoretical potential profile is shown in black, the recovered profiles using both MBAR and DHAM are shown in red and blue, respectively.

when looking at the sampled reaction coordinate (distance between the ions). one can also notice both overlapping and the changes in the distributions (Fig. 6.3.c).

The free energy profile of the NaCl complex can be **easily described by a one-dimensional representation**, such as the interatomic distance between both atoms. Using the data produced on the unbiased MD simulation started from a bound state, I was able to produce an accurate free energy profile to use as a theoretical value. For this, I used both the multistate Bennet acceptance ratio (MBAR) analysis and the general transition-based reweighting analysis method (TRAM) with 0 bias, which produced similar profiles shown in Fig. 6.4. For the ACHREMD simulations, using the same analysis methods as previously described I unbiased the data produced on the different replicas, obtaining an **accurate profile** as shown in Figure 6.4. This is consistent with previous studies [186].

### 6.3.3 EDTA-Mg Dissociation

For the EDTA-Mg case, the approach proved to be very **useful to observe the dissociation** and sample the rare event, whereas **a long unbiased MD simulation was never able to sample the unbound state** or see the dissociation event. However,

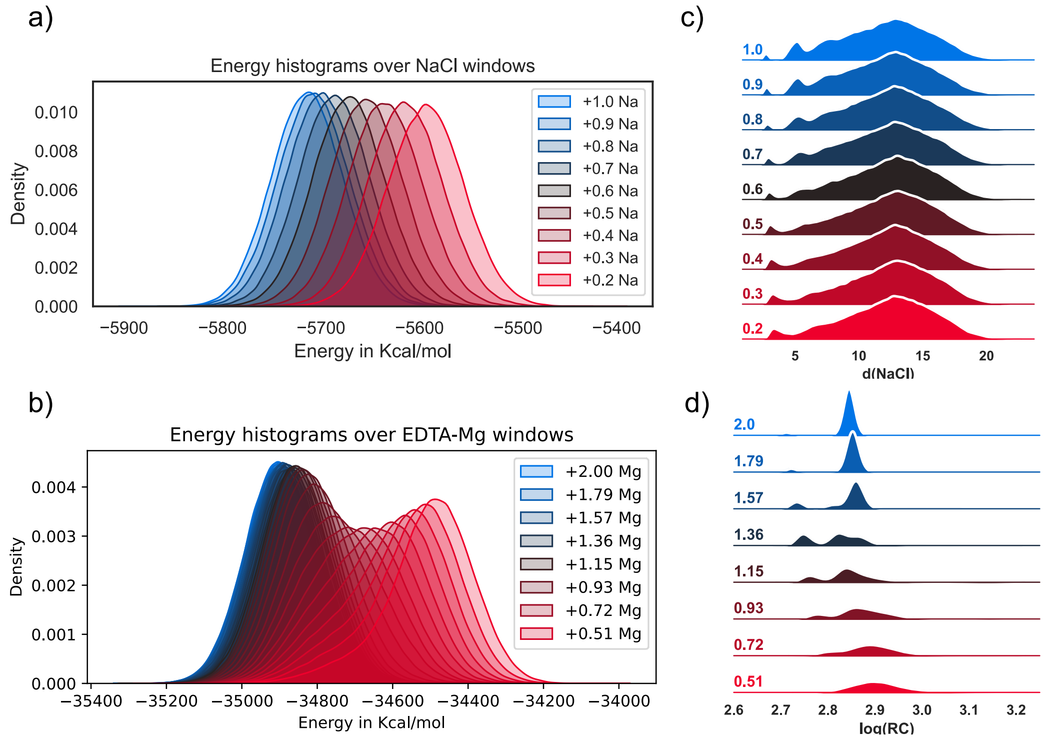


**Figure 6.3**

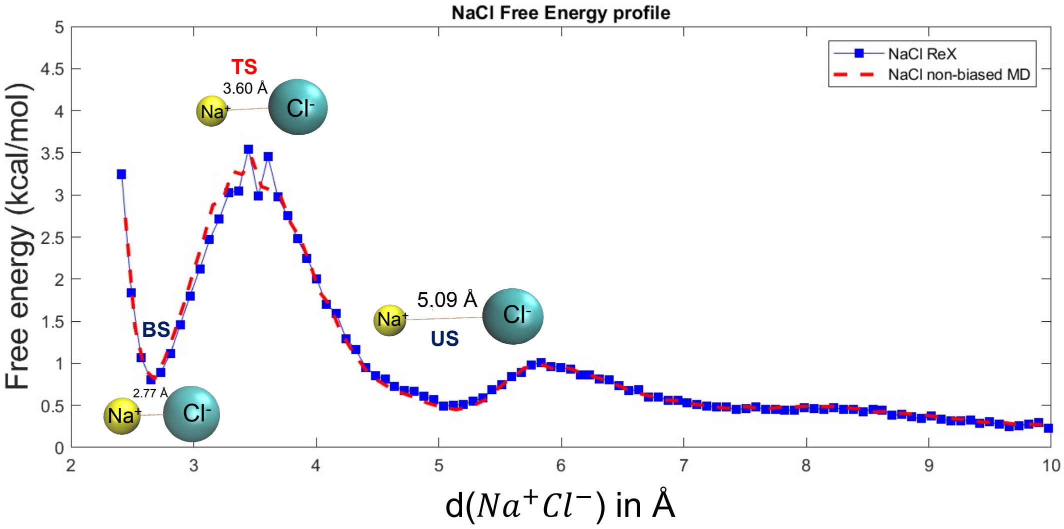


**Figure 6.4:** Reconstructed free energy profile for the unbiased MD data and the achremd replica exchange.

the reconstruction of **the profile was not as accurate**. I believe this is due to the low likelihood of this dissociation to happen, the barrier is so high that it is very difficult to elucidate the probability of transition, is also not sampled as it is very unlikely. However this transition has been observed for several replicas, but it is very fast,

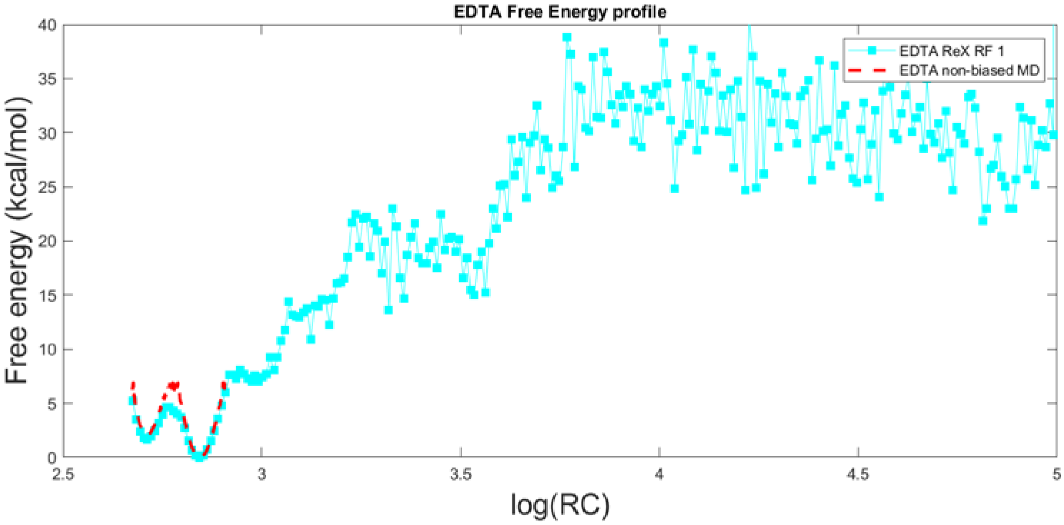


**Figure 6.5:** Recovered free energy profile from the unbiased MD (red, dashed) and the ACHREMD run (blue) for the EDTA-Mg Complex. MBAR was used for the unbiased MD and WHAM for the replica exchange.

thus having a very high energy barrier to cross (see Fig. 6.5). The reaction coordinate used for this is described in the computational details, section 6.2.3. I also attempted to reproduce the experiment using Umbrella Sampling directly and recovering the profile. After having to use a very big force constant and properly placing 50 windows (see Fig. D.1), I could ensure overlapping between the windows. After sampling for 200 ns, the obtained profile was not yet properly recovered, as seen in Fig. D.2. However, when US is compared to the bound states sampled by the long unbiased MD, the profile is totally off from the original for the states between 2.5Å and 3Å, which are the only value sampled in unbiased MD.

### 6.3.4 CRISPR-Cas1/Cas2 Active Site Rearrangement

After running 1.2 µs, it was observed in several windows at different charges, a **reorganization of the active site residues** which enabled a transient potassium ion entering the active site to coordinate nearby residues and complex itself with the substrate and the other metal ion. This fact suggested that a reorganization of the active site was indeed needed and suggested that there is the **possibility for this active site to coordinate 2 metal ions**, involved as well in catalysis of the DNA phosphate, which is common among NTP enzymes. Some examples of the snapshots found in these windows is available in Fig. 6.6. Of all different binding

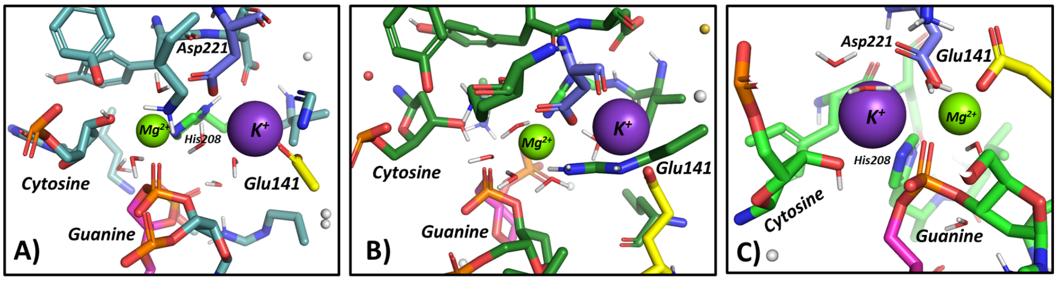


**Figure 6.6:** Different active site geometries after the transition of a K+ (purple) at different trajectories during the enhanced sampling. Glu141 in yellow, Asp221 in deep blue, His208 in light green. (A) Shows the transient ion entering from the right side and coordinating Glu141 while His208 coordinates the Mg2+(in Green) and Asp221 isn't interacting at all. Guanine base from the DNA (GUA) is coordinating the Mg2+. (B) Shows a transition from the right side again, coordinating both Glu141 and Asp221 but moving away His208. GUA is not coordinated. (C) Shows the best candidate coordinating all three residues (His208, Asp221 and Glu141) and both the GUA and the CYT (Cytosine) coordinated by both metal ions.

modes found, the most suitable one (Fig. 6.6.C), contained distances chemically relevant for the reaction and was selected to do further calculations and validate this geometry.

Using the aforementioned structure from the enhanced sampling and replacing the K+ atom with a $Mg^{2+}$ and swapping the charges back to normal (+2.0) , 2 constrained MD replicas were run for several nanoseconds while slowly releasing the constraint to allow for the solvent and nearby residues to accommodate to the sudden change of metal ion charge. From 20 Kcal/mol to 0 Kcal/mol during a 100 ns run, the active site did not rearrange and the distance between both $Mg^{2+}$ and between CYT:O and GUA:P remained constant and close to standard 2 metal coordinated distances. The evolution of these distances is portrayed in Figure 6.7.

## 6.4 Conclusions and Further Work

This work has shown that the ACHREMD framework works good for both the toy model and the NaCl complex, while having a sub-optimal profile for the EDTA-Mg. However, it has proven useful to sample the desired event easily, and outperform umbrella sampling at obtaining a path with intermediates. It is advised to use this for specific systems where the electrostatic are key to flatten the landscape that

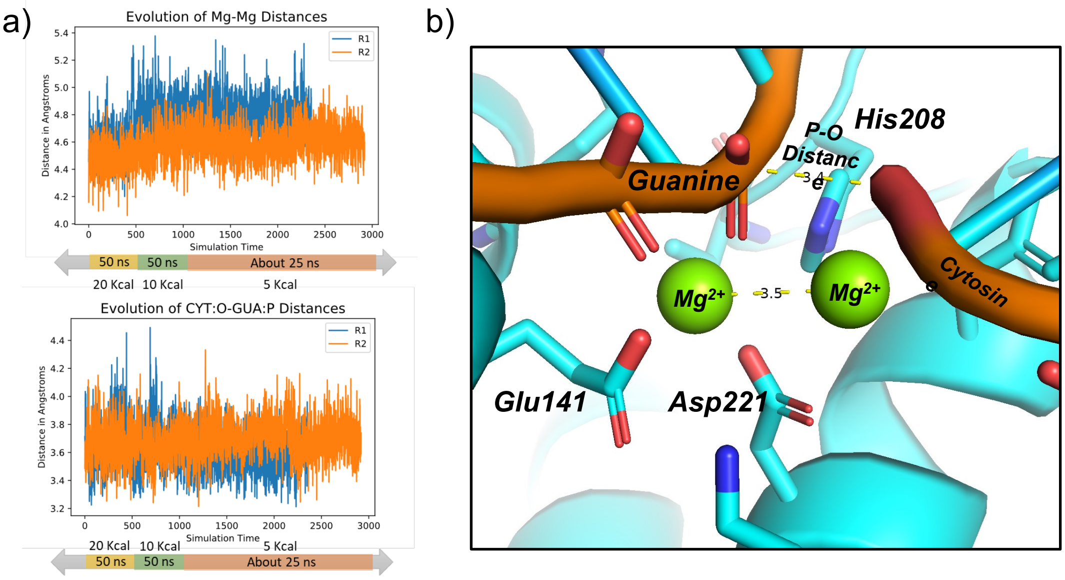


**Figure 6.7:** (a) Plots of the evolution of distances during the two replicas of constrained MD ran from the previously mentioned best candidate for active site geometry shown in (b). Top: Evolution of the distance between magnesium ions during the slow constraint release. Bottom: Evolution of the distance between the Cytosine's Oxygen (O) and Guanine's Phosphate group (P). (b) Pymol representation of the active site geometries used to run. Same geometry as previously shown in Figure 13C but the $K^+$ is changed to a $Mg^{2+}$.

otherwise are hard to be sampled.

Regarding the CRISPR model, the achremd has allowed for the reorganization, which allowed for further hypothesizing the active site's geometry. The proposed geometry and the idea of having a two-metal ion-based catalysis is not new for a CRISPR complex. A previous study has shown the DNA cleavage in CRISPR-Cas9 to be a two-metal based mechanism [187], having a water-assisted catalysis. However, more investigation is needed to confirm a similar mechanism occurs for the Cas1/Cas2. Further improved resolved structures/models and more experimentation are needed as well to study its catalysis and dynamics.

Further experimentation on other systems with ACHREMD is required. I hope this work further pinpoints the current challenge that the EDTA-Mg presents. While being a simple system, it's high energy barriers are difficult to sample and have enough transitions between to be able to properly describe its kinetics. When attempting to use a one-dimensional collective variable to describe it, one is unable to properly capture high energy barriers, mainly to the high-dimensionality that governs this transition, which gets lost in **hysteresis**.

Hysteresis, in the context of protein conformational dynamics, refers to the observation that the pathway taken to reach a certain structural state affects the ease with which the protein can return to its initial state. This means that the energy barriers for transitioning between different conformations can vary depending on the direction of the transition. In essence, the protein's behavior can show memory of its past states and transitions, leading to a non-reversible or asymmetric response in its conformational changes. In the scenario where the chosen CV is inadequate or insufficient to accurately capture the true pathway connecting two distinct states of a system, hysteresis can manifest as an artificial delay or inconsistency in the transitions observed during molecular simulations. The system might exhibit conformational changes that appear sluggish or biased due to the limitations of the selected collective variable. This discrepancy can hinder the simulation's ability to accurately model the actual dynamics and hinder the identification of valid transition pathways between states. Additional efforts to solve this problem and more refined free energy estimators should be developed in order to tackle these challenges, this being a relevant problem in the MD field. Once again, ML can bring forth the field with either connecting the states with generative models [188, 189], or by learning new reaction coordinates [190] that allow for better more detailed transitions.

## Chapter 7

# General Conclusions

To provide some successfull conclusions, I will revise first some of the main contributions of this work:

- A **computational protocol for ligand unbinding** has been used succesfully and validated for both a kinase domain and a transmembrane protein.
- A **framework for analyzing relevant interactions** in molecular dynamics has been explored and validated.
- Testing **analytical models for assessing feature analysis techniques** with time-series with time-dependent and independent relevance have been developed.
- A **protocol for enhanced sampling simulations** on systems with electrostatic-relevant interactions has been established.

After revising the main contributions I will break these points down and provide a more conclusive note.

Ligand unbinding paths were obtained for both CDK2 and hMR3 for three different inhibitors each. Unbiased simulations for observing the unbinding of these systems is out of the timescales one can currently achieve easily. This is true especially for tioropium and hMR3 which has a 24h residence time. Not only it was **able to observe this events, but in the case of CDK2 recover the kinetics of the unbinding**. In the case of hMR3, such a complex transition, especially with so many different degrees of freedom can be highly degenerated and prove difficult to observe consistently, making it hard for the free energy estimators to recover their PMF. This is even more pronounced for the second barrier (complete exit of the ligand), where it has multiple allowed paths to cross from the extracellular vestibule to the bulk. Additionally, by identifying the transition state, the exiting was sampled thoroughly and analyzed in more detail. This allowed for gaining insights on the tendency of the ligands, their exiting contacts, major rearrangements, etc. This information is invaluable for drug design efforts, and if its speed and accuracy are improved, it could lead to a protocol for virtual screening of drug molecules and determination of residence time. This should be compared with other similar methods for obtaining ligand-unbinding, such as number of contacts metadynamics, or any other method that can achieve this. Although metadynamics may take long to converge, the string method used in our protocol is quite slow and computationally expensive, whereas the ”unbinding protocol” itself is quite fast. This first protocol could be followed by another optimization instead of the string method for instance. The development of any combination of our method is encouraged and more experimentation is needed to improve the method. Currently, the optimization of the path is both too slow, and computationally demanding, especially for big systems and in complex cases, it proves difficult to converged to a refined unbinding path. Thus, a more scalable and efficient path refinement method is advised. There is a need for improvement in this field towards more affordable computational protocols for drug discovery.

