Applications of machine learning in molecular simulations

Transcending barriers

Stefan Doerr

TESI DOCTORAL UPF / ANY 2016

DIRECTOR DE LA TESI
Prof. Gianni de Fabritiis
Departament de Ciències Experimentals i de la Salut

This thesis is dedicated to the people that make this world a better place, through their actions, their ideas and their words.

Acknowledgments I would like to thank everyone who contributed to all my work so far, as well as all my friends and my family which make my life ever brighter. I would like to also thank the volunteers on GPUGRID without which we could never have performed all the invaluable calculations. Many thanks also to everyone who was and is at the multiscalelab for the great time we have together, making our working environment so much fun.

Abstract

The structures, interactions and dynamics of proteins determine their function. Molecular dynamics provide a unique tool for understanding molecular function, as it provides atomic resolution combined with time resolution. Molecular dynamics however faces multiple issues, one of which is that the intensive calculations limit the time of the processes it can resolve. Energetic barriers slow down the dynamics of the system often beyond what a single simulation can reach. Markov models together with advanced sampling techniques allow us to use short simulations to rapidly explore these slow processes. In this context, the first goal of this thesis has been to further develop sampling methods and investigate improvements in Markov model construction. The second goal of the thesis has been to advance the field of molecular dynamics by providing a unified framework for molecular discovery improving reproducibility of experiments, allowing large scale experiments and easing access of scientists to the field.

Resum

L'estructura, les interaccions i les dinàmiques determinen les funcions de les proteines. La Dinàmica Molecular constitueix una eina única per l'estudi de les funcions moleculars, ja que proveeix tant resolució atòmica com resolució en el temps. No obstant, la Dinàmica Molecular presenta una sèrie de limitacions com per exemple el fet de que els intensius càlculs necessaris per produir-ne en limiten el temps dels processos que poden resoldre. Les barreres energètiques enleteixen les dinàmiques dels sistemes més enllà del que normalment pot abarcar una sola simulació. Els Models de Markov conjuntament amb tècniques avançades de mostreig ens permeten utilitzar simulacions curtes per explorar ràpidament aquests processos lents. En aquest contexte, el primer objectiu d'aquesta tesi ha estat desenvolupar mètodes de mostreig i investigar millores en la construcció de models de Markov. El segon objectiu d'aquesta tesi ha estat avançar el camp de la Dinàmica Molecular tot proveint un programari unificat per a l'estudi molecular que millori la reproducibilitat dels experiments, que permeti experiments a gran escala i que faciliti l'accés dels científics al nostre camp.

Contents

List of Figures

Chapter 1

INTRODUCTION

1.1 Significance of structural biology

Proteins provide the building blocks of cells and thus life itself. They are also often the driving cause in disease and therefore understanding the function of proteins can help us design novel drugs for diseases as in HIV-1 [1], influenza [2] and more [3, 4, 5]. Proteins are produced through a process which converts a string of information encoded in our DNA into a three-dimensional structure. This structure and in some cases the movement encoded in that structure, dictates the protein's function (see Figure 1.1). Consequently, a mutation in the sequence can cause a structural change which in turn changes the protein's interactions with other molecules, causing disease. Thus, to investigate the function of proteins at an atomic level we need to understand their structure and ideally their movements as well.

Indeed, proteins don't exist by themselves; they perform their functions through interactions with small chemical ligands, other proteins, DNA, lipids and more. Therefore, in our study of structural biology we cannot focus only on single proteins, but on what we must view as a whole biological system which encapsulates the essential information that is required to describe the protein's function.

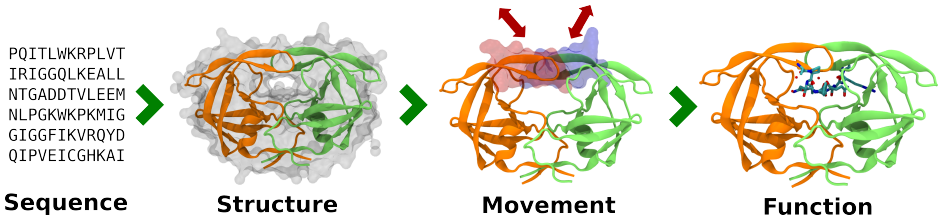


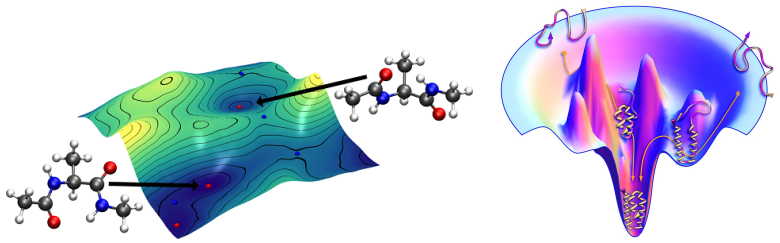
Figure 1.1: The protein aminoacid sequence of the HIV-1 protease is encoded into a 3D structure. The structure dictates the movement of the two highlighted flaps which open and close and those allow the protein to perform its function. In this case the function is demonstrated by a piece of the Gag polyprotein located in the catalytic site, which the protease cleaves.

1.2 Exploring space

Given a typical system under investigation which can consist of hundreds of thousands of atoms, its phase space (or configurational space if we do not take into account the momenta) is extremely high-dimensional. Exhaustively exploring such a space using current scientific methods is impossible in all but the most trivial cases. One such simple case of an energy surface that can be sampled quite exhaustively can be seen in Figure 1.2a. The dialanine peptide provides a nice demonstration case as due to the planarity of protein backbones, its accessible conformational space can be described quite well using only two dimensions; the ϕ and ψ dihedral angles of its backbone, with the third dimension showing the free energy of each conformation.

A popular view, as depicted in Figure 1.2b, is that energy surfaces of biological systems are very rough surfaces riddled with local minima and a funnel which leads to the global minimum which is the system's

[2] http://www.sfb716.uni-stuttgart.de/en/research/subprojects/research-area-c/c6/description.html and http://phys.org/news/2012-11-scientific-advances-result-year-puzzle.html



(a) (b)

Figure 1.2: a) Alanine dipeptide free energy surface with representative conformations shown for two minima. The surface x-y coordinates correspond to the dihedral angles and the height and color of the surface corresponds to the energy. b) Cartoon depiction of a protein folding funnel. Figures taken from [2].

functional state (i.e. folded state for proteins, bound state for protein interactions etc.). As high energy configurations have a lower probability of occurring, higher energy areas effectively become barriers in the exploration of the phase space and lower energy areas become attractors in which the system can spend a large amount of time. A biological system will typically have to overcome multiple barriers on its way of transitioning from a non-functional to a functional configuration and the heights of those barriers will determine the speed of such a process.

Even though structural biologists often focus only on investigating the lowest energy state of the system, this is mostly due to past practical and theoretical limitations. Experimental methods such as X-ray crystallography were only able to determine the single most populated structure and the prevalent image of the energy funnel theory [6] put a strong emphasis on the lowest energy state; the focus, however, has shifted in the last decades. Practically for many systems, long-lived ”metastable” states, i.e. states with a significant equilibrium probability other than the global minimum, can play an equally important role in the functioning of a biological system [7, 8, 9, 10, 11, 12, 13]. One such extreme example being a group of proteins called intrinsically disordered proteins (IDPs) which

possess very flat energy surfaces, constantly moving along that surface, while still performing important functions such as in cellular signaling [14]. Therefore, it is important to study as many states as possible and their dynamics to best understand a biological system.

1.2.1 The slow processes

By definition, such long-lived "metastable" states have to be separated from other states with large barriers. The slow processes which connect those states and their kinetics are some of the most interesting observables to study in biological systems as they correspond to protein conformational changes [15, 16], folding processes [17, 18], binding processes [19] and more. Thus, if we want to study long-lived states and their transitions in a molecular system, we need to be able to overcome these barriers through our methods of investigation. Unfortunately, few methods exist that are able to accurately determine those slow processes and therefore it has been a subject of extensive study in the field as well as one of the main subjects of this thesis.

1.3 Methods for investigating the configurational space

Investigating biomolecular structure has a long history starting from 1958 when the first three-dimensional crystal structure of Myoglobin was solved [21]. Many experimental and computational methods have been developed since then to tackle the problem of resolving the structure of proteins and their complexes. However, none of them has managed to achieve absolute dominance over the others. This is due to the fact that all of the methods are limited either in the range of time they can investigate or the range of sizes they can resolve as show in Figure 1.3.

Experimental methods are, and probably will remain, the golden standard of structural investigation. This keeps being proven by the incredible growth of the Protein Data Bank (PDB) shown in Figure 1.4. Most struc-

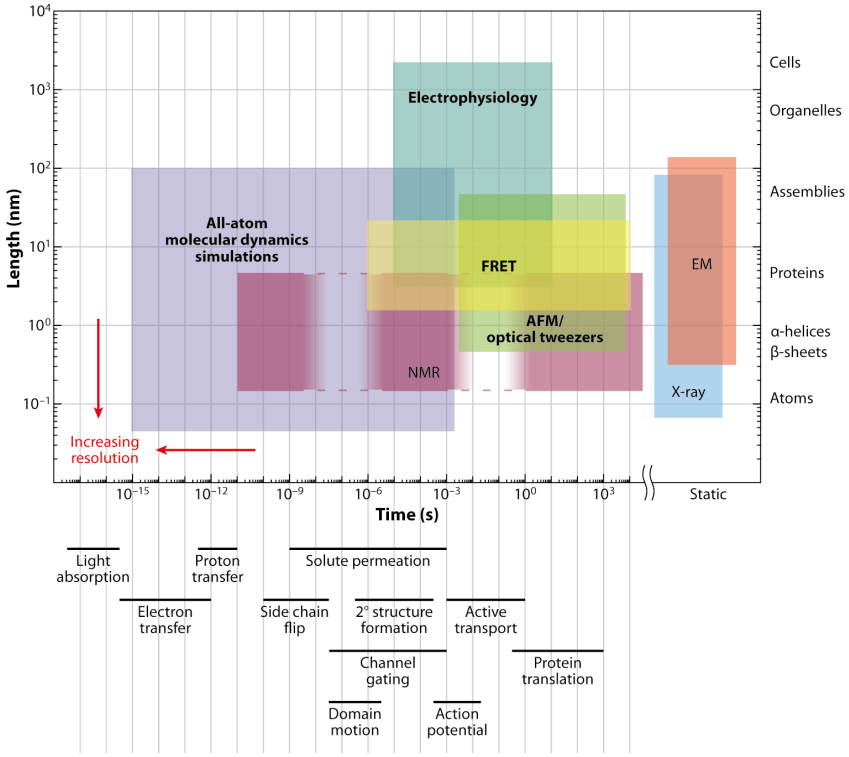


Figure 1.3: Different investigative methods used in structural biology cover a wide range of timescales and sizes that they can resolve. This plot shows the ranges each method is able to cover. Under the x-axis are given some examples of biological processes happening in those timescales and on the right of the y-axis the dimensions of some biological molecules and assemblies. Figure taken from [20].

tures are contributed through X-ray crystallography and it seems like this trend will continue, with NMR (nuclear magnetic resonance) finding its application for dynamic and ligand binding studies and the newcomer EM (electron microscopy) showing growing potential.

Methods can be grouped in static and dynamic methods. Static meth-

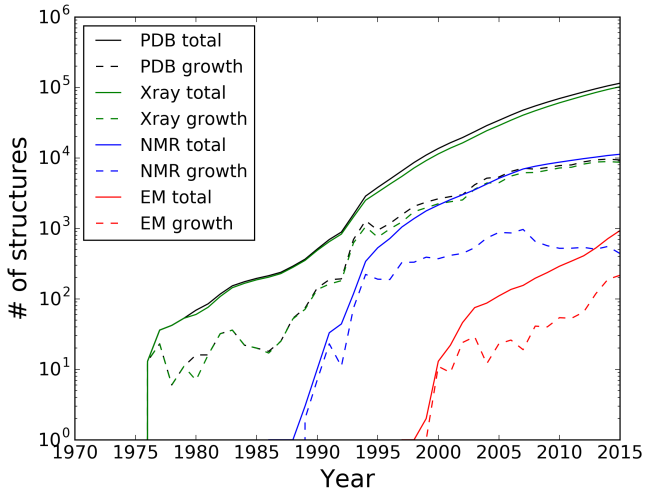


Figure 1.4: Total amount of structures resolved and yearly growth for each of the three main methods (X-ray crystallography, NMR and EM) used for structure determination of proteins in the PDB database.

ods (crystallography and EM) are able to study a huge range of system sizes. This allows them to determine with atomic resolution structures from small proteins to very large assemblies. However both crystallography and EM have limitations. They are only able to obtain a static image of the system which is a space and time average typically of the lowest energy configuration and are unable to provide kinetic information. The space and time averaging means that the image they provide is a single configuration which does not come directly from the configurational space of the system but instead from an average of points from it. As global minima are quite stable, this average is accurate enough for the largest parts of the structures to be considered as a sample from the configurational space. The exception being flexible protein loops where the point averaging problem manifests itself by often not allowing their structural determination. Crystallography is also limited by the necessity of crystallization of the proteins making for example difficult the study of

membrane proteins [22] which are of high medicinal importance. Additionally, crystal structures of proteins can cause artificial conformations in the proteins [23] as well show misleading ligand binding pockets by trapping ligands between the packed copies of the proteins [24]. Cryo-EM is able to overcome the problems of crystallization of very large macromolecular structures [25, 26, 27], but is however unable to provide the atomic resolution of crystallography.

If we want to study however slow processes, we need to use other methods which can provide a time-resolution. Such dynamic methods include experimental methods such as NMR (nuclear magnetic resonance), FRET (fluorescence resonance energy transfer), AFM (atomic force microscopy, optical tweezers) as well as computational methods such as MD (molecular dynamics). Dynamic experimental methods are able to identify multiple long-lived "metastable" states and in some cases provide kinetic information [28, 29, 30, 31, 17, 32, 33, 34]; they are however more limited in the range of sizes they can resolve compared to the static methods.

1.3.1 Molecular dynamics

Molecular dynamics (MD) provides a very strong tool for investigating slow processes as it can resolve both structural and kinetic information simultaneously at atomic resolution. Multiple studies have been performed showing its applicability on a wide range of biomolecular systems, including protein folding experiments [35, 36, 37], protein-ligand binding [38, 39, 40, 41, 42, 43, 44, 45], protein conformational changes [46, 47], ion channel simulations [48, 49, 50, 51], protein-induced membrane dynamics [52, 53, 54] and intrinsically disordered proteins [55].

MD is a computational method which represents the set of atoms of a system in phase space as point masses with velocities. It then uses numerical integration schemes such as Verlet integration to integrate Newton's equations of motion and propagate the movement of the atoms based on interaction forces between them. As the real quantum processes occurring between atoms are very computationally expensive to simulate,

MD uses approximations of the forces between atoms called force-fields [56, 57, 58]. This means that MD does not move the system along the exact energy surface of the system but instead tries to approximate that surface. This can obviously lead the simulations to unphysical regions of the phase space producing erroneous results, which has caused significant criticism and skepticism over the accuracy of MD. Recent advances [59, 60, 61, 62], however, have significantly improved forcefields and demonstrate by comparison to experiments that the approximation of that energy surface is adequate for many problems [63, 35].

Another limitation of MD is that for integration methods to maintain numerical stability, the integration time-step has to be significantly smaller than the fastest processes in the system. As bond vibrations in biological systems happen at timescales as low as tens of femtoseconds, MD is forced to use integration steps in that range (typically less than 5fs). Unfortunately, most biologically interesting processes tend to happen at much higher timescales, ranging from nanoseconds to minutes and longer. This means that huge amounts of integration steps (10^6 steps for nanosecond simulations) have to be performed to simulate those events. Until recently these computations were carried out on consumer CPUs and CPU clusters. However, due to the high degree of parallelization in force calculations and great advancements in GPU technology, computations have now moved to GPUs which has provided a great boost in computing power, allowing single simulations to reach up to microseconds in a few days of simulation time.

Due to this increase in computational power, MD has now become a viable option for investigating slow biological processes (see examples in Figure 1.5). Compared to experimental methods it is able to sample much larger areas of the phase space, at full atomic resolution, allowing the identification of previously unseen long-lived states as well as the kinetics between them [45, 55, 38]. Currently, two philosophies of sampling are prevalent. The one being the performance of few long simulations on expensive specialized hardware [64, 65, 66], while the other, called high-throughput MD, being the performance of large amounts of relatively short simulations on cheap hardware or crowd-computing plat-

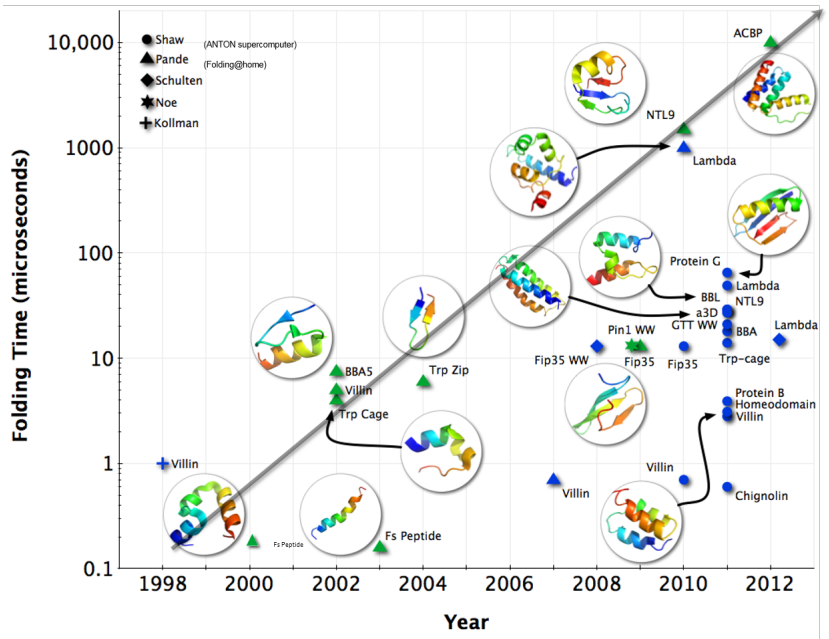


Figure 1.5: Increase in computing power over the last years has lead to the investigation of increasingly slow biological processes, here demonstrated on protein folding. Figure courtesy of Vijay Pande.

forms [67, 68]. On first sight, a large amount of short simulations does not seem very appealing as a way for investigating slow processes. The development however of the methodology behind Markov state models, has made high-throughput MD a very appealing option with tangible results.