The MLTSA is able to predict the outcome at very early times, in a matter of picoseconds (less than 1 ns) with very different sets of data. Being the dataset made from active site distances, or simply XYZ coordinates, the models are still able to forecast the outcome. As shown in this work, although both MLP and GBDT do not differ a lot in performance, in some cases complex input data works better with MLP, whereas sometimes GDBT is better at ranking and detecting relevant features. Other advanced architectures may be able to capture the time-dependent relevance differently, transformers, for example, would be a good candidate for testing both the ability to predict from early times and its attention mechanism allows for the visualization of the weights to each feature. This is probably the rational next step to go towards in the hopes of further developing the ML approaches to understanding MD data. Combining the MLTSA and the unbinding could provide, with slight improvements, a near **fully automated framework for the assessment of ligand molecules and the understanding of residence times**.

Validation, of course is still in need when working with ML. The analytical models developed in this thesis aim to create a dataset that closely resembles a complex time series derived from a real-world problem, exhibiting both complexity and time-dependency, similar to the challenges faced when working with protein data from MD simulations. As mentioned in this work, spiral datasets are one of the most challenging datasets for classification tasks. Although both the 1D and 2D analytical model were designed with the MLTSA analysis in mind, they are still **useful for testing other approaches in a computationally cheap fashion**. They can both pave the ground for more time-series related testing with room for customization and problem-tailoring. In the 1D model, the number of input features can grow as much as needed, and its degree of mixing as well. In the 2D model, the framework is already prepared for using custom potential shapes, there is already a Z-shaped potential available. Additionally, other projections/transformations to the original data con be done in order to add complexity to the predictions. On the spiral shape, one can also increase the number of outcomes/classes with a parameter, similar to a case where a protein has several microstates, leaving the ground for potential methodologies able to characterise states in complex MD systems.

The ACHREMD **successfully dissociated complex stable systems** such as EDTA-MG, and allowed for **major rearrangements even in big complex systems** such as CRISPR Cas1/Cas2. Although I was able to recover the free energy profile for NaCl and the toy model, in the case of the EDTA-Mg complex this was not possible. It remains unclear if this is because of the complexity of the system or the high energy barrier, since the produced profile using US was still noisy and inaccurate. However, observing the dissociation was fast using ACHREMD without the need of strong force constants that could lead to inaccuracies and artifacts, which is favourable in this context. This protocol was problem-tailored and hopefully will enable researchers to observe rare events closely bound to electrostatic interactions. Regarding the more complex system, Cas1/Cas2, note this is preliminary and there is still room for improvement and further validation. The idea of having a 2 metal ion based catalysis has to be further explored and tested, as well as an updated model of the complex. Hopefully the proposed model will set the challenge for other experimental groups to structurally resolve both reactant and product states, and hopefully an intermediate.

Overall, this thesis has contributed **towards challenges in molecular dynamics by both developing and exploring methods to sample rare events** (unbinding protocol and ACHREMD), and making use of the high-dimensional data these simulations produce to gain molecular insights (MLTSA). This will allow for further progress towards drug discovery and enzyme design. Not only that, but the complementary tools and methodologies (MLTSA and analytical models) will also aid the data analysis community towards the development of explainable AI, which will hopefully bloom soon.

## Appendix A

# MLTSA Supplemetary Information

## A.1 Additional Trainings

### 1D Analytical Model

**Table A.1:** Average training accuracy for a higher complexity (higher degree of mixing between potentials) for the GBDT and MLP).

| **Model** | **Train Acc.** $<\%>$ | **Test Acc.** $<\%>$ | **Validation Acc.** $<\%>$ |
|---|---|---|---|
| *GBDT* | 100 | 100 | 90.53 |
| *MLP* | 92.80 | 91.60 | 93.33 |

## A.2 Higher complexity/Degree of mixing

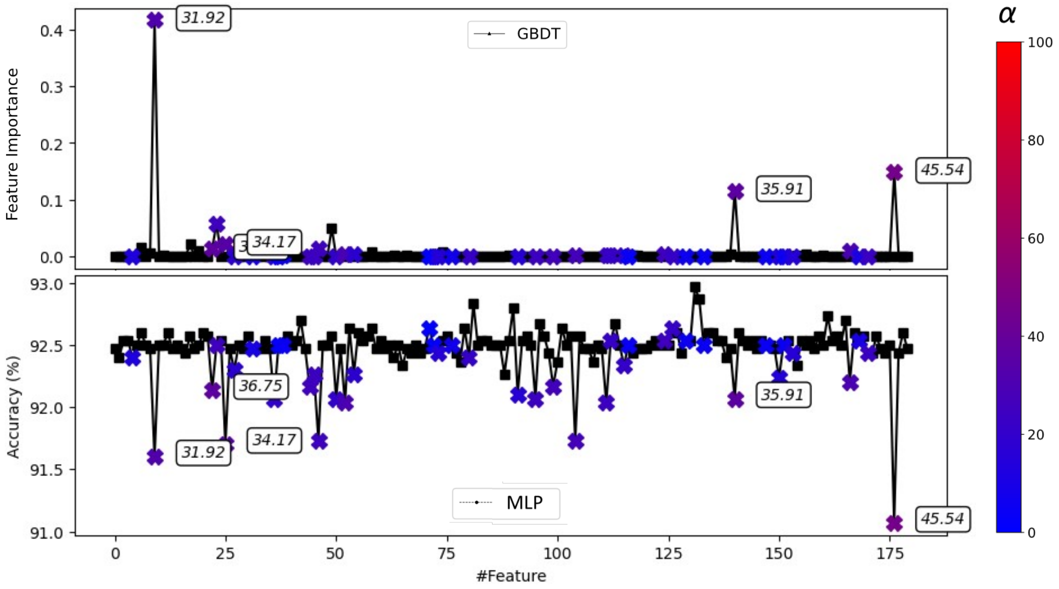


**Figure A.1:** Comparison between GBDT (top) and MLTSA with MLP (bottom) feature analysis methods for the higher complexity and degree of mixing datasets. Correlated features are marked from blue (0%) to red (100%) depending on the mixing coefficient, $\alpha$ (x symbols, color scale on the right, five highest mixing coefficients also displayed for the datapoints). Uncorrelated features (small black symbols) are at 0 FI for GBDT and show no loss of accuracy for MLTSA with MLPs. Correlated features all show a significant AD for the MLP, while only the top correlated features have high FI using GBDT.

## A.3 LSTM Training on 1D

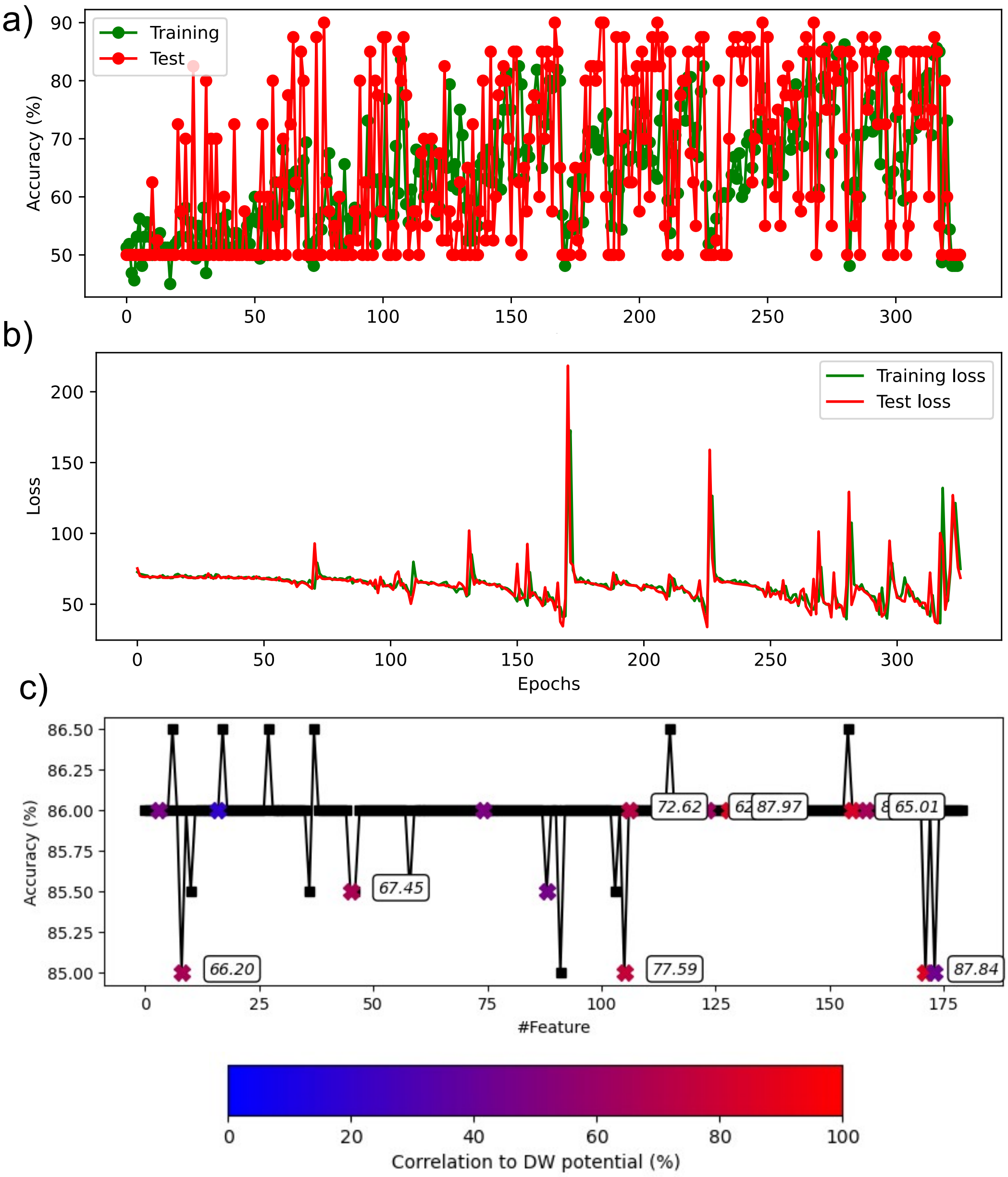


**Figure A.2:** a) shows training evolution through epochs for test and training sets. b) shows the evolution of the loss as well for both sets. c) is the resulting accuracy drop results.

# Appendix B

# CDK2 Supplementary Information

## B.1 MD Details

The initial atom coordinates for the three systems were built using high resolution crystal structures with the following PDB codes: 3SW4 (Resolution=1.7 Å), 4FKU (Resolution=1.47 Å), and 4FKW (Resolution=1.8 Å). I present the results for 4KFU in the SI Section. The systems were modelled using the AMBER ff14SB force field, and the ligands using the general Amber force field (GAFF). The ligand's atomic partial charges were obtained using density functional theory (DFT) $\omega$B97X-D/def2TZVPP3 as implemented in Gaussian 09 Revision E. The full system was solvated with 12,000-14,000 TIP3P water molecules. Na+ and $C^{1-}$ ions were added to neutralize the system and set a salt concentration of 0.14 M. All the MD simulations were performed using NAMD 2.12.

The three systems were first minimized using a standard protocol via the steepest descent algorithm for a total of 150,000 steps followed by the equilibration of the restrained protein (1 kcalmol$^{-1}$Å$^{-2}$ force applied to each heavy atom of the protein) for 10 ns in NVT ensemble at 300 K via a standard MD procedure. All the production runs were performed with the NPT ensemble with a time step of 2fs. Pressure was maintained at 1atm by a Nosé-Hoover Langevin piston. Temperature was maintained at 298 K using Langevin dynamics with a damping coefficient $\gamma$ of 0.5 ps$^{-1}$ applied to all atoms. SHAKE was applied to all bonds involving hydrogen and nonbonded interactions were calculated with a cutoff of 12 Å, and a switching distance of 10 Å. The particle mesh Ewald method was used for long-range electrostatic calculations with a grid density of $> 1$ Å$^{-3}$.

An initial unbiased simulation of 20 ns was performed for each ligand. This initial simulation allows the system to equilibrate and gives us an initial trajectory to calculate the first CVs.

## B.2 Atom Clustering

Residues with atoms that have a rotational degree of freedom with multiple equivalent positions are clustered together. During the unbinding process, if a new contact is found with one atom belonging to the cluster, then the harmonic restraint will be applied to the centre of mass of the selected clustered atoms. The use of clustered atoms reduces the fluctuation caused by the rotation of such bonds, affecting the overall harmonic restraint.

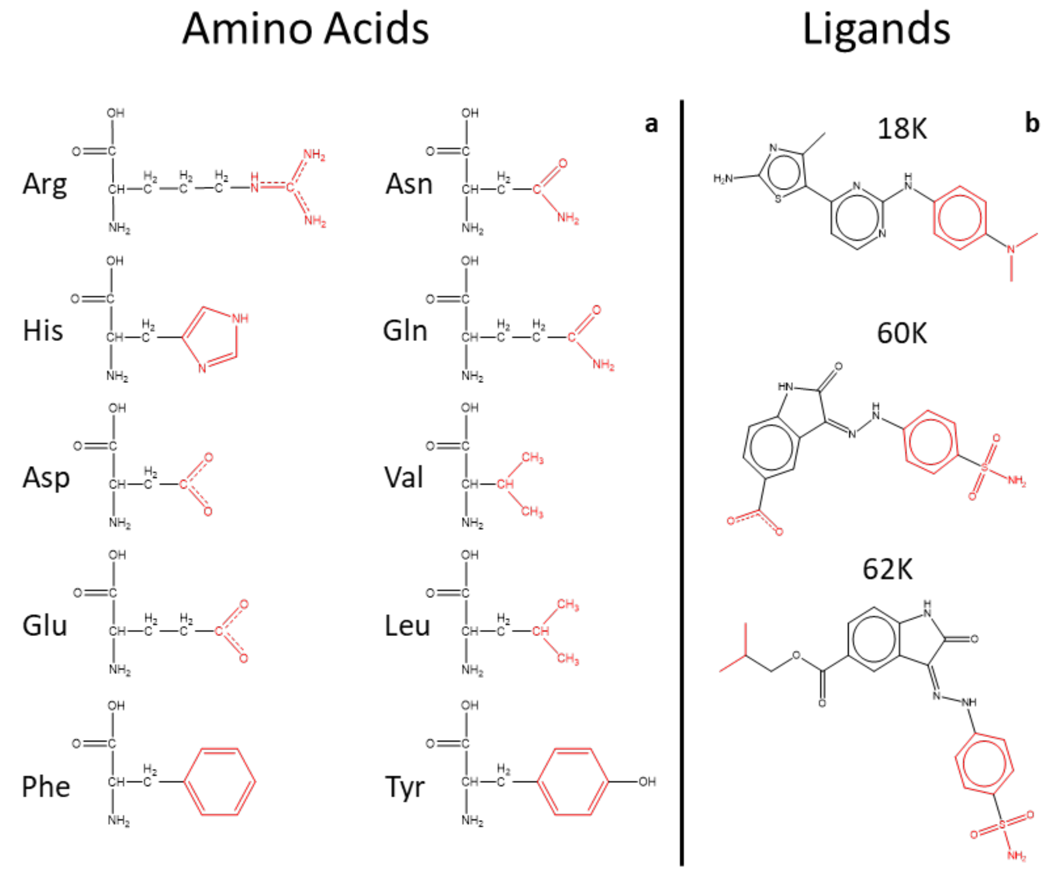


**Figure B.1:** Chemical structures of the residues with clustered atoms, highlighted in red; a for the amino acids and b for the ligands.

## B.3 Labeling and Input Features

The classification of the ligand in the bound position (IN) and unbound position (OUT) is calculated by analyzing the last 250 ps of the downhill trajectories. For each frame I extracted and sum two key distances between the ligand and the protein (see Table B.1) and average these for all the frames of the last 250 ps. If the sum of these distances is below a given IN-threshold the trajectory is classified as IN, if the value is above the OUT-threshold then is classified as OUT (see Table B.1).

**Table B.1:** Key distances used to automate the IN/OUT labelling of the 5 ns-long downhill trajectories. These are used to create a dataset suitable for the ML algorithm to learn the classification with the selected CVs as inputs (X) and the labels IN/OUT as targets (Y).

| System | Distances | IN-threshold | OUT-threshold |
|---|---|---|---|
| *3sw4* | LIG(N9)-LEU83(O) - LIG(N7)-LEU83(N) | 10 | 12 |
| *4fkw* | LIG(N3)-LEU83(O) - LIG(O9)-LEU83(N) | 11 | 13 |

## B.4 ML Training

To understand the relationship between the accuracy of predictions and the data used to make those predictions, I trained the MLP with several different datasets. As described in the main text, each trajectory provided a set of distances from the simulated trajectory at particular timeframes, and each dataset was made up of a set of such timeframe elements. The trainings used different timeframes of the trajectories: at 0.3, 0.5, 0.75, 1, 1.5, 3 and 5 ns. For each of these datasets I calculated the accuracy of the predictions for each of the three systems. The models provide good accuracy from the very initial frames of the simulations. For example, at 0.1 ns I have an accuracy of 79.5% for ligand 18K and 83.6% for 62K.