1.4 Markov state models

Markov state models are kinetic models which model the dynamics of a system as a memory-less jump process. Markov models essentially attempt to approximate the propagator of the continuous dynamics of a system [69, 70]. To better understand what the propagator is, we have

to switch from a single-configuration view to an ensemble view of our system which means that we are looking at a distribution of configurations of our system in the configurational space. The propagator Q then propagates or moves a distribution of configurations $x(t)$ found at time t through time, showing what that distribution will look like in the future. In the time-continuous version it does this by $dx(t)/dt = Q * x(t)$ and in the time-discrete case with time-step τ, it does by $x(t+\tau) = Q(\tau) * x(t)$. The propagator only considers the current state of the system to calculate future states, which makes it Markovian (memory-less). Due to its nicer mathematical properties we will consider in the rest of the text the transfer operator $\mathcal{T}$ which is closely related to the propagator Q, but instead of propagating probability densities, propagates probability densities weighted by the stationary distribution $\mu(x)$. In an example of a simple four-well 1D energy potential we easily visualize the stationary distribution of the system (Figure 1.6a) and the transfer operator (Figure 1.6b).

In most MD analysis cases we choose to work with discrete Markov state models which simplify the problem by discretizing both the configurational space of the system as well as the time. For example in Figure 1.6a one can see how the 1D potential can be discretized into four states (A,B,C and D). In this case of time- and space-discrete Markov models we are approximating the transfer operator $\mathcal{T}$ by a transition probability matrix T over the discrete states. The elements of the transition probability matrix $T(\tau)$ give the probabilities $T_{ij}(\tau)$ of moving from state i to state j after a time τ, while losing however any information of transitions within the boundaries of the state itself. As we can see, when we define a discrete-time Markov model we have to decide on a jump time τ also known as the lag-time for which the matrix gives the transition probabilities. Since dynamics of a molecular system are certainly not memory-less (different momentum can drive the same starting configuration to different ending configurations), choosing a lag-time which is longer than the longest equilibration time among the states helps us define a jump time for the model at which our system is approximately memory-less.

The powerful advantage of Markov models is that they allow us, as

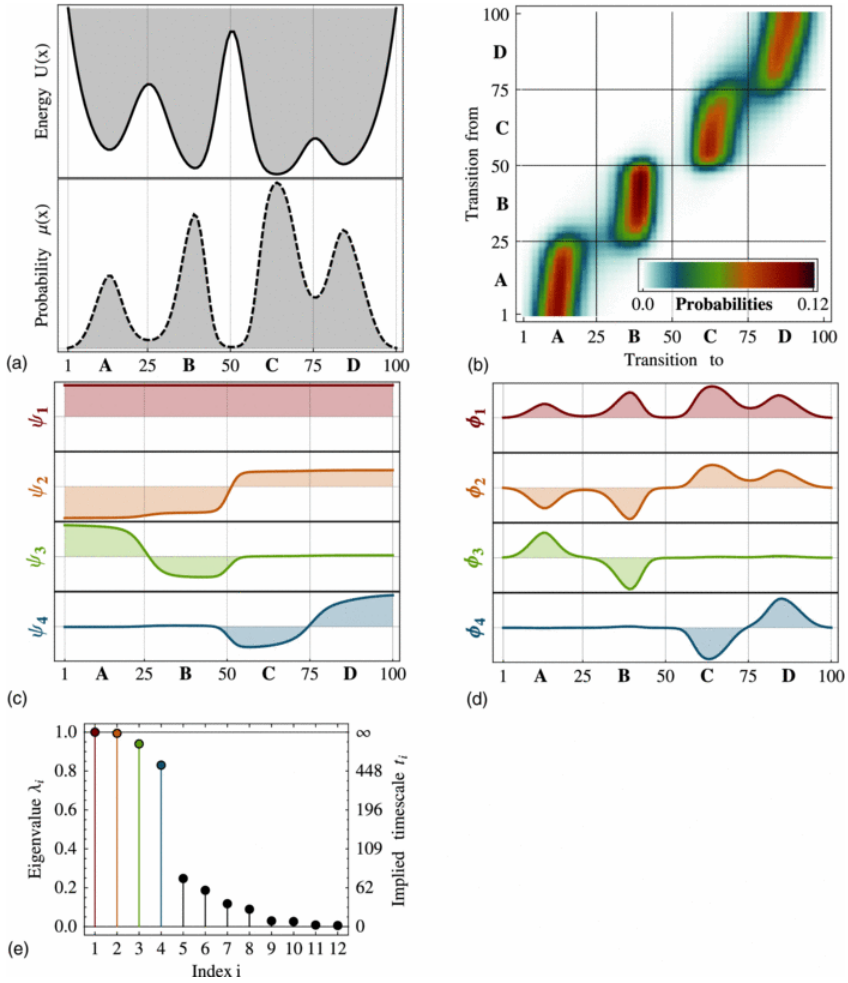


Figure 1.6: The figure shows a 1D energy potential and its equilibrium distribution in a), the transfer operator in b), the eigenvectors of the transfer operator in c), the eigenvectors weighted by the stationary distribution in d) and the eigenvalues in e). Figure taken from [70].

mentioned before, to combine individual simulations into a single statistical model and thus are perfectly suited for high-throughput molecular dy-

namics. By simply discretizing the configurational space explored during the simulations using a clustering technique, we can count the transitions c_{ij} that occurred from any state i to any state j within lag-time τ and use this count matrix C to estimate the transition probability matrix $\hat{T}$. This can be done with a simple maximum-likelihood estimator or better with estimators which take into account the fact that the matrix needs to be reversible [70]. Thus we can obtain a Markov model from thousands of independent trajectories which describes all the dynamics explored during those simulations. As we only need to count transitions between states, it is not necessary for a single simulation to explore the entire slow process but instead the Markov model allows us to use much shorter simulations as long as they are able to overcome the individual barriers between the discrete states within the lag-time of the model.

1.4.1 Relation to slow processes

So how do these Markov models relate to the slow processes of a system? Due to T being a Markov transition matrix, its first eigenvalue λ_1 is guaranteed to be 1. This means for the eigenvalue equation $T\phi_i = \lambda_i\phi_i$, where λ_i and ϕ_i the i-th eigenvalue and eigenvector of the matrix, that $T\phi_1 = \phi_1$. What this tells us is that our transition probability matrix does not modify anymore the probabilities of the vector ϕ_1 which means that there is no more change in the system and hence we have reached the equilibrium distribution. Therefore, the first eigenvector ϕ_1 corresponds to the equilibrium distribution $\mu(x)$ of the system as can be seen in Figure 1.6d. This allows for a very interesting interpretation. The total slow dynamics of a system are a superposition of individual slow processes and these slow processes are each associated to the eigenvalues and eigenvectors of the matrix T. The sign of the eigenvectors, once normalized by the equilibrium distribution as in Figure 1.6c tells us with which set of states each slow process is associated (second eigenvector being A,B to C,D which indeed has the highest barrier, third A to B and fourth C to D). Thus the eigenvector sign structure can be used to identify the long-lived metastable states at specific time-resolutions. On the other hand, the

eigenvalues tell us the timescale t_i of slow process i by using the relation $t_i = -\frac{\tau}{ln\lambda_i}$ as seen in Figure 1.6e. Beyond that, the transition matrix can also be used to calculate on- and off-rates between states, free energies and fluxes which can be used to investigate the transition pathways leading from any state to any other.

1.4.2 Discretization

All of these properties of a Markov state model are very useful, however there are some caveats in the construction of a model. As mentioned before, discretizing the configurational space and thus the transfer operator has the consequence that all information on transitions between configurations inside a state is lost and the only information that is kept is about transitions between states. This means that if we want to model correctly the slow dynamics of our system we need to take great care of where we place the barriers of our states. In the case of Figure 1.6a and the first and second row of Figure 1.7 it seems quite obvious that placing the limits of the states on top of the energetic barriers will allow us to describe the slow dynamics of the system near perfectly. If we made the mistake of placing the limits of the states on any other position as in the 3rd row of Figure 1.7 we would incur a high error in our estimation of the slow dynamics due to wrongly estimated eigenvectors and eigenvalues. However, we can do an even better job of estimating the slow dynamics than by placing our state limits directly on the barriers. As discrete space eigenvectors approximate the real eigenvectors of the system using step-functions, we can increase the resolution of the eigenvectors in the transition regions by either naively adding more states as in the 4th row of Figure 1.7 or better by very finely discretizing the barriers as in the 5th row of Figure 1.7, making the approximated eigenvector more nuanced and exact and thus reducing the approximation error. This approximation error of eigenvalues and eigenvectors is critical for a Markov model analysis as it has been shown to affect all above-mentioned quantities obtained from a Markov model [71, 70, 72, 73].

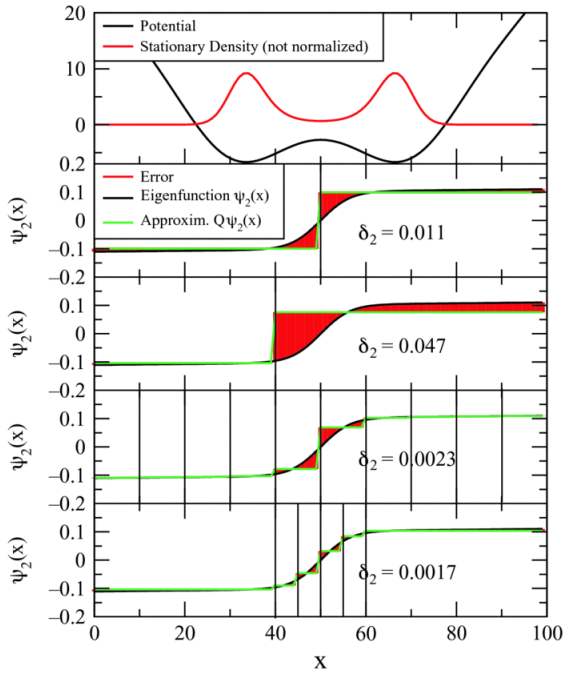


Figure 1.7: A double-well potential and its equilibrium distribution in the first row. Following rows show various attempts at discretizing the space (vertical black lines) with more or less states and different locations as well as the real (black sigmoid) and approximated (green) second eigenvector which corresponds to the slowest process of the system, and the approximation error δ_2. Figure taken from [70].

1.4.3 Dimensionality reduction

The fact that more states on transition regions result in better estimations of the slow processes is critical for the construction of Markov models. The barrier locations are however unknown a priori to the clustering methods which we use for state definition. Therefore, the one alternative is to use large amounts of states, hoping that enough of them will fall on the barriers and thus reduce our approximation error. The other alternative is

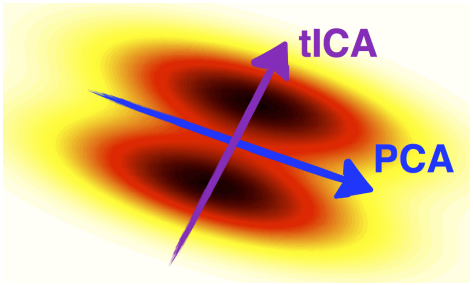


Figure 1.8: A double-well potential and the first independent component of TICA and principal component of PCA. The x and y dimensions correspond to a 2D configurational space and increasingly dark colors indicate lower energy areas. Figure credited to C. R. Schwantes.

to project our configurational space onto a different space which stretches transition regions (barriers) and condenses flat minima, thus forcing our clustering methods to put most states on the transition regions. Various projection methods can be used for that and since after projecting we can typically do away with the dimensions which correspond to fast processes, these are practically used as dimensionality reduction methods.

One of the most used dimensionality reduction methods for MD data in the past has been PCA [74, 75, 76]. If we assume that the largest configurational change in our system corresponds to the slowest process, then PCA is perfectly suited to project our configurations for MSM construction. It projects the configurations on the dimensions of the largest variance and thus would allow us to put more clusters on the barriers. Unfortunately biological systems are rarely that simple. Often, the slowest process is one which causes minimal changes to the configuration i.e. a high barrier separates very close regions of the configurational space. This has been demonstrated in work where for example a small register shift in a beta-sheet corresponded to a slow process [77].

This reasoning lead to the development of a projection method very well suited for solving this problem. TICA [77, 69, 78] extends PCA by combining the information in the covariance matrix with information

from a time-lagged covariance matrix to project simulation data onto the slowest degrees of freedom, thus minimizing the eigenvector and eigenvalue approximation errors. TICA thus provides the optimal projection for the slow dimensions in the regime of linear projection methods. As such, it has found wide application in recent publications [79, 80, 78].

To illustrate the difference between PCA and TICA, an example is given in a simple double-well potential in Figure 1.8 where large configurational variation does not correspond to the slowest process and TICA ends up being orthogonal to PCA.

1.5 Adaptive sampling methods

As molecular dynamics simulations try to accurately reconstruct the system's kinetics, we can imagine that crossing high barriers is one of the biggest issues, as the higher the barrier, the lower the probability of crossing it. Indeed naive high-throughput simulations which are all initiated from similar initial conditions will tend to get stuck in the same regions of phase space wasting lots of simulation time, while neither improving the quality of the Markov model, nor detecting new long-lived states or finding the global minimum.

Various enhanced sampling techniques have been used in the past to overcome barriers [81], including umbrella sampling [82, 83], accelerated MD [84, 85], metadynamics [86, 87, 88] and simulated tempering and replica-exchange simulations [89, 90, 91]. They all attempt either by applying biases or by increasing the temperature to accelerate the sampling of slow processes. The problem with all these methods however is that even when they are able to sample well the phase space, they cannot provide kinetics, which are of great interest when trying to make quantitative connections to experiments. Additionally, many of them require prior knowledge of the system, and the choice of reaction coordinates can hide dynamics orthogonal to those. Coarse-grained MD simulations [92, 93, 94] also allow for faster simulations and thus faster sampling of slow processes, they are however not sufficiently accurate in cases where

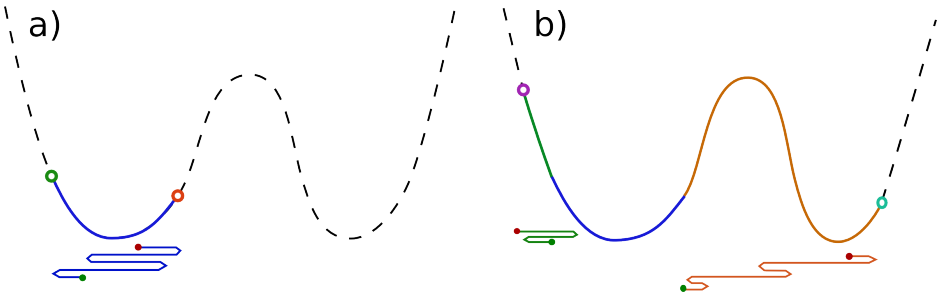


Figure 1.9: Example of an adaptive sampling technique on a 1D double-well potential. In a) a single simulation explores the first minimum (solid blue line) while the rest is unexplored (dashed line). The two hollow circles are chosen as the points from which to start two new simulations (green and orange) which as shown in b) manage to sample a larger area of the potential. Next simulations could start from the purple and teal circles. The curved lines under the potential illustrate simulations moving along the single coordinate starting from the green point and ending at the red.

detailed interactions are important as in small fragment binding.

Adaptive sampling is a methodology developed for MD simulations and typically used in high-throughput MD which aims to accelerate the sampling of the phase space without biasing or low-dimensional reaction coordinates, thus providing valid kinetics and avoiding the waste of computation associated with naive sampling techniques. The principle behind all adaptive sampling methodologies is to utilize information obtained from previous simulations to guide further sampling of the phase space in a more intelligent way. This guiding is not done using biases; pure MD simulations are run, producing unbiased kinetics. Instead, once a number of simulations have completed, the knowledge gathered from these simulations is used to determine from where to initiate new simulations. We can imagine that such a checkpointing mechanism can also make it possible to cross large barriers, similarly to how base camps are used when climbing mountains. By discretizing the crossing process

into many small steps and by focusing the simulations on sampling the slow processes identified through previous simulations, the barriers can be overcome in a faster manner (see Figure 1.9 for an example). This paradigm allows adaptive sampling methods to utilize very short trajectories and high-throughput environments to run swarms of simulations in parallel which only have to reach the next checkpoint instead of simulating the entire slow process in one go. It is shown in multiple studies that such adaptive sampling methods can speed up the exploration of the phase space or convergence of a desired observable by orders of magnitude [95, 96, 97].

The relationship between adaptive sampling methods and mathematical optimization methods is worth mentioning, as they both face similar issues of transcending barriers in their search. Heuristic optimization methods which attempt to overcome barriers such as particle swarm optimization [98] and tabu search [99, 100] can serve as inspiration for the design of adaptive methods. The related discussion in optimization literature on the trade-off between exploration and exploitation (i.e. overcoming barriers and following a gradient) has also inspired new work in the field [96] which exploits experimental information of the system to guide the sampling while balancing it with exploration to avoid getting trapped in local minima, showing an order of magnitude faster convergence to the desired observable than through other adaptive methods.

1.6 Molecular discovery and research reproducibility

The final subject which has occupied this thesis are three big problems in the field of molecular discovery. The accessibility of high-throughput MD, the reproducibility of molecular simulation experiments and the ability to handle the large amounts of data associated and generated by high-throughput experiments. As shown in Figure 1.10, a typical molecular discovery pipeline can consist of multiple steps leading from a molecular structure to a whole prepared system and finally through MD simulations

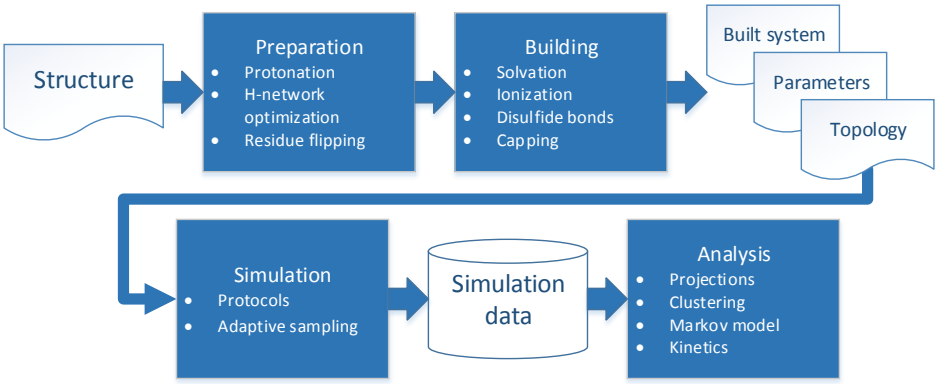


Figure 1.10: A typical molecular simulation pipeline starting from a structure file and ending in an analysis producing the desired observables such as kinetics.

to the desired observables.