Details of the trained models during the MLTSA using 0.1, 0.15, 0.3, 0.5, 0.75, 1, 1.5, 3, and the full 5 ns length of the downhill trajectories for each system (4fkw and 3sw4) are listed below. In addition to testing the different lengths of trajectories, the percentage of data to use from the latter end of the trajectory at each time frame (i.e., the 50% latter end of 0.1 ns would correspond to data from 0.05 ns to 0.1 ns) was also tested. The number of simulations available and the number of epochs

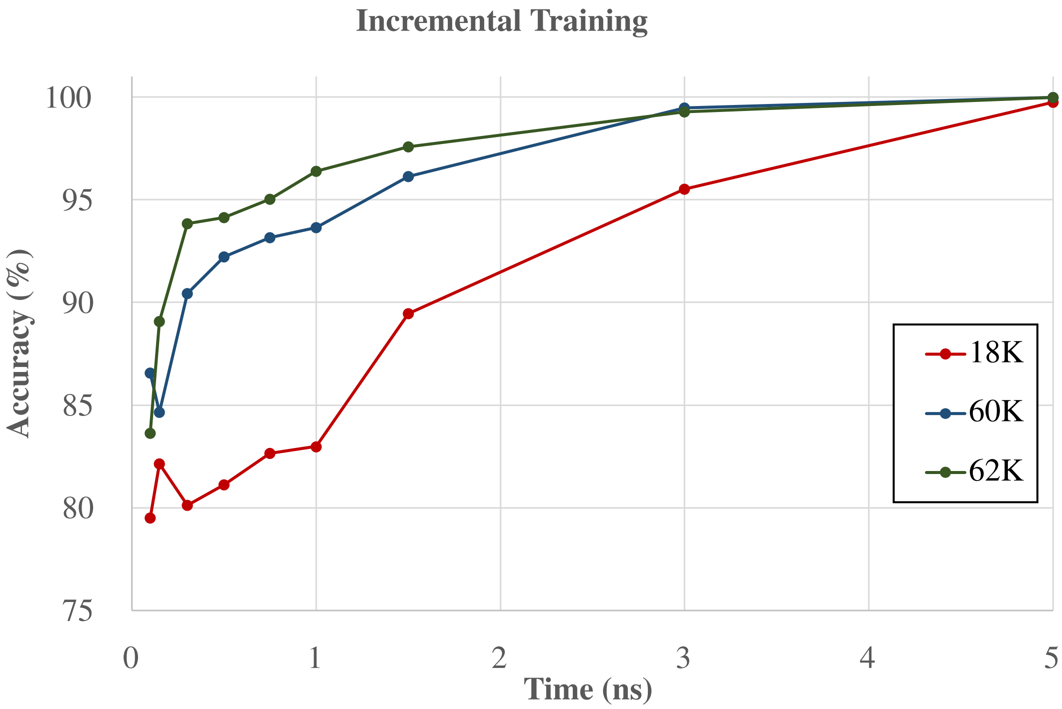
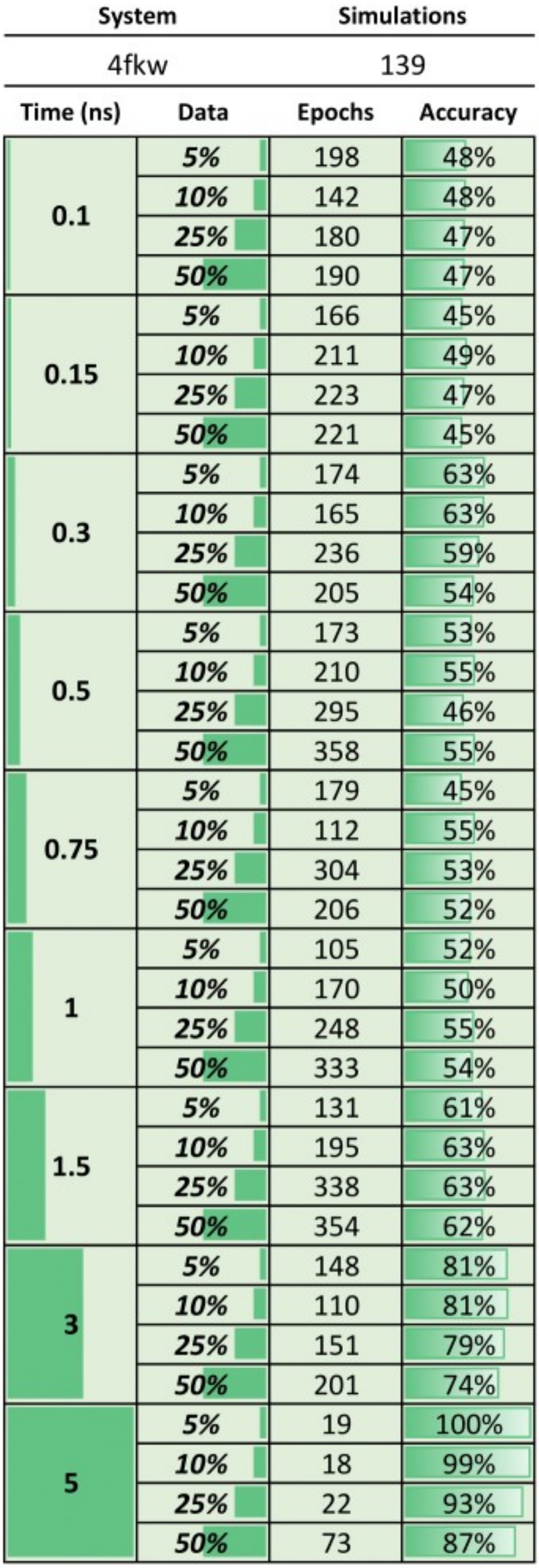


**Figure B.2:** ML accuracy prediction at different time frames using MLP for 18K in red and 62K in green.

until convergence for each model are also listed, as well as their accuracy on a set of independent simulations (Validation set). This set is comprised of the 25% of the available data from 4fkw and 3sw4 having 35 and 37 simulations to test the accuracy, respectively.

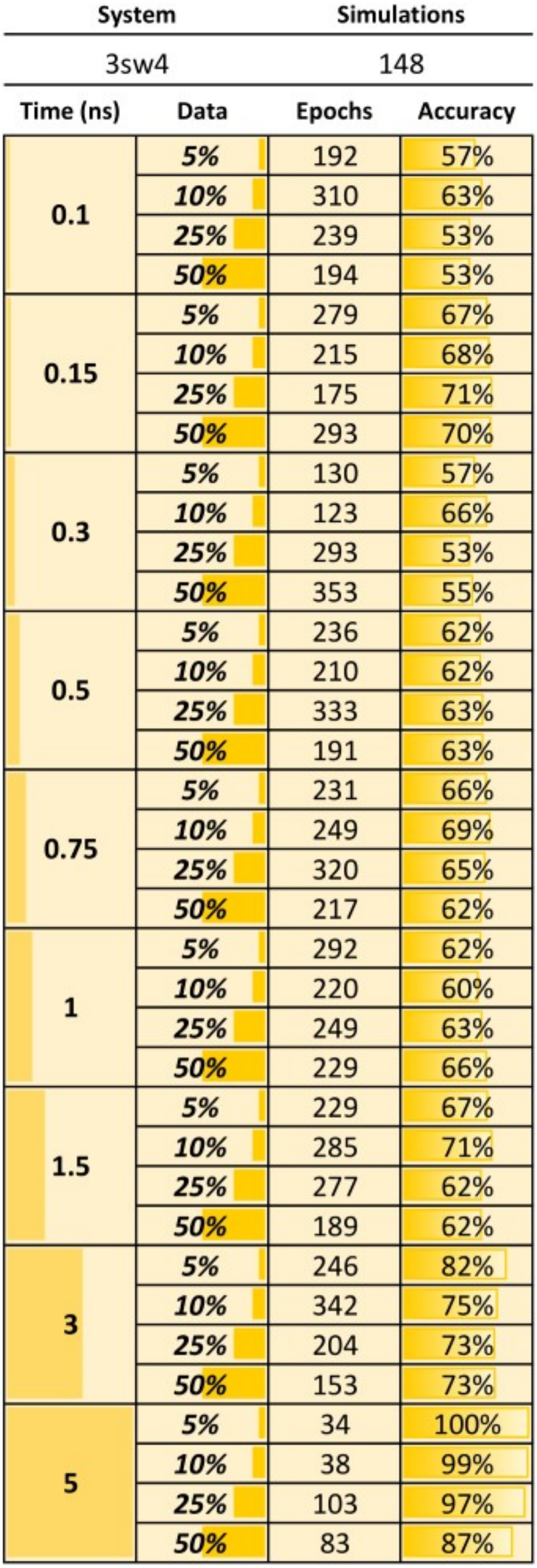
The tables below comprise the details of the models tested on 4fkw and 3sw4 data as well as their accuracy on the validation set. The first column corresponds to the time frame of trajectory data used from the beginning. The data column corresponds to the percentage of latter simulation time used to train each model. The third column has the number of epochs until convergence of the model and the last column shows the accuracy on the validation set.

**Table B.2:** Trainings at different time-frames for the 4fkw dataset



| System | | Simulations | |
|---|---|---|---|
| 4fkw | | 139 | |
| **Time (ns)** | **Data** | **Epochs** | **Accuracy** |
| **0.1** | ***5%*** | 198 | 48% |
| | ***10%*** | 142 | 48% |
| | ***25%*** | 180 | 47% |
| | ***50%*** | 190 | 47% |
| **0.15** | ***5%*** | 166 | 45% |
| | ***10%*** | 211 | 49% |
| | ***25%*** | 223 | 47% |
| | ***50%*** | 221 | 45% |
| **0.3** | ***5%*** | 174 | 63% |
| | ***10%*** | 165 | 63% |
| | ***25%*** | 236 | 59% |
| | ***50%*** | 205 | 54% |
| **0.5** | ***5%*** | 173 | 53% |
| | ***10%*** | 210 | 55% |
| | ***25%*** | 295 | 46% |
| | ***50%*** | 358 | 55% |
| **0.75** | ***5%*** | 179 | 45% |
| | ***10%*** | 112 | 55% |
| | ***25%*** | 304 | 53% |
| | ***50%*** | 206 | 52% |
| **1** | ***5%*** | 105 | 52% |
| | ***10%*** | 170 | 50% |
| | ***25%*** | 248 | 55% |
| | ***50%*** | 333 | 54% |
| **1.5** | ***5%*** | 131 | 61% |
| | ***10%*** | 195 | 63% |
| | ***25%*** | 338 | 63% |
| | ***50%*** | 354 | 62% |
| **3** | ***5%*** | 148 | 81% |
| | ***10%*** | 110 | 81% |
| | ***25%*** | 151 | 79% |
| | ***50%*** | 201 | 74% |
| **5** | ***5%*** | 19 | 100% |
| | ***10%*** | 18 | 99% |
| | ***25%*** | 22 | 93% |
| | ***50%*** | 73 | 87% |

**Table B.3:** Trainings at different time-frames for the 3sw4 dataset



| System | | Simulations | |
|---|---|---|---|
| 3sw4 | | 148 | |
| **Time (ns)** | **Data** | **Epochs** | **Accuracy** |
| **0.1** | ***5%*** | 192 | 57% |
| | ***10%*** | 310 | 63% |
| | ***25%*** | 239 | 53% |
| | ***50%*** | 194 | 53% |
| **0.15** | ***5%*** | 279 | 67% |
| | ***10%*** | 215 | 68% |
| | ***25%*** | 175 | 71% |
| | ***50%*** | 293 | 70% |
| **0.3** | ***5%*** | 130 | 57% |
| | ***10%*** | 123 | 66% |
| | ***25%*** | 293 | 53% |
| | ***50%*** | 353 | 55% |
| **0.5** | ***5%*** | 236 | 62% |
| | ***10%*** | 210 | 62% |
| | ***25%*** | 333 | 63% |
| | ***50%*** | 191 | 63% |
| **0.75** | ***5%*** | 231 | 66% |
| | ***10%*** | 249 | 69% |
| | ***25%*** | 320 | 65% |
| | ***50%*** | 217 | 62% |
| **1** | ***5%*** | 292 | 62% |
| | ***10%*** | 220 | 60% |
| | ***25%*** | 249 | 63% |
| | ***50%*** | 229 | 66% |
| **1.5** | ***5%*** | 229 | 67% |
| | ***10%*** | 285 | 71% |
| | ***25%*** | 277 | 62% |
| | ***50%*** | 189 | 62% |
| **3** | ***5%*** | 246 | 82% |
| | ***10%*** | 342 | 75% |
| | ***25%*** | 204 | 73% |
| | ***50%*** | 153 | 73% |
| **5** | ***5%*** | 34 | 100% |
| | ***10%*** | 38 | 99% |
| | ***25%*** | 103 | 97% |
| | ***50%*** | 83 | 87% |

For the models used in the study, these are the loss evolutions through time:

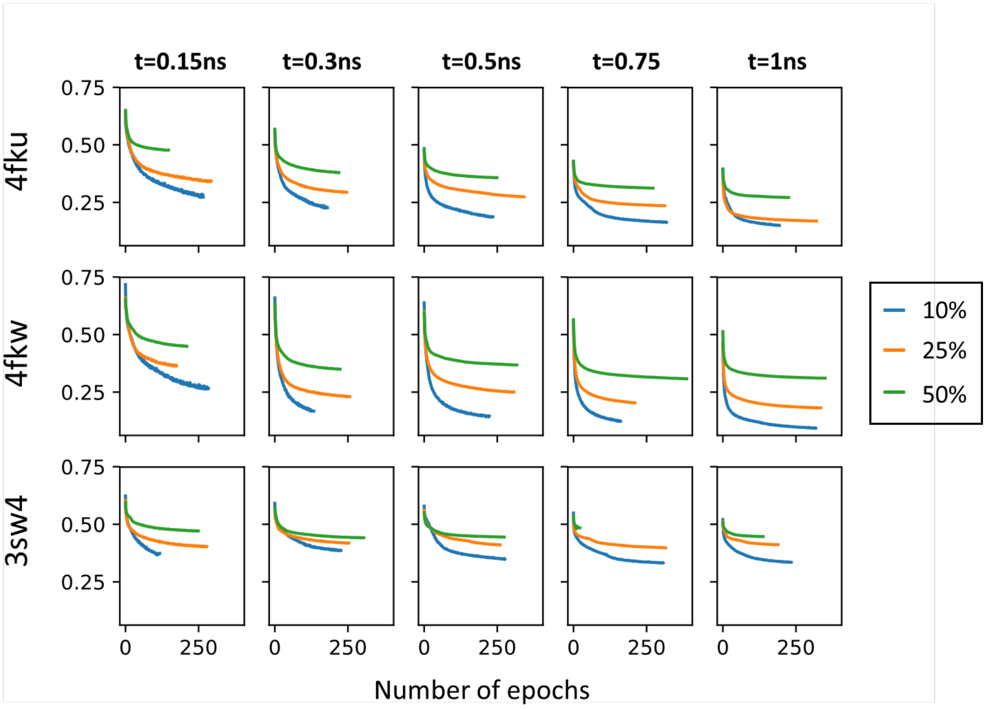


**Figure B.3:** Plots of the loss function evolution through the training epochs at different time frames with different percentages of data from the end for 3sw4 (18K ligand), 4fkw (62K ligand) and the excluded 4fku (60K ligand) CDK2 systems.

## B.5 Validation of ML Analysis

Figures S14.I and S14.II compare the results of the training against a simple binary classification model which attempts to classify the outcome as IN/OUT based on the CV values at a specific time (0.15, 0.3 and 0.5 ns). The dots show the CV values (as a sum of two key distances from Table B.1) and are colored according to their outcome, red as OUT and green as IN. I then calculated the accuracy of the binary prediction at different thresholds represented by the black bars to obtain the highest possible accuracy using a single cutoff value (blue arrow). I compared these with the values obtained from the MLP (blue data, top of Figs. S14.I-II) and the GBDT (yellow data, top of Figs. B.4, B.5, B.6).

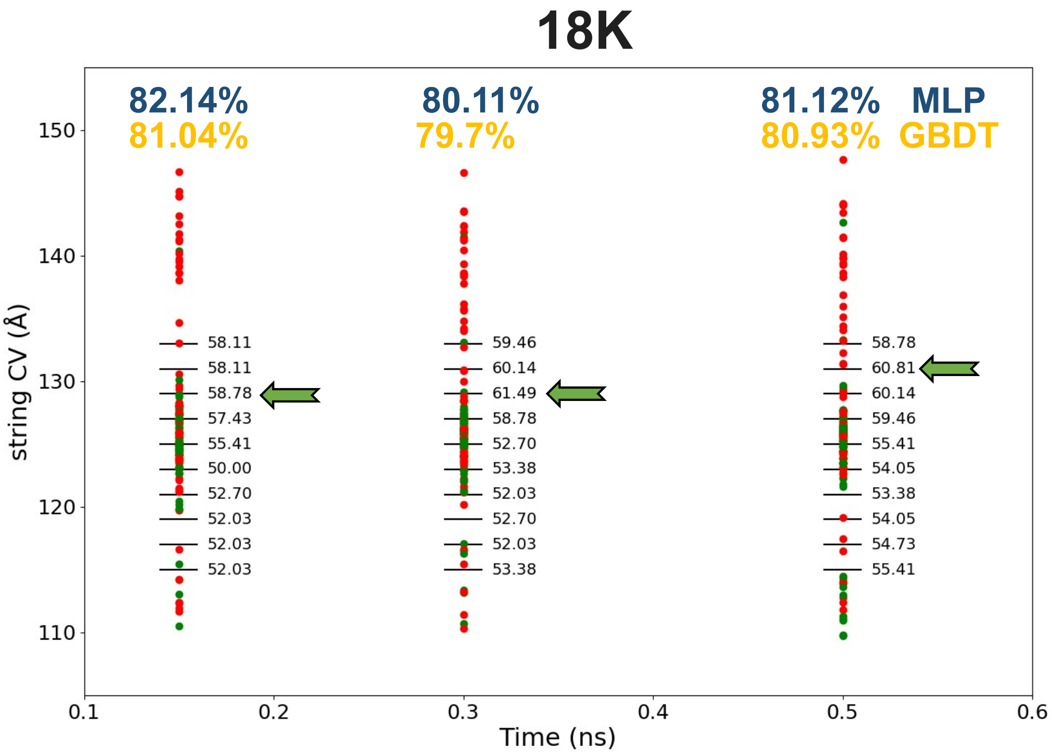


**Figure B.4:** Comparison of the accuracy obtained from the MLTSA training (blue data) and GBDT (yellow data) with a simple binary classification model for ligand 18K at 0.15, 0.3 and 0.5 ns. Data points corresponding to different trajectories show the actual value of the string CV for IN (green) and OUT (red) trajectories.

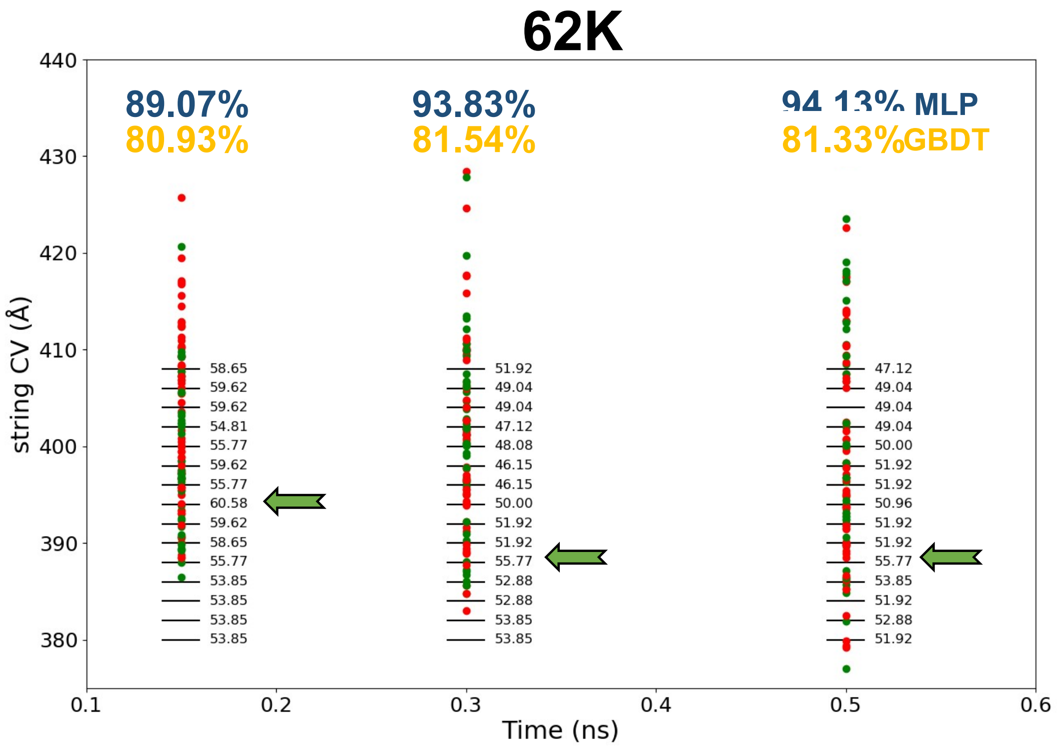


**Figure B.5:** Comparison of the accuracy obtained from the MLTSA training (blue data) and GBDT (yellow data) with a simple binary classification model for ligand 62K at 0.15, 0.3 and 0.5 ns. Data points corresponding to different trajectories show the actual value of the string CV for IN (green) and OUT (red) trajectories.

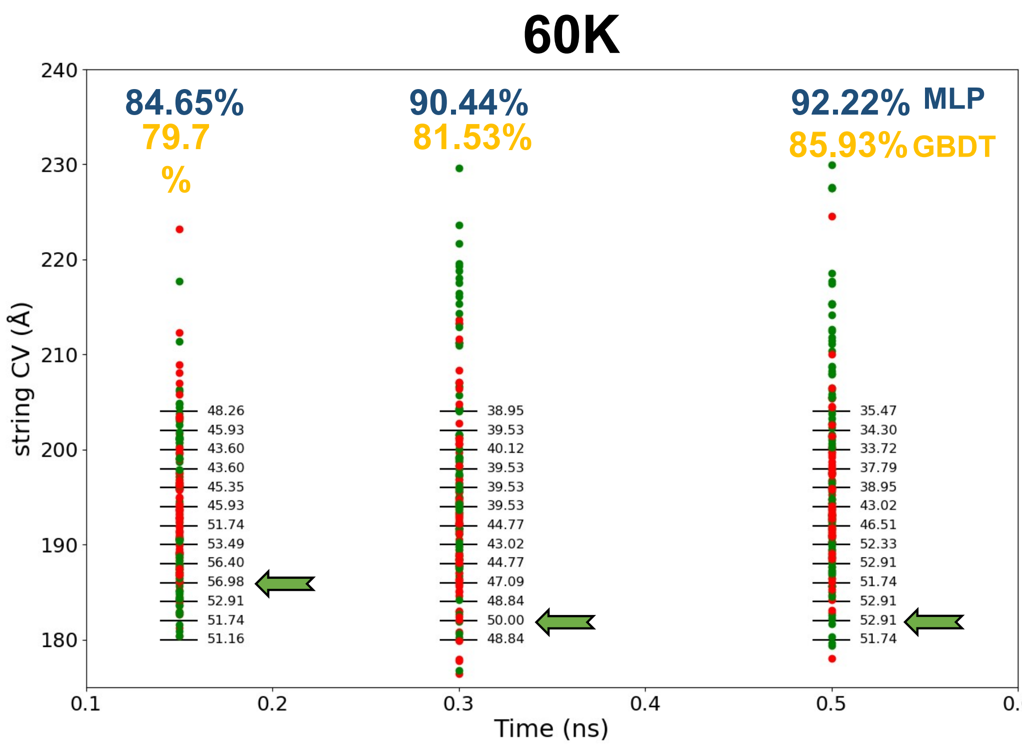


**Figure B.6:** Comparison of the accuracy obtained from the MLTSA training (blue data) and GBDT (yellow data) with a simple binary classification model for ligand 60K at 0.15, 0.3 and 0.5 ns. Data points corresponding to different trajectories show the actual value of the string CV for IN (green) and OUT (red) trajectories.

## B.6 Free Energy Profiles

For each system, I performed three independent replicas. The PMF is plotted along the string windows. For each replica, the number of distances included in the string depends on the unbinding trajectory. The number of distances used in each system are given in Table B.4.

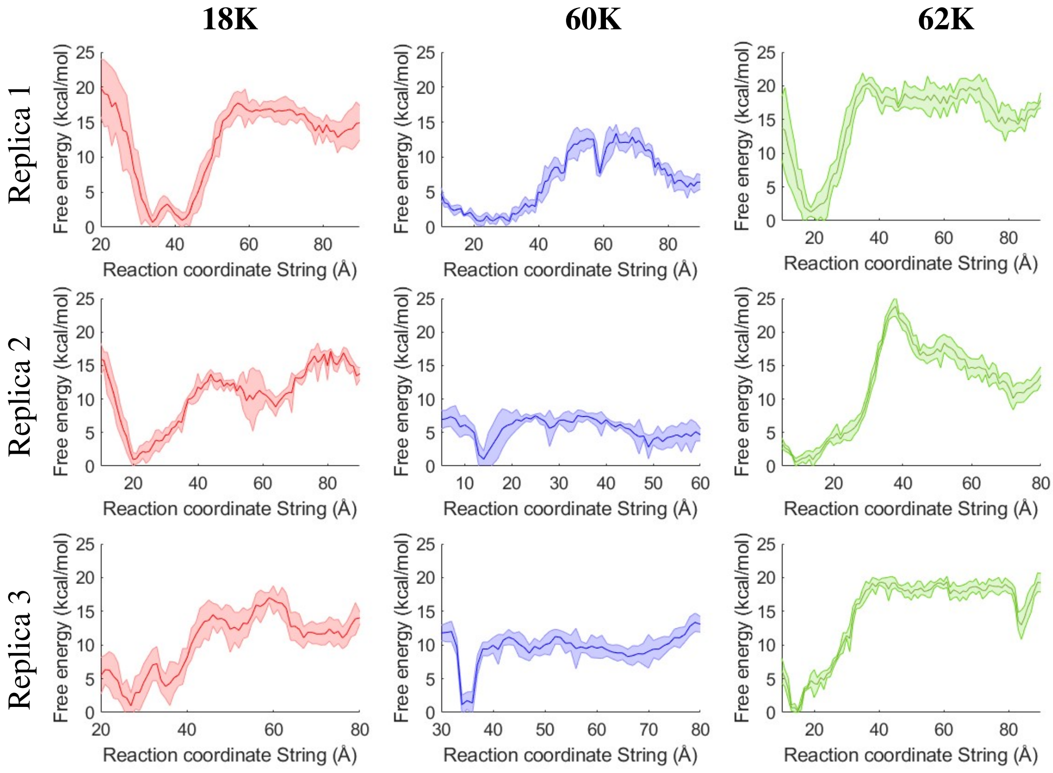


**Figure B.7:** PMF of the unbinding path for 18K, 60K and 62K. The standard error shown as shaded area are obtained by dividing the full dataset into 4 subgroups.

**Table B.4:** Number of distances included in the RC for each replica.

| System | Number of distances |
|---|---|
| *18K Replica 1* | 21 |
| *18K Replica 2* | 38 |
| *18K Replica 3* | 33 |
| *62K Replica 1* | 46 |
| *62K Replica 2* | 38 |
| *62K Replica 3* | 43 |

## B.7 60K/4FKU System

An additional ligand was tested with our unbinding approach, an oxindole carboxylic acid derivative (60K) based on the 4fku structure (Fig. B.8). The unbinding procedure was carried out as described for the other ligands. After performing the string calculations for the 4fku system, 60K presented a change in conformation, more specifically, a cis-trans conversion of the hydrazineyl N=C bond (Fig. B.9).

This conformational change would only be expected at very high energy costs, and it is a combined artifact of the force field and the biasing procedure. The Z (cis) to E (trans) conversion allowed the 60K ligand to unbind with a significantly lower free energy barrier than its analogue, 4fkw (Figs. B.10 and B.11). They both share a dihedral angle ($\phi$), which corresponds to this transformation, defined between atoms N6-N9-C14-C16 for 60K and N1-N3-C25-C26 for 62K (Fig. B.10). On one hand, this is partly due to the initially strong constraints from the string method that can be corrected in the future. On the other hand, this is also due to the too low energy of the trans form and the too low barrier for the isomerisation as compared to the DFT calculations (Fig. B.11). Tas a result, the final unbinding free energy barrier (Fig. B.7, middle, blue) is 10 kcal/mol lower than the experimental (20.01 (±0.12) kcal mol-1) value for all three replicas (9.96 (±1.5) kcal mol-1).

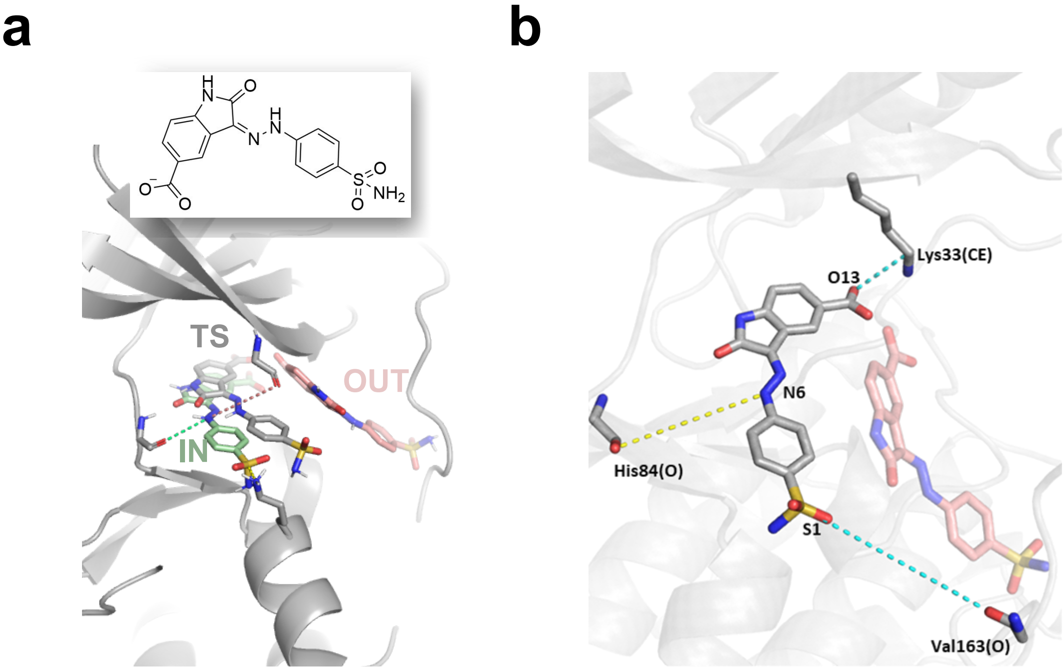


**Figure B.8:** Left (a): CDK2 bound to 60K, the chemical structure of the ligand oxindole carboxylic acid derivative is drawn in the inset. Bound state (IN) originated from PDB structure 4fku. Structural details of the ATP pocket are shown with the ligand in the bound state (green), unbound (red) and transition state (grey). Right (b): common CVs obtained from the unbinding replicas of 60K, representative distances are shown in dashed lines (yellow: interaction from the initial structure, cyan: interaction found during the unbinding trajectory), red sticks represent the ligand when it is outside the pocket. The displayed distances appear in all three replicas for 60K.

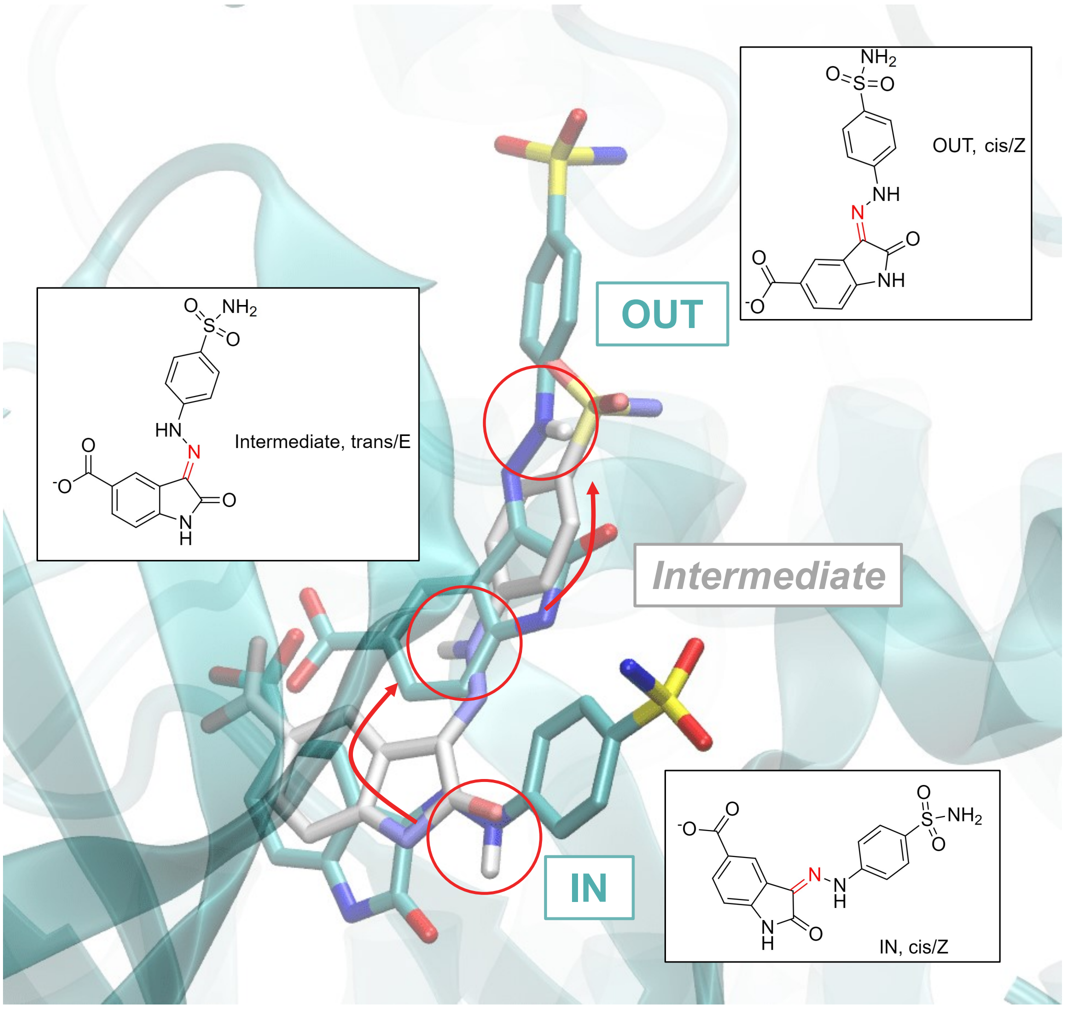


**Figure B.9:** 60K ligand structures within the CDK2 binding pocket from three different umbrella windows portraying the cis-trans conversion through the unbinding pathway from IN (cis/Z, red circle) via the intermediate (trans/E, red circle, white sticks) to OUT (cis/Z, red circle) structures.

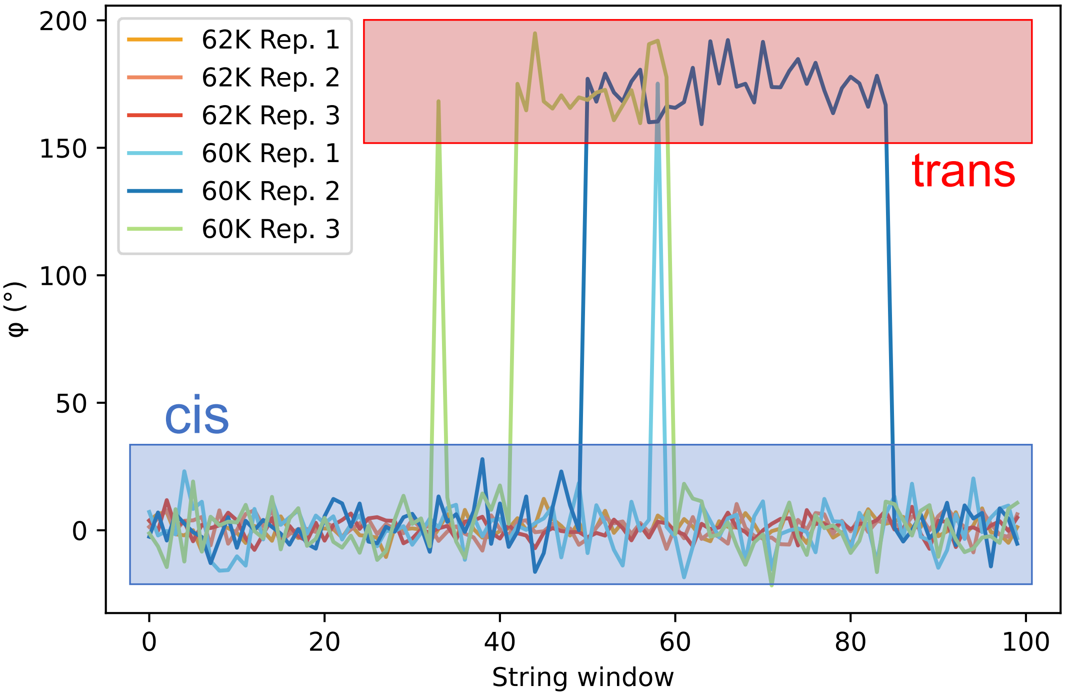


**Figure B.10:** Values of the dihedral angles ($\phi$ ) for both 4FKU (60K) and 4FKW (62K) throughout the umbrella sampling windows of all three replicas. $\phi$ is defined as the dihedral angle between atoms N6-N9-C14-C16 for 60K and C25-C26-N1-N3 for 62K. The conformations at $\phi$  0° correspond to the cis isomers and at $\phi$  180° correspond to the trans isomers.