Each of the steps of this pipeline is traditionally performed with the use of different tools. Different simulation software (like AMBER [57], CHARMM [56], GROMACS [101] etc.) often came with their own forcefields as well as their own tools to prepare simulations for them. Therefore, the choice of the desired simulation software would also dictate the forcefield, parametrization and preparation tools that are used. On the analysis part, multiple molecular simulation analysis packages exist [102, 103, 104] with overlapping functionality. As a result, the choice of the combination of tools with which this pipeline will be assembled is left up to the researcher whose preferences often depend on personal experience.

However, molecular dynamics as many other academic areas is a field full of abandoned or over-complicated software, dated protocols and lacking documentation and standards. Forcefield modifications might appear without publications supporting or demonstrating the practical effects they have on simulations; in other cases they are modified by labs independent of the original developers. Often, the user of a given simula-

tion software cannot depend on the tools of that software for the simulation preparation as some tools have not been developed for that software. For example, to perform a membrane simulation in AMBER, their protocol [3] advises the user to produce a membrane through VMD [105] which makes CHARMM format membranes and then convert it to AMBER format by using a Python script. Web-servers have proliferated as well (like CHARMM-GUI [106], ParamChem [107], etc.) due to their ease of use or due to licensing issues prohibiting the distribution of the software.

As a result, the typical MD simulation pipeline can consist of a collection of bash scripts, files obtained through web-servers, modified forcefield files, a host of different software for preparing, building and analyzing a system spanning multiple programming languages and graphical interfaces. Essentially, even in the case of a perfectly documented procedure, such a pipeline can be near impossible to reproduce often even by the same person. Tool versions change, web-servers disappear, files get forgotten in the file-system. This does not help advance the field of molecular dynamics which already has enough challenges to face in becoming an established tool for molecular discovery. To confound the problem even more, the amount of data associated with high-throughput experiments is ever-increasing. The ability of scientists to investigate thousands of systems in parallel can produce big organizational problems as the pipeline needs to be automated and replicated thousands of times.

For these reasons, there is a very urgent need in the direction of making molecular discovery more consistent, automatized and reproducible through standardization and a uniform platform which allows start-to-end molecular discovery.

[3] `http://ambermd.org/tutorials/advanced/tutorial16/`

Chapter 2

OBJECTIVES

The main objective of the thesis has been the advancement of high-throughput MD to improve the discovery and description of slow biomolecular processes. This advancement has been attempted by two contributing factors. The first being the development of adaptive methods that allow the acceleration of phase space exploration by learning from existing simulation data; the second being the development of a unified and easy to use framework to support and improve molecular discovery using MD simulations.

2.1 Accelerate molecular simulations

Adaptive sampling methods provide a powerful tool for accelerating the sampling of the phase space. By learning from previous simulations they can intelligently sample the phase space avoiding traps and speeding up computations without biasing the kinetics. Many different variants have been proposed and in this thesis in all three publications we present adaptive sampling in different forms, from manual adaptive sampling to automatic sampling based on Markov state models. The objective has been to develop a general adaptive sampling method that would work on both protein folding and protein-ligand binding systems and would provide a significant speed advantage to justify its implementation. In the duration

of the thesis, adaptive sampling has grown to become the standard sampling methodology for various groups including our own and has been applied to many biological systems and problems.

2.2 Integrated platform to support high-throughput molecular dynamics

The second goal has been to ease the access to high-throughput MD for scientists. By developing a software called HTMD we aimed for the development of a platform that allows start-to-end molecular discovery in a single environment with minimal external dependencies. The objective in this is to make molecular discovery much more accessible to the wider audience including people who have never used MD before, increase the reproducibility of MD based experiments and simplifying the management of high-throughput experiments and their associated data.

Chapter 3

PUBLICATIONS

3.1 Kinetic Characterization of Fragment Binding in AmpC β-Lactamase by High-Throughput Molecular Simulations

P. Bisignano, S. Doerr, M. J. Harvey, A. D. Favia, A. Cavalli, and G. De Fabritiis. Journal of Chemical Information and Modeling 54. 362-366 (2014).

Summary
In this paper we demonstrated the binding of a small fragment (carboxythiophene) to the AmpC β-Lactamase. As the crystal structure of the fragment bound to the protein shows multiple binding poses, the purpose of the work was to demonstrate the ability of Markov state models and high-throughput MD to obtain kinetics of binding for all crystallographically detected poses of the fragment. Through the simulations we were able to see that most poses were quite unstable, which lead us to conclude that these poses could be stabilized by the crystal packing of the protein and saturation effects. Additionally we were able to observe how one of the poses seen in the crystal structures (distal site) corresponds to a very wide binding area in the oxyanion hole in which the fragment can asso-

ciate freely with the protein. Lastly, we showed how the binding site in the tunnel site was hiding another even deeper binding pocket which opens through a conformational change in the protein which happens through both conformational selection and induced fit.

Bisignano P, Doerr S, Harvey MJ, Favia AD, Cavalli A, De Fabritiis G. Kinetic characterization of fragment binding in AmpC β-lactamase by high-throughput molecular simulations. J Chem Inf Model. 2014 Feb 24;54(2):362-6. doi:10.1021/ci4006063

3.2 On-the-Fly Learning and Sampling of Ligand Binding by High-Throughput Molecular Simulations

S. Doerr and G. De Fabritiis. Journal of Chemical Theory and Computation 10. 2064-2069 (2014)

Summary
In this paper we decided to pursue the first application of automatic adaptive sampling to a protein-ligand binding system. We decided to investigate the Benzamidine-Trypsin system and developed an automatic adaptive sampling strategy based on Markov models. The adaptive strategy employed in this study uses the unbinding time calculated by the MSM to restart simulations from the ligand poses with the highest unbinding time (lowest k_{off}). This worked exceedingly well for the specific system, speeding up calculations by an order of magnitude and the visual nature of the binding process allowed us to display in a graphical way how adaptive methods work.

Doerr S, De Fabritiis G. On-the-Fly Learning and Sampling of Ligand Binding by High-Throughput Molecular Simulations. J Chem Theory Comput. 2014 May 13;10(5):2064-9. doi: 10.1021/ct400919u

3.3 HTMD: High-Throughput Molecular Dynamics for Molecular Discovery

S. Doerr, M. J. Harvey, Frank Noé and G. De Fabritiis. Journal of Chemical Theory and Computation 12. 1845-1852 (2016).

Summary
In this paper we present a software we developed called HTMD. With this software we decided to tackle three big problems in our field; namely the difficulty of entry for beginners, secondly the irreproducibility of experiments and third the ability to perform and manage large amounts of simulation-based experiments. We managed to provide a unified platform in Python which can be used to perform molecular discovery starting from a PDB file and performing the whole pipeline of preparing, building and simulating a biomolecular system as well as analyzing the resulting simulation data with the powerful tools provided by Markov state modeling. Therefore, in a single script a user can go from a PDB structure to kinetics with minimal knowledge of molecular dynamics and without needing to refer to any other external tools, programs or scripts. With the integration of Jupyter notebooks, an experiment can be now written from inside a browser as a nicely formatted report including text, images, plots and code showing all steps necessary for replicating the experiment. Additionally we had the chance to demonstrate the integration of adaptive sampling straight into the HTMD platform as well as our new adaptive sampling methodology which is applicable to both ligand binding and protein folding and which provided an order of magnitude speedup.

Doerr S, Harvey MJ, Noé F, De Fabritiis G. HTMD: High-Throughput Molecular Dynamics for Molecular Discovery. J Chem Theory Comput. 2016 Apr 12;12(4): 1845-52. doi: 10.1021/acs.jctc.6b00049

Chapter 4

UNPUBLISHED WORK

In this chapter we present work that was performed for two further publications. Some of the results presented here are still preliminary.

4.1 Reproducible system building and simulation of membrane proteins with HTMD

S. Doerr, T. Giorgino, M. J. Harvey and G. De Fabritiis.

Summary
In our previous work we already presented the simulation analysis and adaptive sampling capabilities of HTMD. In the current work we are introducing the system preparation, building, simulation and protocols functionality available through HTMD. System building and simulation is a critical part of the MD based molecular discovery pipeline and as such, HTMD intends to provide all tools necessary to reach from single PDBs to a fully built system for one of the two currently supported forcefields (CHARMM, AMBER) and then allow the simulation of the system on different simulation resources through a standardized interface. To demonstrate the power that such a tool provides, we automatically build and equilibrate 648 protein membrane systems using structures from the

OPM database. We then perform a short analysis to determine the quality of the built systems and their equilibration.

Reproducible system building and simulation of membrane proteins with HTMD

S. Doerr,[†,‖] Toni Giorgino,[‡,‖] M. J. Harvey,[¶] and G. De Fabritiis[*,§]

†*Computational Biophysics Laboratory (GRIB-IMIM), Universitat Pompeu Fabra, Barcelona Biomedical Research Park (PRBB), C/ Doctor Aiguader 88, 08003 Barcelona, Spain*

‡*Institute of Neurosciences, National Research Council of Italy (IN-CNR), 35127 Padua, Italy*

¶*Acellera, Barcelona Biomedical Research Park (PRBB), C/ Doctor Aiguader 88, 08003 Barcelona, Spain*

§*Institució Catalana de Recerca i Estudis Avançats (ICREA), Passeig Lluis Companys 23, Barcelona 08010, Spain*

‖*Contributed equally to this work*

E-mail: gianni.defabritiis@upf.edu

Abstract

HTMD is a programmable scientific platform intended to simplify and increase reproducibility of simulation-based research. In this paper we demonstrate the functionalities of HTMD that enable the preparation of a molecular dynamics simulation starting from a PDB structure, parametrize it through well-known forcefields, apply standardized protocols, and launch the corresponding simulations on a variety of computational resources. We demonstrate the automation potential and flexibility of the system building features of HTMD, applying it to hundreds of proteins in the OPM (Orientation of Proteins in Membranes) database, automatically building and performing all-atom simulations of most eukaryotic membrane proteins resolved to date.