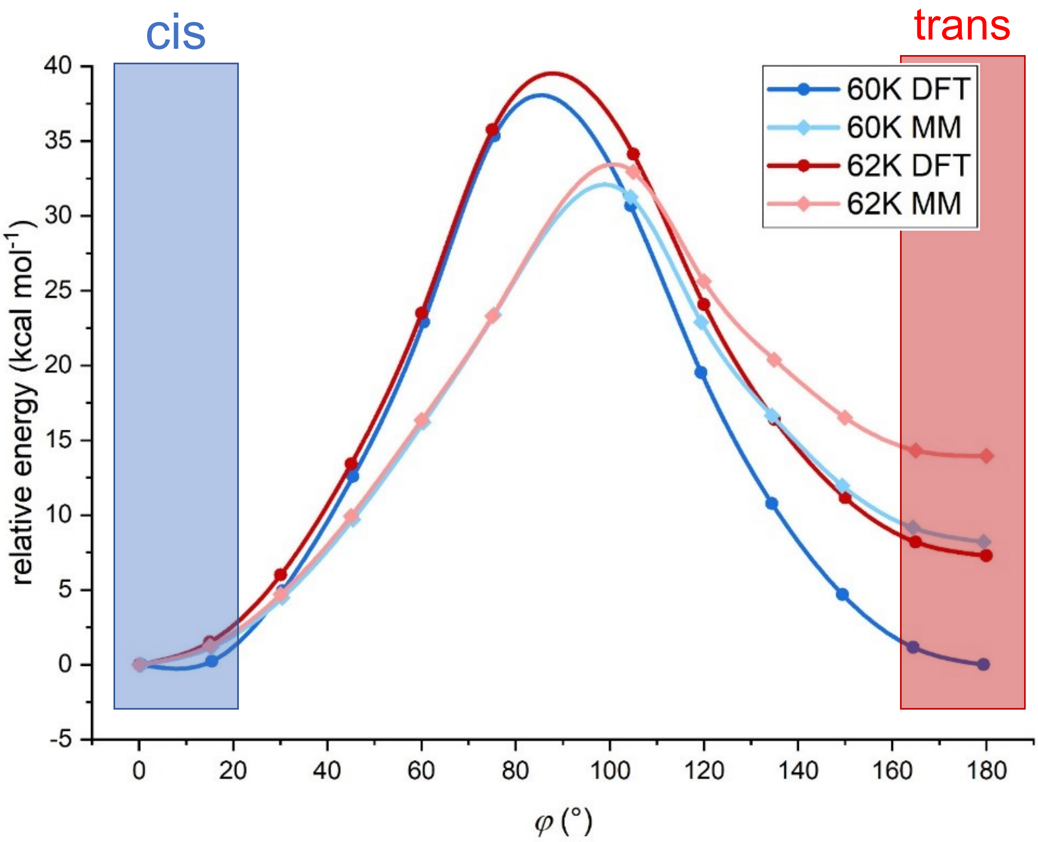


**Figure B.11:** Relative energy values for 60K and 62K calculated at DFT (dark blue and dark red circles, respectively) and MM (light blue and pink squares, respectively) levels of theory for the cis-trans interconversion based on DFT optimized geometries along $\phi$ . The dihedral angle ($\phi$ ) is defined between atoms N6-N9-C14-C16 for 60K and N1-N3-C25-C26 for 62K. The rotational barriers are lower in the force field (MM) than calculated at the DFT level. Note that the MM relative energy of the trans isomer with respect to the cis for 62K is about 10 kcal/mol higher than for 60K, contributing to the different behavior observed between the two similar ligands.

## B.8 Gradient Boosting Decision Trees Results

We used GBDT as an alternative approach to the MLP. The model was trained using the same amount of data fed for the MLP. We compared the results obtained from the MLTSA against the feature importances given by the GBDT. Overall, features resulting important from the MLTSA are also present in the GBDT, however, depending on the system we analyzed, additional important features were also detected from the GBDT's important features. This suggests that the more complex non-linear behavior might lead to different performances for GBDT and the MLP as compared to the analytical model system.

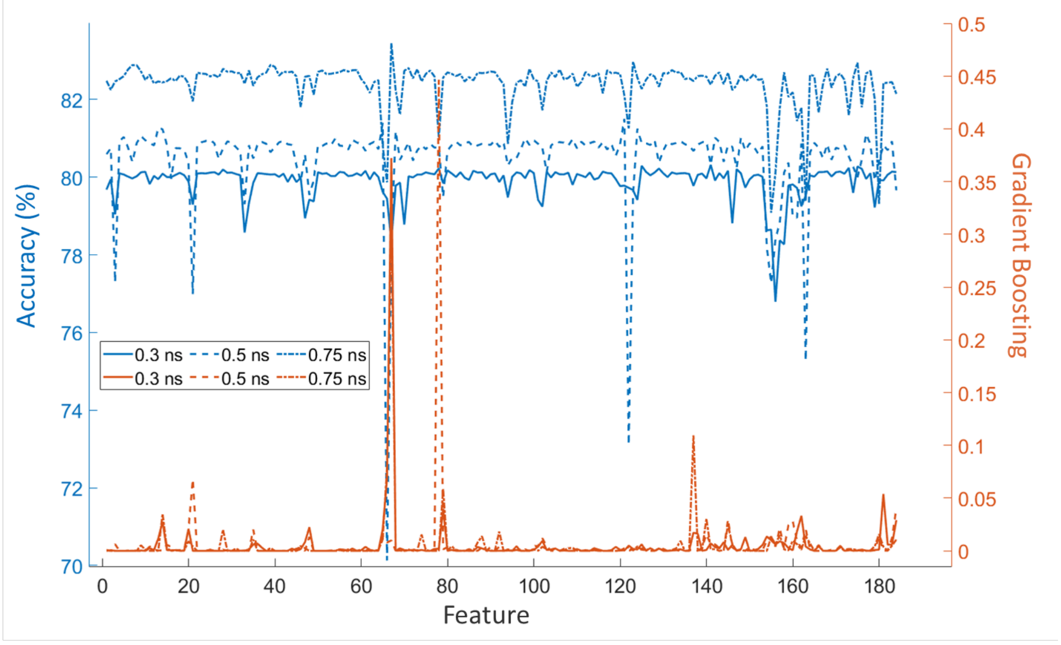


**Figure B.12:** Comparison between GBDT feature importance (orange) and MLTSA accuracy drops (blue) at different times for the three systems for ligand 18K.

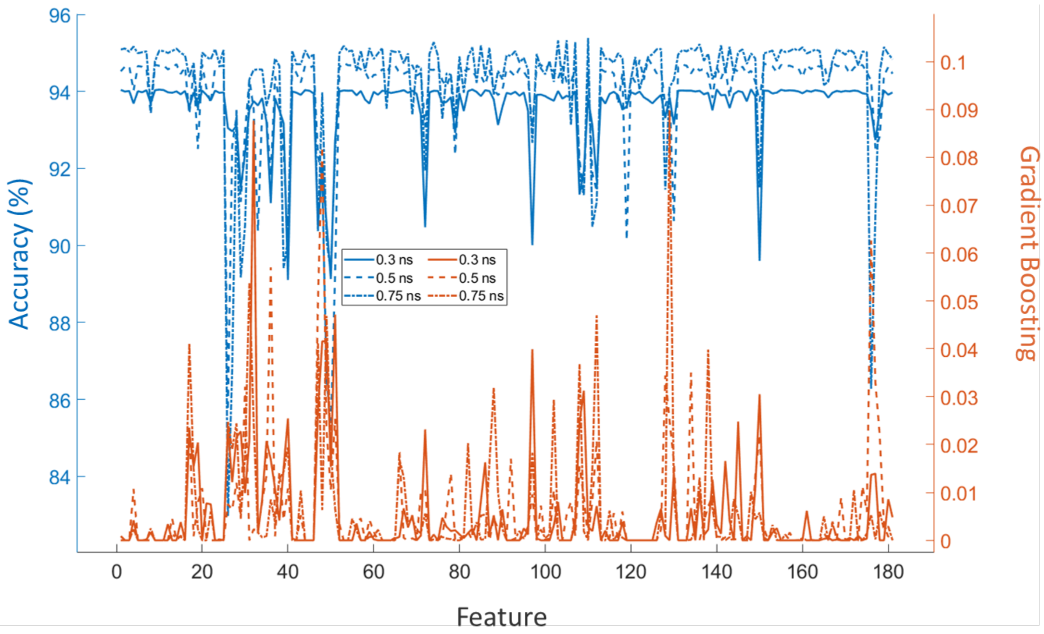


**Figure B.13:** Comparison between GBDT feature importance (orange) and MLTSA accuracy drops (blue) at different times for the three systems for ligand 62K.

## B.9 Additional Resources

### B.9.1 Animated trajectories

Animated GIF files showing the string trajectories for the three systems (3sw4, 4fku and 4fkw) of all replicas are available at the GitHub repository:

- https://github.com/pedrojuanbj/MLTSA-V1

### B.9.2 Software package

A Python package of the analytical MLTSA example and corresponding Python code is accessible under the Python Package Index (PyPi) database:

- https://pypi.org/project/MLTSA/

# Appendix C

# hMR3 Supplementary Information

## C.1 Unbinding

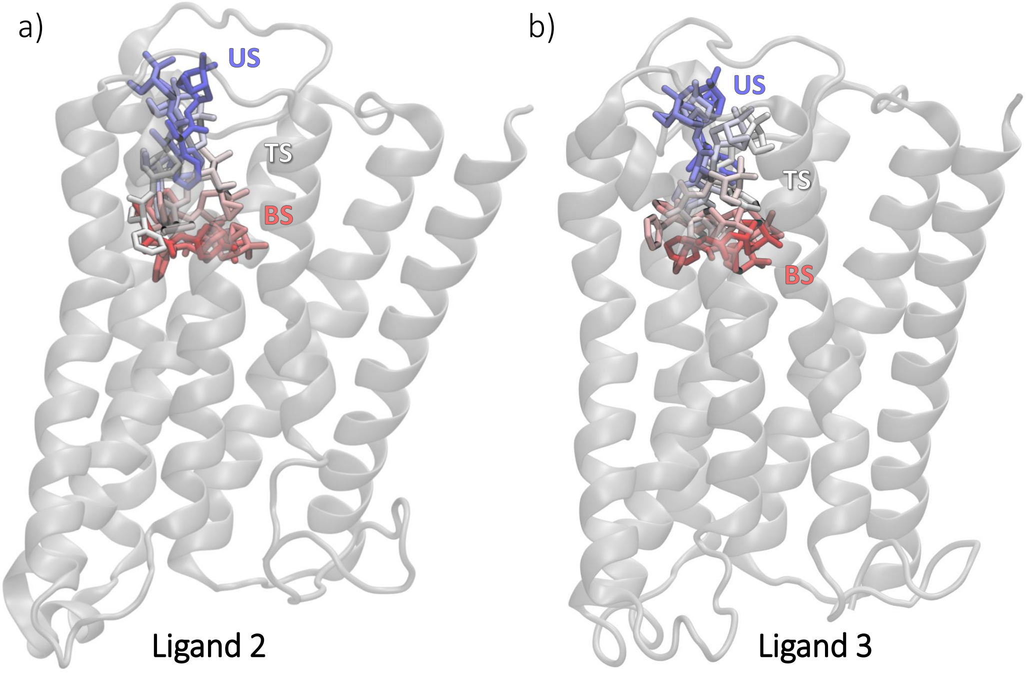


**Figure C.1:** Overlay of structures found throughout the unbinding path of ligands **2** and **3** starting from hMR3's orthosteric binding site. Ligands, represented as sticks, start at the BS (red) and move through the TS (white) to reach the US (blue) on the extracellular vestibule of the receptor (grey, in cartoon).

## C.2 Dataset Creation

All datasets containing interatomic distances were created using mdtraj, the outcomes of the simulations were assigned using the original biasing CV value from unbinding. If after 5ns the distance was under XÅ it was assigned IN, otherwise if bigger than YÅ it was assigned OUT.

### C.2.1 XYZ-PCA Dataset

To assess the importance of protein-protein distances, an additional dataset was created using the XYZ coordinates of all protein atoms (except hydrogens). The number of protein atoms for the system is  2247, which yields around 6741 coordinates. To be able to reduce this number, a PCA projection was applied to the data. In Fig. C.2, the resulting explained variance ratio can be found for the different PCA components resulted from the calculation.

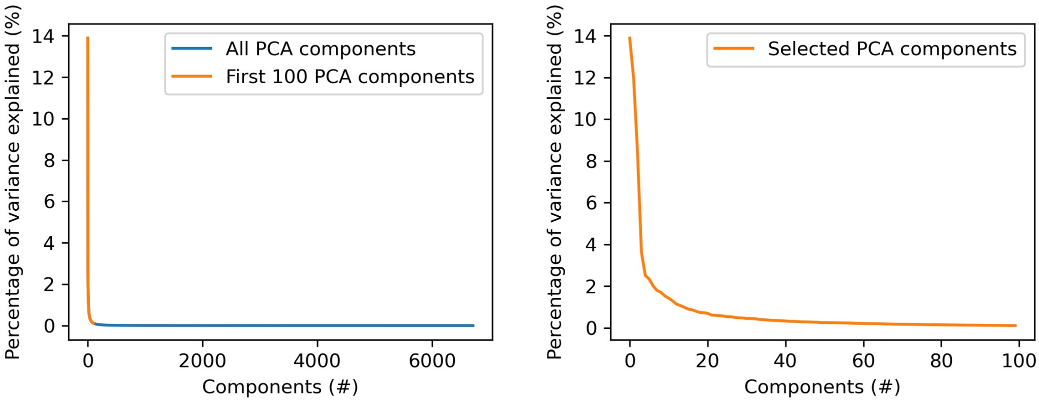


**Figure C.2:** Explained variance ratio for the fitted PCA from the XYZ-PCA set. Left: Values of all PCA components projected. Right: Values more in detail for the selected components which are the first 100 components. The first 100 were selected because of the rapid decay to 0% explained variance and make sure no information was lost.

At around 100 components, the explained variance ratio dropped close to 0. I decided to use these 100 new components as features for training.

### C.2.2 3Å and 6Å Datasets

The dataset was created by selecting all protein atoms within 3Å of any of the ligand atoms in the starting TS structure and defining all interatomic distances between ligand atoms and protein heavy atoms.

### C.2.3 3Å+Loop Datasets

To assess the importance of particular residues outside the pocket, datasets with all interatomic distances from ranging from I222 to T231 were added to the previous 3Å set, including the ECL2-TM5 junction region. (ECL2-TM5 dataset). To validate the relevance of the ECL2-TM5 loop, an additional dataset with residues I501 to C516 around ECL3 were also tested. However, no distances from this loop were identified as important either by MLP or GBDT models.

## C.3 ML Models

### C.3.1 MLP

The MLP model was setup using the MLPClassifier from Scikit-Learn [59], using 3 layers (input, hidden, output) with as many input nodes as features used in the first layer, 100 hidden nodes in the second layer using the ReLU [60] activation function and 1 output layer made of 1 node with a logistic activation function (0-IN,1-OUT). The model was optimized using the Adam solver [61], with a learning rate of 0.001, iterating over data until convergence or upon reaching the maximum number of iterations (500 epochs). Convergence is determined by the tolerance and the number of epochs with no change in loss. When having 10 consecutive epochs with less than 0.0001 improvement on the loss, the training stops, and it is considered that the model has reached convergence. All datasets used the same parameters for training.

### C.3.2 GBDT

The GBDT model used was setup using the GradientBoostingClassifier implementation from Scikit-Learn [59], I trained 500 decision stumps as weak learners minimizing a logistic loss function, with a learning rate of 0.1. The Friedman Mean Squared Error (MSE) was used for to assess the split of the internal nodes, using a minimum of 2 samples to split, and 1 sample required at the leaf nodes. The maximum depth of the individual estimators was 3, without a limit on the maximum number of features to consider for the best split or the leaf nodes. Training was done using a validation fraction of 0.1 internally.

## C.4 Trainings

To find the optimum time-range from the downhill trajectories to train the ML models, a training at different times was performed. Using a time window of 0.05 ns at a time, I tested from 0.05 ns to 0.5 ns. It was found that between 0.05 ns and 0.1 ns would be as early as possible without sacrificing reasonable accuracy.

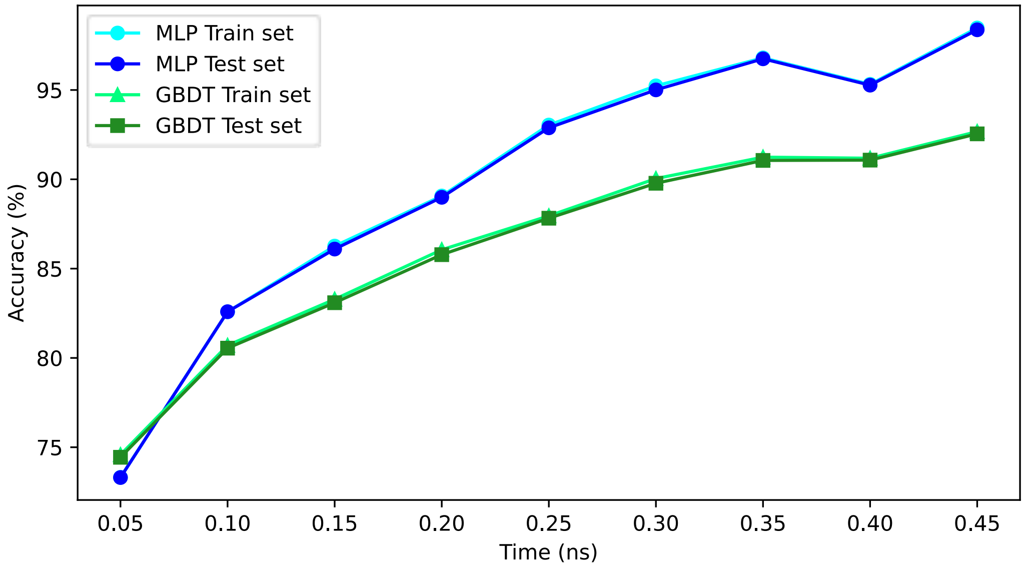


**Figure C.3:** MLP (Blue) and GBDT (Green) accuracy for the test and train sets at different times throughout the downhill trajectories using early near-TS data until 0.5 ns. Note the dataset used is the 3Å set.

**Table C.1:** Train (70% data) and test (30% data) accuracy for the MLP and GBDT models on the different datasets.

| Dataset | | MLP (%) | GBDT (%) |
|---|---|---|---|
| ***3Å set*** | Train | 77.80 ± 0.32 | 76.15 ± 0.75 |
| | Test | 76.34 ± 0.52 | 75.89 ± 0.74 |
| ***6Å set*** | Train | 75.18 ± 0.23 | 76.64 ± 0.12 |
| | Test | 75.71 ± 0.25 | 76.31 ± 0.11 |
| ***3Å+ECL2 set*** | Train | 76.82 ± 0.95 | 80.31 ± 0.15 |
| | Test | 76.27 ± 1.14 | 80.14 ± 0.12 |
| ***XYZ-PCA set*** | Train | 100 | 93.42 ± 0.82 |
| | Test | 100 | 93.25 ± 0.85 |
| ***Allres set*** | Train | 77.85 ± 0.01 | 79.27 ± 0.01 |
| | Test | 77.71 ± 0.01 | 79.01 ± 0.01 |
| ***Allres+wat set*** | Train | 81.27 ± 0.04 | 80.22 ± 0.03 |
| | Test | 81.05 ± 0.04 | 79.82 ± 0.03 |

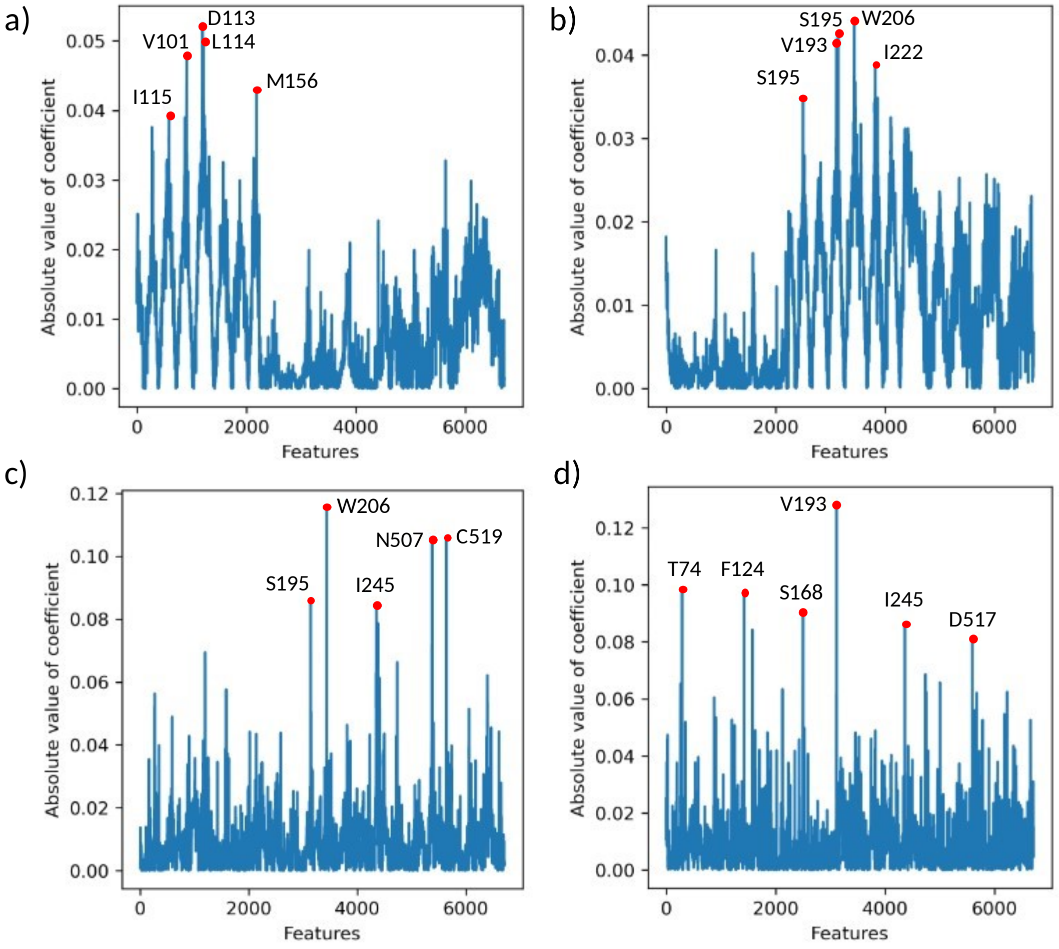


**Figure C.4:** Atomic Cartesian coordinates-feature contributions to the top PCA components 1 (b), 2 (b), 23 (c), and 59 (d) selected.

## C.5 PCA

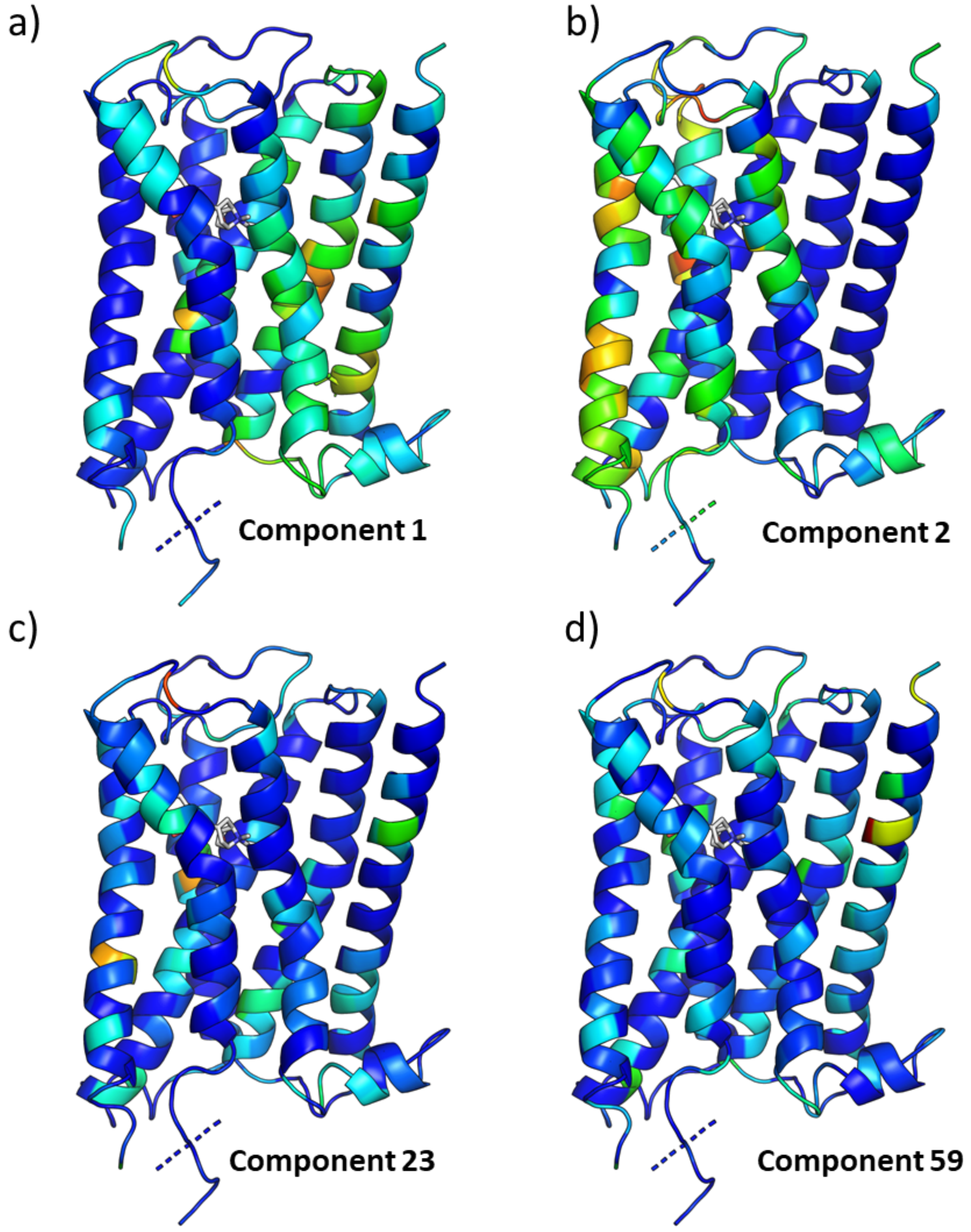


**Figure C.5:** Protein representation of the average atomic contributions to the PCA components 1 (b), 2 (b), 23 (c), and 59 (d) colored by the R-factor (blue to red: 0 to 1). Ligand in white.

## C.6 Additional analysis

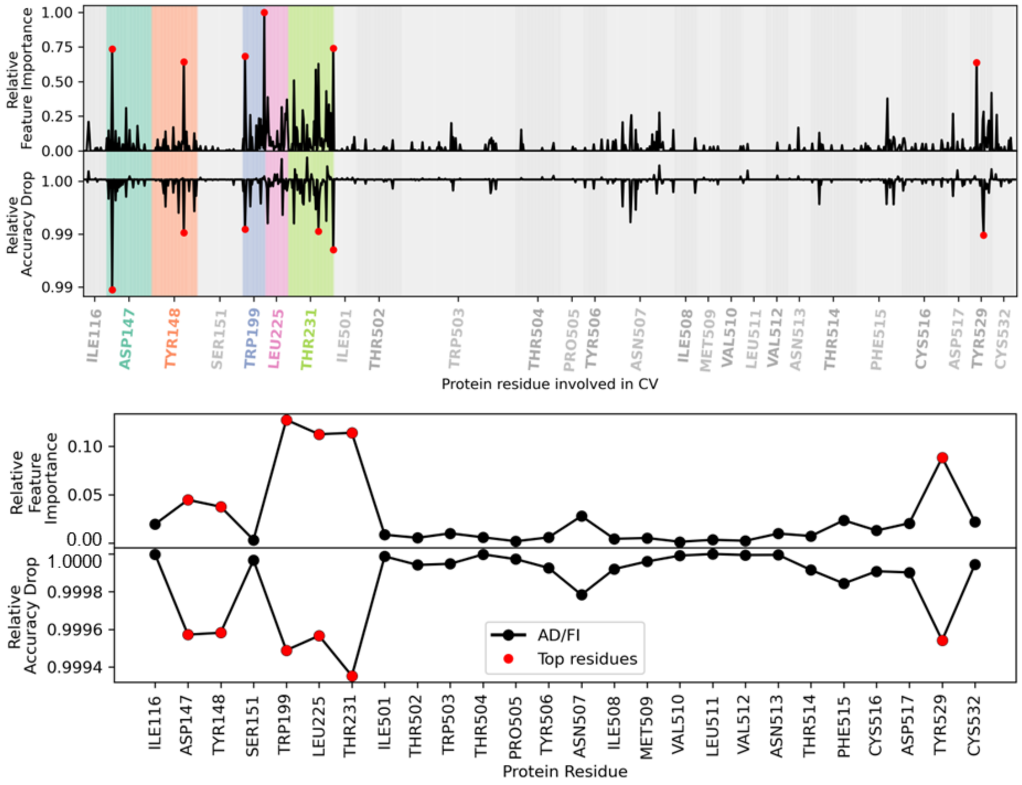


**Figure C.6:** Top: RAD and RFI for the 3Å adding interatomic distances from an alternate loop ECL3. Residues newly included range from I501 to C516. Top distances marked in red, top residues highlighted in color. Bottom: Average per residue RAD and RFI for the 3Å+ECL3. Top residues marked in red.

## C.7 Additional Resources

A GitHub repository with the multi-PDB and GIF unbinding trajectories through the string windows and the PDB TS structures can be found at the project's GitHub.

# Appendix D

# ACHREMD Supplementary Information

## D.1 Umbrella Sampling EDTA-Mg

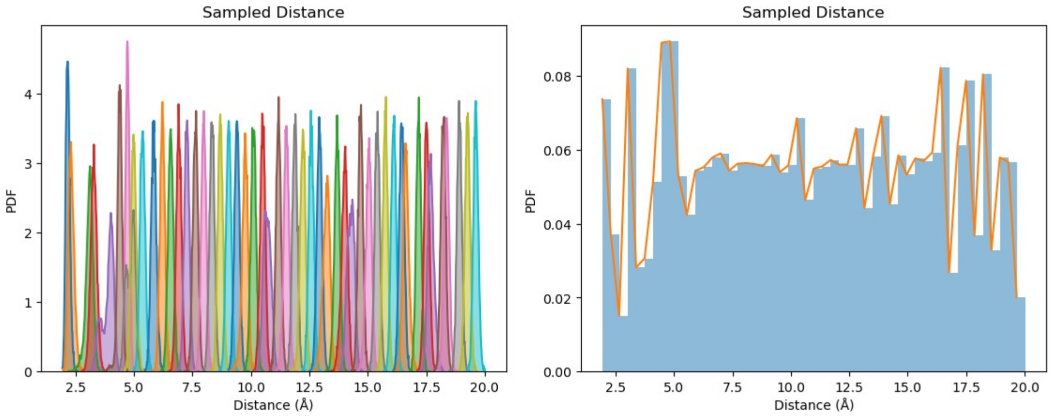


**Figure D.1:** Left: Histograms of each of the 50 windows during the US run for the EDTA-Mg system. Right: the combined histogram of all of the sampled distances from all windows.

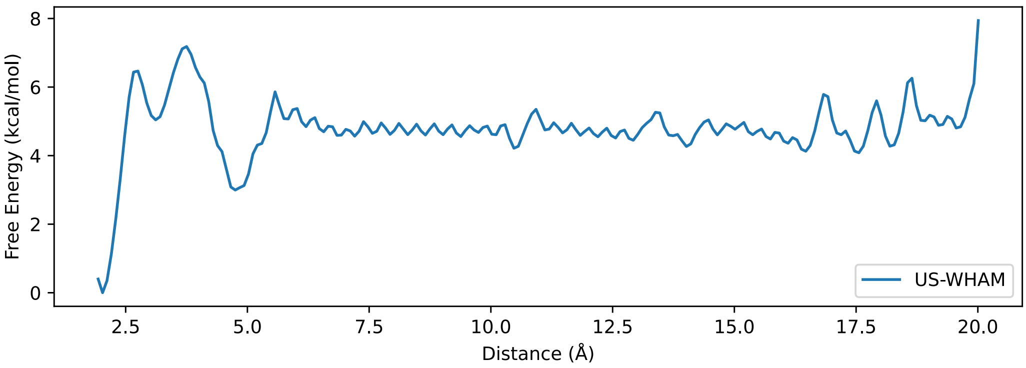


**Figure D.2:** Full free energy profile recovered using WHAM for the umbrella sampling run on the EDTA-Mg system.

## Appendix E

# Colophon

This document was set in the Times Roman typeface using LaTeX and BibTeX, composed with a text editor.

Molecular representations were done with Pymol and VMD, Microsoft PowerPoint was used for the complex figures and illustrations. Graphs Were created with matplotlib and seaborn.

# Bibliography

(1) Badaoui, M.; Buigues, P. J.; Berta, D.; Mandana, G. M.; Gu, H.; Földes, T.; Dickson, C. J.; Hornak, V.; Kato, M.; Molteni, C.; Parsons, S.; Rosta, E. *Journal of Chemical Theory and Computation* **2022**, *18*, 2543–2555.

(2) Buigues, P. J.; Gehrke, S.; Badaoui, M.; Mandana, G.; Qi, T.; Bottegoni, G.; Rosta, E. *bioRxiv* **2023**, 2023.01.03.522558.

(3) Huggins, D. J.; Biggin, P. C.; Dämgen, M. A.; Essex, J. W.; Harris, S. A.; Henchman, R. H.; Khalid, S.; Kuzmanic, A.; Laughton, C. A.; Michel, J.; Mulholland, A. J.; Rosta, E.; Sansom, M. S. P.; van der Kamp, M. W. *Wiley Interdisciplinary Reviews: Computational Molecular Science* **2019**, *9*, e1393.

(4) Bock, L. V.; Gabrielli, S.; Koláˇ, M. H. **2023**, 1–30.

(5) Schreiber, G.; Haran, G.; Zhou, H. X. *Chemical Reviews* **2009**, *109*, 839–860.

(6) Singh, A. N.; Ramadan, K.; Singh, S., *Experimental methods to study the kinetics of protein-protein interactions*; Elsevier Inc.: 2022, pp 115–124.

(7) Wu, Y.; Zeng, L.; Zhao, S. *Biomolecules* **2021**, *11*, 1–19.

(8) Chan, H. C.; Filipek, S.; Yuan, S. *Scientific Reports* **2016**, *6*, 1–11.