1 Introduction

Classical molecular-dynamics (MD) is a compute-intensive technique which simulates the dynamic behavior of atomistic systems by modeling them as many-body systems moving under the Newtonian forces determined by a classical potential. The quality of the potential energy functions has undergone extensive refinements over the last decades, and is vastly responsible for the predictive ability of MD which has been demonstrated by its application over a spectrum of biomolecular systems encompassing globular, transmembrane and unstructured proteins, drug-ligand interactions, and several others.[1–6] MD based discovery however still faces a variety of problems which hinder its wider application.

Historically, simulation software like AMBER,[7] CHARMM,[8] GROMACS,[9] LAMMPS[10] etc. introduced both force-fields and their corresponding file formats. Many were developed with their own set of tools for preparing molecular systems and applying the forcefield to the corresponding models. Even though more recent MD engines introduced support for multiple forcefields (e.g. ACEMD,[11] NAMD,[12] OpenMM,[13] Desmond[14]), the multiplicity of combinations of file formats and preparation procedures still hampers the steps preliminary to the actual simulation, making them ad-hoc efforts, not easily documented and repro-

ducible.

There is therefore a need for a protein-oriented framework to enable the building of systems in a fast, reproducible and hands-off manner, making it possible to process hundreds or thousands of structures with minimal intervention, and then simulate them, possibly in a way independent of (a) the chosen forcefield; (b) the simulation software and (c) the computing resources to be employed.

In this paper we will focus on the first part of MD based molecular discovery by describing the system preparation, building and simulation facilities introduced in the HTMD framework meant to satisfy the previously stated requirements. The facilities of HTMD for large-scale and Markov-model based post-simulation analysis have been described in a previous publication.[15]

1.1 Related work

Over the last years, various software packages have been developed to aid molecular dynamics system preparation, building and simulation. These can be largely grouped into three categories: graphical user interface based packages, web-based services and programming frameworks.

Programs such as VMD,[12] Maestro[16] and MOE[17] have met with great success as they provide powerful graphical user interfaces for *interactive* molecule inspection, manipulation and system preparation. The visual aspect makes them very accessible and allows users to easily and quickly verify the results of their actions as well as modify and undo changes. It however makes it hard to integrate them into a pipeline without access to a strong scripting language and their manual aspect and user intervention limits their application to only a few systems at a time.

Web-based interfaces such as CHARMM-GUI[18] and MDWeb[19] have also become very popular. They provide a web-browser-based service for preparing a system for simulation. As such, no software is required and they are accessible from anywhere with an internet connection. CHARMM-GUI especially has become very powerful, which after the latest developments[20] supports system building for multiple forcefields and simulation software such as NAMD, GROMACS, AMBER and OpenMM, making it extremely flexible. The modifications that can be performed are however restricted to the set of operations exposed through the web service as the user cannot intervene between steps to manually modify files. Additionally, it is very inconvenient to integrate them into a unified molecular discovery workflow and it can be relatively difficult to reproduce the whole procedure which is documented in log files and would need to be redone manually through drop-down menus, text fields and radio boxes. Web-servers and graphical interfaces, in other words, fall short of the objective of providing a scriptable and reproducible framework for building and simulating MD systems.

Important efforts have already been made towards programmable frameworks for system building and simulation, including VMD,[12] ParmEd,[7] OpenMM[13] and Ensembler.[21] VMD, in particular, allows the building of CHARMM systems through the TCL scripting language; ParmEd allows topology and parameter modification and conversion between various forcefield formats; OpenMM allows building systems for AMBER, AMOEBA[22] and CHARMM forcefields as well as simulation on CPU and GPUs and Ensembler can build and equilibrate whole protein superfamilies through the use of template modeling.

These frameworks provide very strong starting points for molecular simulation based discovery. Although HTMD might lack some functionality distributed in the aforementioned software packages, to our knowledge it is the only software providing an *integrated environment* for the system preparation, building, simulation and analysis. This allows user to perform molecular discovery, going from raw PDB files to the estimation of thermodynamics and kinetics of binding and folding, conveniently expressed and documented through the Python language.

The use of a mature and rich programming platform such as Python permits users to import and seamlessly integrate domain-specific

third-party libraries, such as RDKit for cheminformatics,[23] Scikit-learn[24] for machine learning, NumPy for custom data analysis and innumerable others.

2 Methods

2.1 Protein preparation

The first step required for all-atom computational experiments is the pre-processing of the three-dimensional structures provided as PDB files as found in the RCSB database. This step is required because the location of protons is generally not resolved in crystallographic experiments, and must be recovered by indirect methods. One of the approximations done in most current high-throughput simulation methods is the assumption of constant protonation states of chemical groups.

Until constant-pH simulation methods become commonplace,[25] it is important to setup the simulated system so that residues are in the protonation states most likely for their chemical environment. A related issue arising when preparing a system is whether neutral histidine residues should be protonated at the δ or ϵ nitrogen. Finally, carboxamide groups of ASN and GLN sidechains have distinctly asymmetric hydrogen bonding donor-acceptor arrangements; although they are usually free to flip during the dynamics, a proper collective arrangement of their orientation minimizes the electrostatic energy and optimizes the hydrogen-bonding network, thus allowing simulations to start from a more stable configuration.

HTMD allows the estimation of protonation states and optimization of the hydrogen-bonding network through a `proteinPrepare()` method (Listing 1). The preparation procedure relies in the first place on the PROPKA3.1 software[26,27] for estimating residues' charge state; PROPKA estimates pKa values on the basis of residue's models, desolvation effects, hydrogen bonding, and other electrostatic interactions. After the computation of per-residue pKa values, a modified version of the PDB2PQR software[28] is used to perform the combinatorial optimization of the hydrogen bonding network. PDB2PQR uses a Monte Carlo algorithm to sample χ dihedral angles of HIS, ASN and GLN residues to optimize their hydrogen-bonding network; δ or ϵ nitrogen protonation of histidines is also decided at this stage according to the most stable configuration, together with the resolution of major steric conflicts through a "debumping" algorithm.[29]

Invoking the system preparation function returns a protein structure protonated according to the optimum H bonding criterion and most likely charge states at the specified pH. The structure can be used at later stages of the building process. In particular, the charge states of residues are marked with names that are independent of the forcefield, which allows the later system building functions to apply the modifications in the way appropriate for the specific back-end (e.g. patches, in CHARMM's terminology).

The preparation function also returns a data structure with the decisions taken at the various stages of the preparation procedure; in particular, pKa values, protonation states, any required χ-dihedral flipping, etc., are returned in a data structure suitable for both interactive and scripted inspections. In the case of transmembrane proteins, the data returned includes a warning flag for titratable residues exposed to the hydrophobic environment, whose pKa predictions may be inaccurate.[30]

2.2 System building

HTMD provides a `Molecule` class with a suite of methods designed to facilitate system manipulation, preparation and visualization.[15] In particular, methods are available to convert a molecule taken from the PDB into a fully solvated and protonated system that can be used as a starting configuration for a simulation. It can be solvated in a water box, various ions can be added, and caps and other forcefield patches can be applied to the molecules at the system-building stage. Currently, HTMD has back-ends for building systems in both CHARMM (PSF) and AMBER (PRMTOP)

Listing 1: Protein preparation

```
# The opiod mu receptor dimer (pre-oriented)
m = Molecule("4DKL.pdb")

# Prepare and optimize at pH 7
mopt, pd = proteinPrepare(m, pH=7.0,
    hydrophobicThickness=32.0,
    returnDetails=True)

# Save a report with the modification details
    (protonation, pKa, flipped for H-bonding,
    solvent exposure, etc.)
pd.data.to_excel("mor_report.xlsx")

# Verify histidines' protonation state
his = (pd.data.resname == "HIS")
pd.data[his][["resname", "resid",
    "protonation"]]

# Check membrane-exposed residues
memb_exp = pd.data.membraneExposed
pd.data[memb_exp] \
    .to_excel("mor_exposed_residues.xlsx")
```

Listing 2: System building

```
# Solvate the system
mol = solvate(mol)
# Build using default arguments
bmol = charmm.build(mol, outdir='./build')
# Override various options
charmm.listFiles()
topos  = ['top/top_all36_prot.rtf',
          './benzamidine.rtf']
params = ['par/par_all36_prot.prm',
          './benzamidine.prm']
caps = { 'A': ['first ACE', 'last CT3'],
         'B': ['none', 'none'] }
disu = [DisulfideBridge('A',321,'A',18),
        DisulfideBridge('B',  2,'B',26)]
cmol = charmm.build(mol, topo=topos,
    param=params, caps=caps, disulfide=disu,
    saltconc=0.15, saltcation='POT')
# Build for AMBER
topos = ['./benzamidine.prepi']
params = ['./benzamidine.frcmod']
amol = amber.build(mol, topo=topos,
    param=params, disulfide=disu,
    saltconc=0.15, saltcation='K+')
```

formats; importantly, to maximize compatibility with evolving features of said formats, the actual writing of the topologies is relayed to the well-tested software distributed with the corresponding forcefields (namely `tleap`[7] and `psfgen`[12]).