(9) Copeland, R. A.; Pompliano, D. L.; Meek, T. D. *Nature Reviews Drug Discovery* **2006**, *5*, 730–739.

(10) Copeland, R. A. The drug-target residence time model: A 10-year retrospective, 2016.

(11) Hollingsworth, S. A.; Dror, R. O. *Neuron* **2018**, *99*, 1129–1143.

(12) Tiwary, P.; Limongelli, V.; Salvalaglio, M.; Parrinello, M. *Proceedings of the National Academy of Sciences of the United States of America* **2015**, *112*, E386–E391.

(13) Grossfield, A.; Patrone, P. N.; Roe, D. R.; Schultz, A. J.; Siderius, D.; Zuckerman, D. M. *Living Journal of Computational Molecular Science* **2019**, *1*, DOI: `10.33011/livecoms.1.1.5067`.

(14) Bernardi, R. C.; Melo, M. C.; Schulten, K. *Biochimica et Biophysica Acta - General Subjects* **2015**, *1850*, 872–877.

(15) Hénin, J.; Lelièvre, T.; Shirts, M. R.; Valsson, O.; Delemotte, L. *Living Journal of Computational Molecular Science* **2022**, *4*, 1583–1583.

(16) Alzubaidi, L.; Zhang, J.; Humaidi, A. J.; Al-Dujaili, A.; Duan, Y.; Al-Shamma, O.; Santamaría, J.; Fadhel, M. A.; Al-Amidie, M.; Farhan, L., *Review of deep learning: concepts, CNN architectures, challenges, applications, future directions*; 1; Springer International Publishing: 2021; Vol. 8.

(17) Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. *Readings in Cognitive Science: A Perspective from Psychology and Artificial Intelligence* **1986**, 399–421.

(18) Sherstinsky, A. *Physica D: Nonlinear Phenomena* **2020**, *404*, 1–43.

(19) Wang, H.; Huang, N.; Dangerfield, T.; Johnson, K. A.; Gao, J.; Elber, R. *The Journal of Physical Chemistry B* **2020**, *124*, 4270–4283.

(20) Vaswani, A.; Brain, G.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Polosukhin, I.

(21) Noé, F.; Tkatchenko, A.; Müller, K.-R.; Clementi, C. *Annual Review of Physical Chemistry* **2020**, DOI: `10.1146/annurev-physchem-042018`.

(22) Behler, J.; Parrinello, M. *Physical Review Letters* **2007**, *98*, 146401.

(23) Wang, J.; Olsson, S.; Wehmeyer, C.; Pérez, A.; Charron, N. E.; De Fabritiis, G.; Noé, F.; Clementi, C. *ACS Central Science* **2019**, *5*, 755–767.

(24) Krishnamoorthy, A.; Rajak, P.; Hong, S.; Nomura, K. I.; Tiwari, S.; Kalia, R. K.; Nakano, A.; Vashishta, P. In *Journal of Physics: Conference Series*, IOP Publishing: 2020; Vol. 1461, p 012182.

(25) Tribello, G. A.; Gasparotto, P. *Frontiers in Molecular Biosciences* **2019**, *6*, 46.

(26) Jones, D.; Kim, H.; Zhang, X.; Zemla, A.; Stevenson, G.; Bennett, W. F.; Kirshner, D.; Wong, S. E.; Lightstone, F. C.; Allen, J. E. *Journal of Chemical Information and Modeling* **2021**, *61*, 1583–1592.

(27) Wallach, I.; Dzamba, M.; Heifets, A. **2015**.

(28) Razavian, N. S.; Kamisetty, H.; Langmead, C. J. *Series on Advances in Bioinformatics and Computational Biology* **2012**, *13*, 1–13.

(29) Smith, J. S.; Isayev, O.; Roitberg, A. E. *Chemical Science* **2017**, *8*, 3192–3203.

(30) Schütt, K. T.; Arbabzadah, F.; Chmiela, S.; Müller, K. R.; Tkatchenko, A. *Nature Communications 2017 8:1* **2017**, *8*, 1–8.

(31) Schütt, K. T.; Kessel, P.; Gastegger, M.; Nicoli, K. A.; Tkatchenko, A.; Müller, K. R. *Journal of chemical theory and computation* **2019**, *15*, 448–455.

(32) Mardt, A.; Pasquali, L.; Wu, H.; Noé, F. *Nature Communications* **2018**, *9*, 1–11.

(33) Fleetwood, O.; Kasimova, M. A.; Westerlund, A. M.; Delemotte, L. *Biophysical Journal* **2020**, *118*, 765–780.

(34) Kaptan, S.; Vattulainen, I. *https://doi.org/10.1080/23746149.2021.2006(1) Kaptan, S.; Vattulainen, I. Machine Learning in the Analysis of Biomolecular Simulations. https://doi.org/10.1080/23746149.2021.2006080 2022, 7 (1). https://doi.org/10.1080/23746149.2021.2006080.080* **2022**, *7*, DOI: `10.1080/23746149.2021.2006080`.

(35) Bonomi, M.; Heller, G. T.; Camilloni, C.; Vendruscolo, M. *Current Opinion in Structural Biology* **2017**, *42*, 106–116.

(36) Ma, J.; Sheridan, R. P.; Liaw, A.; Dahl, G. E.; Svetnik, V. *Journal of Chemical Information and Modeling* **2015**, *55*, 263–274.

(37) Tribello, G. A.; Gasparotto, P. *Frontiers in Molecular Biosciences* **2019**, *6*, 1–11.

(38) Tribello, G. A.; Ceriotti, M.; Parrinello, M. *Proceedings of the National Academy of Sciences of the United States of America* **2012**, *109*, 5196–5201.

(39) Berta, D.; Buigues, P. J.; Badaoui, M.; Rosta, E. *Current Opinion in Structural Biology* **2020**, *61*, 198–206.

(40) Jambrina, P. G.; Rauch, N.; Pilkington, R.; Rybakova, K.; Nguyen, L. K.; Kholodenko, B. N.; Buchete, N. V.; Kolch, W.; Rosta, E. *Angewandte Chemie - International Edition* **2016**, *55*, 983–986.

(41) Rosta, E.; Nowotny, M.; Yang, W.; Hummer, G. *Journal of the American Chemical Society* **2011**, *133*, 8934–8941.

(42) Ahlstrand, E.; Hermansson, K.; Friedman, R. *Journal of Physical Chemistry A* **2017**, *121*, 2643–2654.

(43) Li, P.; Merz, K. M. *Chemical Reviews* **2017**, *117*, 1564–1686.

(44) Wu, J. C.; Piquemal, J. P.; Chaudret, R.; Reinhardt, P.; Ren, P. *Journal of Chemical Theory and Computation* **2010**, *6*, 2059–2070.

(45) Xu, M.; Zhu, T.; Zhang, J. Z. *Frontiers in Chemistry* **2021**, *9*, 1–10.

(46) Lopes, P. E.; Guvench, O.; Mackerell, A. D. *Methods in molecular biology (Clifton, N.J.)* **2015**, *1215*, 47.

(47) Braun, E.; Gilmer, J.; Mayes, H. B.; Mobley, D. L.; Monroe, J. I.; Prasad, S.; Zuckerman, D. M. *Living Journal of Computational Molecular Science* **2019**, *1*, 1–28.

(48) E, W.; Ren, W.; Vanden-Eijnden, E.; Weinan, E; Ren, W.; Vanden-Eijnden, E. *J. Phys. Chem. B* **2005**, *109*, 6688–6693.

(49) Ovchinnikov, V.; Karplus, M.; Vanden-Eijnden, E. *The Journal of Chemical Physics* **2011**, *134*, 85103.

(50) Kumar, S.; Rosenberg, J. M.; Bouzida, D.; Swendsen, R. H.; Kollman, P. A. *Journal of Computational Chemistry* **1992**, *13*, 1011–1021.

(51) Mcgibbon, R. T.; Beauchamp, K. A.; Harrigan, M. P.; Klein, C.; Swails, J. M.; Herná Ndez, C. X.; Schwantes, C. R.; Wang, L.-P.; Lane, T. J.; Pande, V. S. **2015**, DOI: `10.1016/j.bpj.2015.08.015`.

(52) Altmann, A.; Toloşi, L.; Sander, O.; Lengauer, T. *Bioinformatics* **2010**, *26*, 1340–1347.

(53) Verikas, A.; Bacauskiene, M. *Pattern Recognition Letters* **2002**, *23*, 1323–1335.

(54) Sugita, Y.; Okamoto, Y. *Chemical Physics Letters* **1999**, *314*, 141–151.

(55) Abrams, C.; Bussi, G. *Entropy* **2013**, *16*, 163–199.

(56) Meli, M.; Colombo, G. *International Journal of Molecular Sciences* **2013**, *14*, 12157–12169.

(57) McSkimming, D. I.; Rasheed, K.; Kannan, N. *BMC Bioinformatics* **2017**, *18*, 1–15.

(58) Wang, D.; Wang, Y.; Evans, L.; Tiwary, P. **2022**.

(59) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, *Journal of Machine Learning Research* **2011**, *12*, 2825–2830.

(60) Fukushima, K. *Biological Cybernetics* **1975**, *20*, 121–136.

(61) Kingma, D. P.; Ba, J. L. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings* **2015**, 1–15.

(62) Copeland, R. A., *Evaluation of Enzyme Inhibitors in Drug Discovery: A Guide for Medicinal Chemists and Pharmacologists: Second Edition*; John Wiley and Sons: 2013, pp 1–538.

(63) Bernetti, M.; Masetti, M.; Recanatini, M.; Amaro, R. E.; Cavalli, A. *Journal of Chemical Theory and Computation* **2019**, *15*, 5689–5702.

(64) Lu, H.; Tonge, P. J. Drug-target residence time: Critical information for lead optimization, 2010.

(65) Bernetti, M.; Cavalli, A.; Mollica, L. Protein-ligand (un)binding kinetics as a new paradigm for drug discovery at the crossroad between experiments and modelling, 2017.

(66) Schuetz, D. A.; de Witte, W. E. A.; Wong, Y. C.; Knasmueller, B.; Richter, L.; Kokh, D. B.; Sadiq, S. K.; Bosma, R.; Nederpelt, I.; Heitman, L. H.; Segala, E.; Amaral, M.; Guo, D.; Andres, D.; Georgi, V.; Stoddart, L. A.; Hill, S.; Cooke, R. M.; De Graaf, C.; Leurs, R.; Frech, M.; Wade, R. C.; de Lange, E. C. M.; IJzerman, A. P.; Müller-Fahrnow, A.; Ecker, G. F. Kinetics for Drug Discovery: an industry-driven effort to target drug residence time, 2017.

(67) Darling, R. J.; Brault, P. A. Kinetic exclusion assay technology: Characterization of molecular interactions, 2004.

(68) Rose, R. H.; Briddon, S. J.; Hill, S. J. *British Journal of Pharmacology* **2012**, *165*, 1789–1800.

(69) Bruce, N. J.; Ganotra, G. K.; Kokh, D. B.; Sadiq, S. K.; Wade, R. C. *Current Opinion in Structural Biology* **2018**, *49*, 1–10.

(70) Wolf, S.; Lickert, B.; Bray, S.; Stock, G. *Nature Communications* **2020**, *11*, DOI: `10.1038/s41467-020-16655-1`.

(71) Dahl, G.; Akerud, T. Pharmacokinetics and the drug-target residence time concept, 2013.

(72) Lotz, S. D.; Dickson, A. *Journal of the American Chemical Society* **2018**, *140*, 618–628.

(73) Nunes-Alves, A.; Kokh, D. B.; Wade, R. C. Recent progress in molecular simulation methods for drug binding kinetics, 2020.

(74) Decherchi, S.; Cavalli, A. Thermodynamics and Kinetics of Drug-Target Binding by Molecular Simulation, 2020.

(75) Jorgensen, W. L.; Ravimohan, C *Citation: The Journal of Chemical Physics* **1985**, *83*, 3050.

(76) Jorgensen, W. L.; Thomas, L. L. *Journal of Chemical Theory and Computation* **2008**, *4*, 869–876.

(77) Laio, A.; Parrinello, M. *Proceedings of the National Academy of Sciences of the United States of America* **2002**, *99*, 12562–12566.

(78) Hamelberg, D.; Mongan, J.; McCammon, J. A. *Journal of Chemical Physics* **2004**, *120*, 11919–11929.

(79) Izrailev, S.; Stepaniants, S.; Isralewitz, B.; Kosztin, D.; Lu, H.; Molnar, F.; Wriggers, W.; Schulten, K. In Springer, Berlin, Heidelberg: 1999, pp 39–65.

(80) Faradjian, A. K.; Elber, R. *Journal of Chemical Physics* **2004**, *120*, 10880–10889.

(81) Hovan, L.; Comitani, F.; Gervasio, F. L. *Journal of Chemical Theory and Computation* **2019**, *15*, 25–32.

(82) Rohrdanz, M. A.; Zheng, W.; Maggioni, M.; Clementi, C. *Journal of Chemical Physics* **2011**, *134*, 124116.

(83) Casasnovas, R.; Limongelli, V.; Tiwary, P.; Carloni, P.; Parrinello, M. *Journal of the American Chemical Society* **2017**, *139*, 4780–4788.

(84) Haldar, S.; Comitani, F.; Saladino, G.; Woods, C.; Van Der Kamp, M. W.; Mulholland, A. J.; Gervasio, F. L. *Journal of Chemical Theory and Computation* **2018**, *14*, 6093–6101.

(85) Capelli, A. M.; Costantino, G. *Journal of Chemical Information and Modeling* **2014**, *54*, 3124–3136.

(86) Rydzewski, J.; Valsson, O. *Journal of Chemical Physics* **2019**, *150*, 221101.

(87) Carter, E. A.; Ciccotti, G.; Hynes, J. T.; Kapral, R. *Chemical Physics Letters* **1989**, *156*, 472–477.

(88) Jung, H.; Covino, R.; Hummer, G. *Artificial Intelligence Assists Discovery of Reaction Coordinates and Mechanisms from Molecular Dynamics Simulations*; tech. rep.; 2019.

(89) Noé, F.; Tkatchenko, A.; Müller, K.-R.; Clementi, C. *Annual Review of Physical Chemistry* **2019**, *71*, 361–390.

(90) Glielmo, A.; Husic, B. E.; Rodriguez, A.; Clementi, C.; Noé, F.; Laio, A. Unsupervised Learning Methods for Molecular Simulation Data, 2021.

(91) Burger, H. C.; Schuler, C. J.; Harmeling, S. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2012, pp 2392–2399.

(92) Rao, H.; Shi, X.; Rodrigue, A. K.; Feng, J.; Xia, Y.; Elhoseny, M.; Yuan, X.; Gu, L. *Applied Soft Computing* **2019**, *74*, 634–642.

(93) Hinton, G. E.; Salakhutdinov, R. R. *Science* **2006**, *313*, 504–507.

(94) Dunbar, J. B.; Smith, R. D.; Damm-Ganamet, K. L.; Ahmed, A.; Esposito, E. X.; Delproposto, J.; Chinnaswamy, K.; Kang, Y. N.; Kubish, G.; Gestwicki, J. E.; Stuckey, J. A.; Carlson, H. A. *Journal of Chemical Information and Modeling* **2013**, *53*, 1842–1852.

(95) Malumbres, M.; Barbacid, M. Cell cycle, CDKs and cancer: A changing paradigm, 2009.

(96) Otto, T.; Sicinski, P. Cell cycle proteins as promising targets in cancer therapy, 2017.

(97) Tadesse, S.; Caldon, E. C.; Tilley, W.; Wang, S. Cyclin-Dependent Kinase 2 Inhibitors in Cancer Therapy: An Update, 2019.

(98) Jessen, B. A.; Lee, L.; Koudriakova, T.; Haines, M.; Lundgren, K.; Price, S.; Nonomiya, J.; Lewis, C.; Stevens, G. J. *Journal of Applied Toxicology* **2007**, *27*, 133–142.

(99) Parry, D.; Guzi, T.; Shanahan, F.; Davis, N.; Prabhavalkar, D.; Wiswell, D.; Seghezzi, W.; Paruch, K.; Dwyer, M. P.; Doll, R.; Nomeir, A.; Windsor, W.; Fischmann, T.; Wang, Y.; Oft, M.; Chen, T.; Kirschmeier, P.; Lees, E. M. *Molecular Cancer Therapeutics* **2010**, *9*, 2344–2353.

(100) Ayaz, P.; Andres, D.; Kwiatkowski, D. A.; Kolbe, C. C.; Lienau, P.; Siemeister, G.; L??cking, U.; Stegmann, C. M. *ACS Chemical Biology* **2016**, *11*, 1710–1719.

(101) Caporali, S.; Alvino, E.; Starace, G.; Ciomei, M.; Brasca, M. G.; Levati, L.; Garbin, A.; Castiglia, D.; Covaciu, C.; Bonmassar, E.; D'Atri, S. *Pharmacological Research* **2010**, *61*, 437–448.

(102) Wang, L.; Lu, D.; Wang, Y.; Xu, X.; Zhong, P.; Yang, Z. *Journal of Enzyme Inhibition and Medicinal Chemistry* **2023**, *38*, 84–99.

(103) Phillips, J. C.; Braun, R.; Wang, W.; Gumbart, J.; Tajkhorshid, E.; Villa, E.; Chipot, C.; Skeel, R. D.; Kalé, L.; Schulten, K. Scalable molecular dynamics with NAMD, 2005.

(104) Maier, J. A.; Martinez, C.; Kasavajhala, K.; Wickstrom, L.; Hauser, K. E.; Simmerling, C. *Journal of Chemical Theory and Computation* **2015**, *11*, 3696–3713.

(105) Wang, J.; Wolf, R. M.; Caldwell, J. W.; Kollman, P. A.; Case, D. A. *Journal of Computational Chemistry* **2004**, *25*, 1157–1174.

(106) Lans, I.; Medina, M.; Rosta, E.; Hummer, G.; Garcia-Viloca, M.; Lluch, J. M.; González-Lafont, *Journal of the American Chemical Society* **2012**, *134*, 20544–20553.

(107) Nair, V.; Hinton, G. E. *Rectified Linear Units Improve Restricted Boltzmann Machines*; tech. rep.