The default behaviour of HTMD building functions, `charmm.build()` and `amber.build()`, is to (1) to neutralize the system by automatically adding ions; (2) add neutral caps to protein terminals; and (3) automatically detect CYS-CYS disulfide bridges and bond them. Any of these automatic choices can be overridden by the user, who can also change forcefield versions, provide his own topology and parameter files, disable ionization or capping of proteins, provide a desired salt concentration, select ion types, manually specify disulfide bonds and specify any modifications or patches supported by the underlying forcefield as shown in Listing 2. The flexibility is especially important for the correct modeling of post-translational modifications and non-standard residues.

Both building functions accept an instance of the `Molecule` class as input and have the same interface. The user can therefore switch between forcefields and simulation software-specific formats with minor modifications to the input parameters.

2.3 Protocols

A successfully built system is ready for simulation. The typical simulation sequence, encompassing minimization, equilibration and production runs, is encoded in a set of protocols in HTMD. Protocols provide sensible default configurations as high-level procedures which can be used as-is, while still being flexible enough to be modified by advanced users for systems for which the default settings are not appropriate. An example usage of the protocols is given in Listing 3.

The equilibration protocol of HTMD consists of an initial system minimization of 500 steps, followed by an equilibration run (Figure 1). Minimization allows water molecules to reorganize, filling in empty spaces and protein side-

Listing 3: Equilibration and production runs using protocols.

```
eq = Equilibration()
eq.numsteps = 100000
eq.temperature = 300
eq.write('./build', './equil')

mdx = AcemdLocal()
mdx.submit('./equil')
mdx.wait()

md = Production()
md.numsteps = 5000000
md.temperature = 300
md.write('./equil', './prod')

mdx.submit('./prod')
mdx.wait()
```

chains to resolve clashes. During the first half of the equilibration, protein heavy atoms are constrained, to allow the rest of the system to equilibrate around the starting protein structure. These constraints are gradually relaxed during the first half of the equilibration; the rest is run without constraints. Additionally, the first 500 steps of equilibration are run in the constant-volume (NVT) ensemble to account for changes in pressure due to the resolving of clashes. The rest of the equilibration is run in the constant-pressure (NPT) ensemble. The user is allowed to change the length of the equilibration runs, provide his own constraints and apply flat-bottom potentials to specific atoms to constrain their movement within a box. The production protocol provided by HTMD can be run either directly on a system provided by the user, or pick off where the equilibration protocol finished by using the produced output files. The production protocol then runs a standard NVT simulation, also allowing application of flat-bottom potentials.

2.4 Apps: Abstracting resources and software from MD runs

MD simulations can be run on different MD engines and each of those engines provides support for specific simulation devices such as

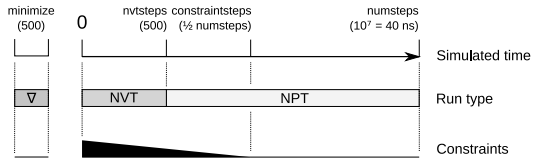


Figure 1: Time sequence of the default equilibration protocol. A steepest descent minimization (∇) is followed by runs in the constant-volume (NVT) and constant-pressure (NPT) ensemble. Run times for the various steps are indicated by the corresponding keywords and default lengths (in simulation steps, usually 4 fs each).

CPUs and GPUs. Additionally, the simulations can be run on a range of different infrastructures, from local computers, to remote clusters or even crowd-computing platforms. Therefore, HTMD provides the `App` class which defines a unique interface for communicating with various simulation software and computing resources. This allows the user to correspondingly change the computing resource and simulation software he is using with minor modifications. More specifically, an `App` sub-class handles all communication with a resource like a cluster or local queuing system, as well as the particular MD software running on it. It exposes an interface consisting of three main methods, namely: a `submit()` method to handle the sending, queuing and starting of simulations; `retrieve()`, which handles the retrieval of completed simulations from the remote resources; and the `inprogress()` method which polls the queuing system about how many simulations are currently running and queued. Apps can be run either asynchronously or synchronously using the `wait()` method which will block script execution until all queued simulations have completed. HTMD is packaged with Apps for running and queuing simulations with the ACEMD engine,[11] either locally (on a single or multiple GPUs), or on the Amazon EC2 cloud. An example of the `AcemdLocal` class which implements a local queuing system for GPU simulations using ACEMD is shown in Listing 3.

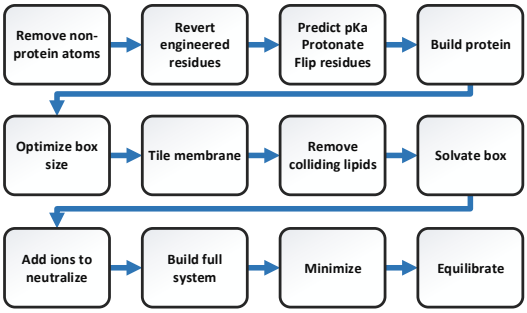


Figure 2: OPM building and simulation protocol workflow starting from a PDB file and ending with an equilibrated system.

3 Results

To demonstrate the efficacy and power that a scriptable environment can provide for system building and simulation, we prepared and simulated most of the eukaryotic proteins available in the OPM database using HTMD. This dataset consists of 708 proteins and thus provides a comprehensive and realistic case for the design and test of an unsupervised automatic build-and-simulate protocol.

The Orientations of Proteins in Membranes (OPM) database provides the predicted spatial arrangements of currently over 3000 membrane proteins, oriented with respect to the hydrophobic core of the bilayer.[31] The OPM provides PDB files, annotated by dummy atoms to indicate the position of the membrane. For this study we worked on the subset of 708 OPM proteins which are located in the eukaryotic plasma membrane. We expect that the rest of the OPM can be built and simulated using the same protocol by substituting the membrane PDB file for the corresponding membrane the proteins are embedded in.

3.1 Preparation protocol

The protocol used for automatically building and simulating all eukaryotic membrane proteins (Figure 2) consists of the following steps. First, all non-protein atoms are removed from the OPM PDB file and all engineered residues are mutated back to their parent residues according to the information automatically retrieved from the RCSB website.[32] The reason for this is that the parameters of arbitrary ligands, cofactors and engineered residues are not defined in the forcefields or easily obtained. (Protocols exist for the parametrization of ligands, but they are out of the scope of this paper; this step was omitted to decrease complexity and runtime.) Next, the likely protonation states of the protein residues are assigned at pH 7, and the hydrogen bonding network is optimized as described in section 2.1. Disconnected protein segments are detected to allow terminal capping; finally, the protein is built in the CHARMM format to add the terminal caps, missing atoms and residue sidechains.

To minimize the total system size, the protein is rotated around the z-axis to have its largest variation in the diagonal and thus best fit into the cubic simulation box. The simulation box is then created to the size of the protein plus 20 Å of additional space to avoid self-interactions of the protein between periodic images. Since we are working with POPC pre-equilibrated membrane PDB files of fixed dimensions, the membrane is tiled along the x and y axes so as to cover the whole x and y plane of the simulation box. Steric clashes between the membrane and the protein are resolved by removing lipid molecules having atoms within 1.3 Å of protein atoms.

Many proteins however form pores inside the membrane, as in the case of ion channels, and the previous rule does not remove lipids inside those pores. Therefore, the protocol uses a convex hull method to detect lipids located inside the protein. It first calculates the convex hull of the protein atoms which are located inside the membrane and then sequentially tests each lipid if it is located inside this hull. If constructing a new convex hull including both the atoms of a lipid and the protein does not modify the convex hull, one concludes that the lipid is located inside the protein and is therefore removed. In this manner, pores are cleaned of lipids.

The system is finally solvated with TIP3P water and neutralized with ions. The built system is minimized and equilibrated for 40 ns using

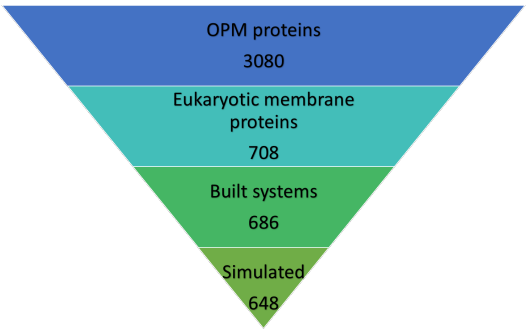


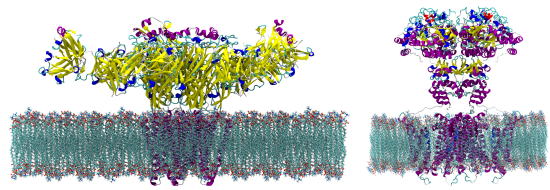
Figure 3: The OPM currently consists of 3,080 protein structures. From those we selected to build and simulate the 708 eukaryotic membrane proteins. Of those, 686 were successfully built and finally 648 simulated.

ACEMD on the volunteer distributed computing resource GPUGRID, following the protocol described in Figure 1.

As shown in Figure 3, out of the 708 OPM proteins, 686 were successfully built using the protocol. Of the 22 proteins that failed to build, 15 failed due to original structures lacking too many heavy atoms (indicating poorly resolved structures which would be hardly suitable for simulation), while the other 7 failed due to various inconsistencies between the OPM and PDB residue naming, or missing information on the RCSB webpage. Out of the remaining 686 systems, 38 caused errors during simulation in ACEMD and had to be discarded, usually because of their large size did not allow processing in a single-GPU setup, which is the baseline set-up for GPUGRID users. In the end, we performed the 40 ns equilibration run for 593 systems on GPUGRID, while another 55 were simulated locally due to too long runtimes (more than 48 hours runtime on an NVIDIA 780 GPU). The two largest built and simulated systems can be seen in Figure 4.