(108) Suardíaz, R.; Jambrina, P. G.; Masgrau, L.; González-Lafont, Rosta, E.; Lluch, J. M. *Journal of Chemical Theory and Computation* **2016**, *12*, 2079–2090.

(109) Rosta, E.; Woodcock, H. L.; Brooks, B. R.; Hummer, G. *Journal of Computational Chemistry* **2009**, *30*, 1634–1641.

(110) Li, Y.; Zhang, J.; Gao, W.; Zhang, L.; Pan, Y.; Zhang, S.; Wang, Y. Insights on structural characteristics and ligand binding mechanisms of CDK2, 2015.

(111) Patel, J. S.; Berteotti, A.; Ronsisvalle, S.; Rocchia, W.; Cavalli, A. *Journal of Chemical Information and Modeling* **2014**, *54*, 470–480.

(112) Wess, J. *Critical Reviews™ in Neurobiology* **1996**, *10*, 69–99.

(113) Heilker, R.; Wolff, M.; Tautermann, C. S.; Bieler, M. *Drug Discovery Today* **2009**, *14*, 231–240.

(114) Bonner, T. I. *Trends Pharmacol. Sci.* **1989**, 11–15.

(115) Venkatakrishnan, A. J.; Deupi, X.; Lebon, G.; Tate, C. G.; Schertler, G. F.; Babu, M. M. *Nature* **2013**, *494*, 185–194.

(116) Fredriksson, R.; Lagerström, M. C.; Lundin, L.-G.; Schiöth, H. B. *Molecular Pharmacology* **2003**, *63*, 1256–1272.

(117) Jakubík, J.; Bačáková, L.; El-Fakahany, E. E.; Tuček, S. *FEBS Letters* **1995**, *377*, 275–279.

(118) Casarosa, P.; Kiechle, T.; Sieger, P.; Pieper, M.; Gantner, F. *Journal of Pharmacology and Experimental Therapeutics* **2010**, *333*, 201–209.

(119) Cazzola, M; Matera, M. G. *British journal of pharmacology* **2008**, *155*, 291–299.

(120) Cazzola, M.; Page, C. P.; Calzetta, L.; Matera, M. G. *Pharmacological Reviews* **2012**, *64*, 450–504.

(121) Mak, G.; Hanania, N. A. *Current Opinion in Pharmacology* **2012**, *12*, 238–245.

(122) Casarosa, P.; Bouyssou, T.; Germeyer, S.; Schnapp, A.; Gantner, F.; Pieper, M. *Journal of Pharmacology and Experimental Therapeutics* **2009**, *330*, 660–668.

(123) Disse, B.; Speck, G. A.; Rominger, K. L.; Witek, T. J.; Hammer, R. *Life Sciences* **1999**, *64*, 457–464.

(124) Barnes, P. J. *Proceedings of the American Thoracic Society* **2004**, *1*, 345–351.

(125) Kruse, A. C.; Kobilka, B. K.; Gautam, D.; Sexton, P. M.; Christopoulos, A.; Wess, J. *Nature reviews. Drug discovery* **2014**, *13*, 549–560.

(126) Moulton, B. C.; Fryer, A. D. *British journal of pharmacology* **2011**, *163*, 44–52.

(127) Casarosa, P; Tautermann, C; Kiechle, T; Pieper, M. P.; Gantner, F Understanding the Mechanism of Long Duration of Action of Tiotropium: Insight into Its Interaction with the Human M3 Receptor. 2009.

(128) Tautermann, C. S.; Kiechle, T.; Seeliger, D.; Diehl, S.; Wex, E.; Banholzer, R.; Gantner, F.; Pieper, M. P.; Casarosa, P. *Journal of Medicinal Chemistry* **2013**, *56*, 8746–8756.

(129) Wolf, S.; Post, M.; Stock, G. **2022**.

(130) Wang, T.; Duan, Y. *Journal of Molecular Biology* **2009**, *392*, 1102–1115.

(131) Kokh, D. B.; Wade, R. C. *Journal of Chemical Theory and Computation* **2021**, *17*, 6610–6623.

(132) Dror, R. O.; Pan, A. C.; Arlow, D. H.; Borhani, D. W.; Maragakis, P.; Shan, Y.; Xu, H.; Shaw, D. E. *Proceedings of the National Academy of Sciences of the United States of America* **2011**, *108*, 13118–13123.

(133) Kruse, A. C.; Hu, J.; Pan, A. C.; Arlow, D. H.; Rosenbaum, D. M.; Rosemond, E.; Green, H. F.; Liu, T.; Chae, P. S.; Dror, R. O.; Shaw, D. E.; Weis, W. I.; Wess, J.; Kobilka, B. K. *Nature* **2012**, *482*, 552–556.

(134) Betz, R. M.; Dror, R. O. *Journal of Chemical Theory and Computation* **2019**, *15*, 2053–2063.

(135) Capelli, R.; Bochicchio, A.; Piccini, G.; Casasnovas, R.; Carloni, P.; Parrinello, M. *Journal of Chemical Theory and Computation* **2019**, *15*, 3354–3361.

(136) Capelli, R.; Lyu, W.; Bolnykh, V.; Meloni, S.; Olsen, J. M. H.; Rothlisberger, U.; Parrinello, M.; Carloni, P. *Journal of Physical Chemistry Letters* **2020**, *11*, 6373–6381.

(137) Rosta, E.; Nowotny, M.; Yang, W.; Hummer, G. *Journal of the American Chemical Society* **2011**, *133*, 8934–8941.

(138) Thorsen, T. S.; Matt, R.; Weis, W. I.; Kobilka, B. K. *Structure (London, England : 1993)* **2014**, *22*, 1657–1664.

(139) Jo, S.; Lim, J. B.; Klauda, J. B.; Im, W. *Biophysical journal* **2009**, *97*, 50–58.

(140) Wu, E. L.; Cheng, X.; Jo, S.; Rui, H.; Song, K. C.; Dávila-Contreras, E. M.; Qi, Y.; Lee, J.; Monje-Galvan, V.; Venable, R. M.; Klauda, J. B.; Im, W. *Journal of computational chemistry* **2014**, *35*, 1997–2004.

(141) Lee, J.; Patel, D. S.; Ståhle, J.; Park, S.-J.; Kern, N. R.; Kim, S.; Lee, J.; Cheng, X.; Valvano, M. A.; Holst, O.; Knirel, Y. A.; Qi, Y.; Jo, S.; Klauda, J. B.; Widmalm, G.; Im, W. *Journal of Chemical Theory and Computation* **2018**, *15*, 775–786.

(142) Brooks, B. R.; Brooks, C. L.; Mackerell, A. D.; Nilsson, L.; Petrella, R. J.; Roux, B.; Won, Y.; Archontis, G.; Bartels, C.; Boresch, S.; Caflisch, A.; Caves, L.; Cui, Q.; Dinner, A. R.; Feig, M.; Fischer, S.; Gao, J.; Hodoscek, M.; Im, W.; Kuczera, K.; Lazaridis, T.; Ma, J.; Ovchinnikov, V.; Paci, E.; Pastor, R. W.; Post, C. B.; Pu, J. Z.; Schaefer, M.; Tidor, B.; Venable, R. M.; Woodcock, H. L.; Wu, X.; Yang, W.; York, D. M.; Karplus, M.; Brooks III, B. R.; Mackerell, C. L. B.; Nilsson, A. D.; Petrella, L; Roux, R. J.; Won, B; Archontis, Y; Bartels, G; Boresch, C; Caflisch, S *J. Comput. Chem* **2009**, *30*, 1545–1614.

(143) Jo, S.; Kim, T.; Iyer, V. G.; Im, W. *Journal of Computational Chemistry* **2008**, *29*, 1859–1865.

(144) Jo, S.; Kim, T.; Im, W. *PloS one* **2007**, *2*, e880–e880.

(145) Jorgensen, W. L.; Chandrasekhar, J.; Madura, J. D.; Impey, R. W.; Klein, M. L. *J. Chem. Phys.* **1983**, *79*, 926–935.

(146) Garcia, M. L.; Giangiacomo, K. M.; Hanner, M.; Knaus, H.-G.; McManus, O. B.; Schmalhofer, W. A.; Kaczorowski, G. J. [14] Purification and functional reconstitution of high-conductance calcium-activated potassium channel from smooth muscle, 1999.

(147) Stephens, P. J.; Devlin, F. J.; Chabalowski, C. F.; Frisch, M. J. *The Journal of Physical Chemistry* **1994**, *98*, 11623–11627.

(148) Neese, F. *WIREs Computational Molecular Science* **2012**, *2*, 73–78.

(149) Neese, F. *WIREs Computational Molecular Science* **2018**, *8*, e1327–e1327.

(150) Neese, F.; Wennmohs, F.; Becker, U.; Riplinger, C. *The Journal of Chemical Physics* **2020**, *152*, 224108.

(151) Kim, S.; Lee, J.; Jo, S.; Brooks 3rd, C. L.; Lee, H. S.; Im, W. *Journal of computational chemistry* **2017**, *38*, 1879–1886.

(152) Huang, J.; Rauscher, S.; Nawrocki, G.; Ran, T.; Feig, M.; de Groot, B. L.; Grubmüller, H.; MacKerell, A. D. *Nature Methods* **2017**, *14*, 71–73.

(153) MacKerell Jr, A. D.; Bashford, D.; Bellott, M.; Dunbrack Jr, R. L.; Evanseck, J. D.; Field, M. J.; Fischer, S.; Gao, J.; Guo, H.; Ha, S.; MacKerell, A. D.; Bashford, D.; Bellott, M.; Dunbrack, R. L.; Evanseck, J. D.; Field, M. J.; Fischer, S.; Gao, J.; Guo, H.; Ha, S.; Joseph-McCarthy, D.; Kuchnir, L.; Kuczera, K.; T. K. Lau, F.; Mattos, C.; Michnick, S.; Ngo, T.; T. Nguyen, D.; Prodhom, B.; E. Reiher, W.; Roux, B.; Schlenkrich, M.; C. Smith, J.; Stote, R.; Straub, J.; Watanabe, M.; Wiórkiewicz-Kuczera, J.; Yin, D.; Karplus, M. *The journal of physical chemistry B* **1998**, *102*, 3586–3616.

(154) MacKerell, A. D.; Feig, M.; Brooks, C. L.; MacKerell Jr, A. D.; Feig, M.; Brooks, C. L. *Journal of the American Chemical Society* **2004**, *126*, 698–699.

(155) Klauda, J. B.; Venable, R. M.; Freites, J. A.; O'Connor, J. W.; Tobias, D. J.; Mondragon-Ramirez, C.; Vorobyov, I.; MacKerell Jr, A. D.; Pastor, R. W. *The journal of physical chemistry. B* **2010**, *114*, 7830–7843.

(156) Klauda, J. B.; Monje, V.; Kim, T.; Im, W. *The Journal of Physical Chemistry B* **2012**, *116*, 9424–9431.

(157) Lee, J.; Cheng, X.; Swails, J. M.; Yeom, M. S.; Eastman, P. K.; Lemkul, J. A.; Wei, S.; Buckner, J.; Jeong, J. C.; Qi, Y.; Jo, S.; Pande, V. S.; Case, D. A.; Brooks, C. L.; MacKerell, A. D.; Klauda, J. B.; Im, W. CHARMM-GUI Input Generator for NAMD, GROMACS, AMBER OpenMM, and CHARMM/OpenMM Simulations Using the CHARMM36 Additive Force Field, 2016.

(158) Feller, S. E.; Zhang, Y.; Pastor, R. W.; Brooks, B. R. *The Journal of Chemical Physics* **1995**, *103*, 4613–4621.

(159) Martyna, G. J.; Tobias, D. J.; Klein, M. L. *The Journal of Chemical Physics* **1994**, *101*, 4177–4189.

(160) Pal, S. K.; Mitra, S. *IEEE Transactions on Neural Networks* **1992**, *3*, 683–697.

(161) Friedman, J. H. *https://doi.org/10.1214/aos/1013203451* **2001**, *29*, 1189–1232.

(162) Breiman, L. *Machine Learning 2001 45:1* **2001**, *45*, 5–32.

(163) Congreve, M.; Dias, J. M.; Marshall, F. H. Structure-Based Drug Design for G Protein-Coupled Receptors, 2014.

(164) May, L. T.; Avlani, V. A.; Langmead, C. J.; Herdon, H. J.; Wood, M. D.; Sexton, P. M.; Christopoulos, A. *Molecular Pharmacology* **2007**, *72*, 463–476.

(165) Valant, C.; Gregory, K. J.; Hall, N. E.; Scammells, P. J.; Lew, M. J.; Sexton, P. M.; Christopoulos, A. *Journal of Biological Chemistry* **2008**, *283*, 29312–29321.

(166) Alfonzo, M. J.; Alfonzo, R. G. D.; Alfonzo González, M.; Becemberg, I. L. D. *Journal of Receptors and Signal Transduction* **2015**, *35*, 319–328.

(167) Borroto-Escuela, D. O.; Correia, P. A.; Romero-Fernandez, W.; Narvaez, M.; Fuxe, K.; Ciruela, F.; Garriga, P. *Journal of Neuroscience Methods* **2011**, *195*, 161–169.

(168) Pitcher, J. A.; Payne, E. S.; Csortos, C.; Depaoli-Roach, A. A.; Lefkowitz, R. J. *Proceedings of the National Academy of Sciences of the United States of America* **1995**, *92*, 8343–8347.

(169) Simon, V.; Guidry, J.; Gettys, T. W.; Tobin, A. B.; Lanier, S. M. *Journal of Biological Chemistry* **2006**, *281*, 40310–40320.

(170) Butcher, A. J.; Prihandoko, R.; Kong, K. C.; McWilliams, P.; Edwards, J. M.; Bottrill, A.; Mistry, S.; Tobin, A. B.; Choi Kong, K.; McWilliams, P.; Edwards, J. M.; Bottrill, A.; Mistry, S.; Tobin, A. B. **2011**, *286*, 11506–11518.

(171) Alfonzo, M. J.; De Alfonzo, R. G.; Alfonzo-González, M. A.; De Becemberg, I. L. *Molecular Membrane Biology* **2013**, *30*, 403–417.

(172) Gyun, J. S.; Jones, B. W.; Hinkle, P. M. *Proceedings of the National Academy of Sciences of the United States of America* **2007**, *104*, 18303–18308.

(173) Rovira, X.; Pin, J. P.; Giraldo, J. *Trends in Pharmacological Sciences* **2010**, *31*, 15–21.

(174) Bulenger, S.; Marullo, S.; Bouvier, M. *Trends in Pharmacological Sciences* **2005**, *26*, 131–137.

(175) Jakubík, J.; El-Fakahany, E. E.; Tuček, S. *Journal of Biological Chemistry* **2000**, *275*, 18836–18844.

(176) McMillin, S. M.; Heusel, M.; Liu, T.; Costanzi, S.; Wess, J. *Journal of Biological Chemistry* **2011**, *286*, 28584–28598.

(177) Jambrina, P. G.; Bohuszewicz, O.; Buchete, N. V.; Kolch, W.; Rosta, E. *Biochemical Society Transactions* **2014**, *42*, 784–790.

(178) Ansari, N.; Rizzi, V.; Parrinello, M. *Nature Communications* **2022**, *13*, 1–9.

(179) Schiebel, J.; Gaspari, R.; Wulsdorf, T.; Ngo, K.; Sohn, C.; Schrader, T. E.; Cavalli, A.; Ostermann, A.; Heine, A.; Klebe, G. *Nature Communications* **2018**, *9*, DOI: 10.1038/s41467-018-05769-2.

(180) Bortolato, A.; Deflorian, F.; Weiss, D. R.; Mason, J. S. *Journal of Chemical Information and Modeling* **2015**, *55*, 1857–1866.

(181) Mattedi, G.; Deflorian, F.; Mason, J. S.; De Graaf, C.; Gervasio, F. L. *Journal of Chemical Information and Modeling* **2019**, *59*, 2830–2836.

(182) Wodak, S. J.; Paci, E.; Dokholyan, N. V.; Berezovsky, I. N.; Horovitz, A.; Li, J.; Hilser, V. J.; Bahar, I.; Karanicolas, J.; Stock, G.; Hamm, P.; Stote, R. H.; Eberhardt, J.; Chebaro, Y.; Dejaegere, A.; Cecchini, M.; Changeux, J. P.; Bolhuis, P. G.; Vreede, J.; Faccioli, P.; Orioli, S.; Ravasio, R.; Yan, L.; Brito, C.; Wyart, M.; Gkeka, P.; Rivalta, I.; Palermo, G.; McCammon, J. A.; Panecka-Hofman, J.; Wade, R. C.; Di Pizio, A.; Niv, M. Y.; Nussinov, R.; Tsai, C. J.; Jang, H.; Padhorny, D.; Kozakov, D.; McLeish, T. *Structure* **2019**, *27*, 566–578.

(183) Ding, T.; Karlov, D. S.; Pino-Angeles, A.; Tikhonova, I. G. *Journal of Chemical Information and Modeling* **2022**, *62*, 4736–4747.

(184) Klink, T. A.; Woycechowsky, K. J.; Taylor, K. M.; Raines, R. T. *European Journal of Biochemistry* **2000**, *267*, 566–572.

(185) Ji, C. G.; Zhang, J. Z. *Journal of the American Chemical Society* **2011**, *133*, 17727–17737.

(186) Timko, J.; Bucher, D.; Kuyucak, S. *Journal of Chemical Physics* **2010**, *132*, 114510.

(187) Casalino, L.; Nierzwicki, Jinek, M.; Palermo, G. *ACS Catalysis* **2020**, *10*, 13596–13605.

(188) Wang, Y.; Herron, L.; Tiwary, P. *Proceedings of the National Academy of Sciences of the United States of America* **2022**, *119*, e2203656119.

(189) Lichtinger, S. M.; Biggin, P. C. *Journal of Chemical Theory and Computation* **2023**, DOI: `10.1021/acs.jctc.3c00140`.

(190) Lamim Ribeiro, J. M.; Tiwary, P. *Journal of Chemical Theory and Computation* **2019**, *15*, 708–719.