3.2 System size

As can be seen in Figure 5 a, eukaryotic membrane proteins in the OPM span a wide range of sizes. Proteins successfully built varied in size between 10 and 3,697 residues (1.1 to 410.5 kDa). Unsurprisingly, the distribution of protein sizes is skewed towards the lower molecular weights, the 90% of them being between 3.0 and 175.5 kDa, with a median of 31.1 kDa.



(a) 3RHW (b) 2R9R

Figure 4: a) largest built system (3RHW) with 733375 atoms. b) largest simulated system (2R9R) with 346204 atoms.

This data set provides a huge and representative variety of protein structures and membrane embedding patterns to be studied. Figure 5 b shows how final system sizes tend to contain an order of magnitude more atoms through the addition of membrane lipids, water molecules and ions, with a high correlation between the number of residues of the OPM file and the final atom count of the system (Spearman's $\rho = 0.93$). Outliers are due to the overall tertiary and quaternary topologies of the proteins, which might be more spread out in the plane parallel to the membrane, requiring less solvent, or normal to it, requiring more solvent (Figures 5 c-d). Shape and localization of the proteins also affect the number of lipids removed from the bilayer by the convex-hull based procedure previously described, the relation being roughly linear with the number of protein residues, and a median of 162 molecules removed (Figure 5 e).

3.3 Protein positioning in membrane

To analyze the location of the protein atoms in the OPM we can divide the system space into three regions and see what fractions of their atoms are located in each. The first region is defined as the solvated region (bulk), the second being the hydrophilic region of the membrane and the third being its hydrophobic core (defined as the region within the innermost lipid

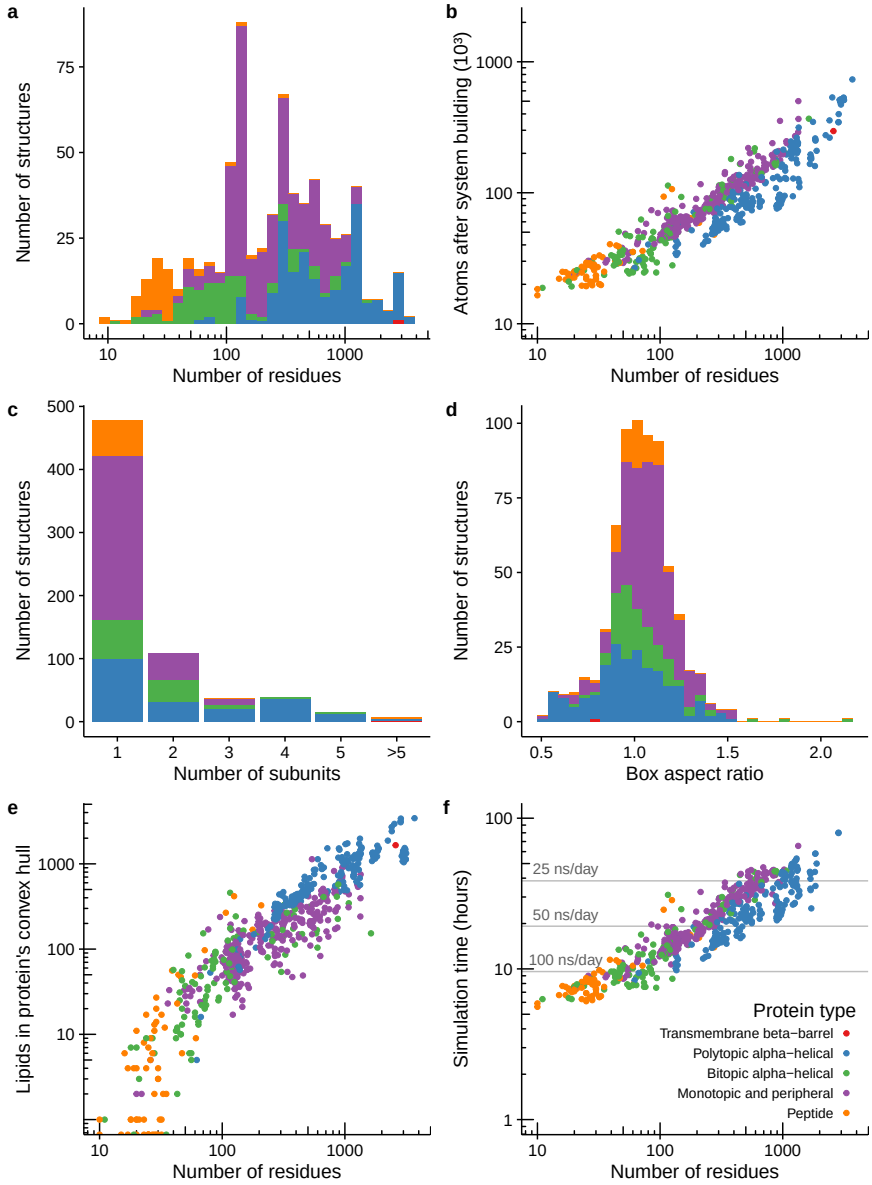


Figure 5: (a) Distribution of sizes (resolved residues) for eukaryotic membrane proteins in the OPM database used in this study. (b) Size of the systems built by the OPM building procedure, in thousands of atoms. (c) Distribution of the number of sub-units (usually due to multimerization). (d) Distribution of aspect ratios of the simulation boxes after building, defined as $a = z/(xy)^{1/2}$. (e) Number of lipid molecules removed by the convex hull procedure to generate the minimal membrane pore accommodating the protein. (f) Simulation time required to run the 40 ns equilibration protocol. In all plots, color indicates the protein type according to the OPM classification.

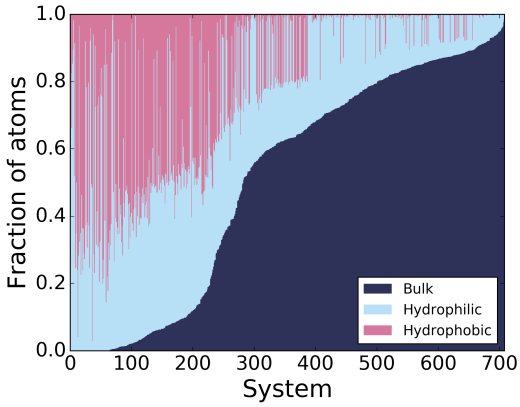


Figure 6: The fraction of protein atoms for each system which are located in the bulk of the solvent, in the hydrophilic part, and in the hydrophobic part of the membrane. Systems are sorted by increasing fraction of atoms in bulk.

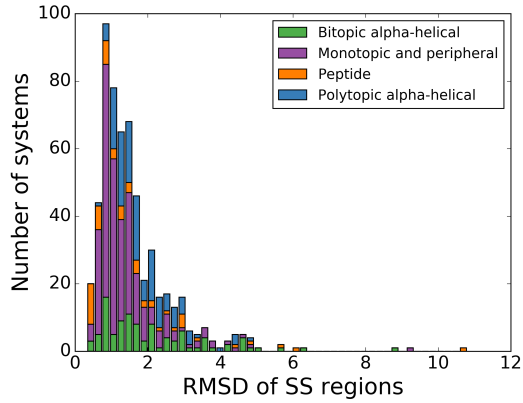


Figure 7: RMSD of the secondary structure regions of the last simulation frame after 40 ns of equilibration with respect to the starting configuration.

oxygen atoms). We expect that proteins which don't reach into the hydrophobic core of the membrane to make weaker interactions with the membrane and thus be less stable. As can be seen in Figure 6, the OPM dataset covers a wide range of protein-membrane interactions, with some proteins existing purely in the membrane (in some cases only extending horizontally on the hydrophilic part) and others having most of their mass outside on the membrane and only injecting a small tail into it.

3.4 Equilibration results

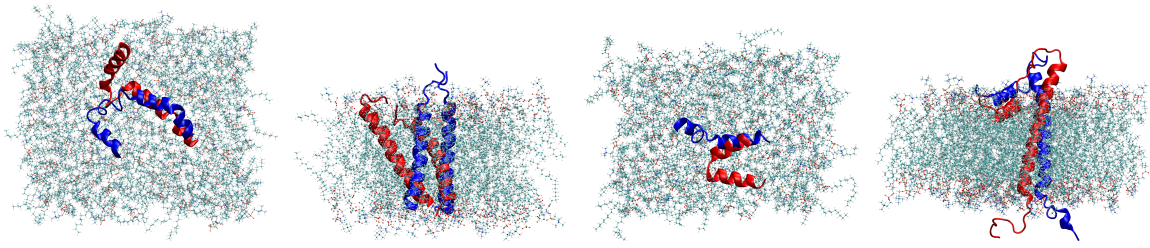
Figure 7 shows the distribution of root mean-squared distance (RMSD) of the secondary structure regions of the proteins at the end of the equilibration simulations compared to the starting structure. Loop regions were excluded from the calculation due to them being very flexible as well as many NMR structures in the OPM containing extended loops in non-native positions. The total RMSD however does not change drastically if loops are included as can be seen in SI Figure S1. From these figures we can see that most proteins stayed quite stable during the equilibration, with only minor conformational changes mostly under 4 Å RMSD.

Inspection of the seven systems with RMSD over 5 Å (Figure 8) reveals that most of them can be attributed to either non-bonded helices detaching from one another in the membrane (Figure 8 b,e), or hinge regions in the proteins which allow a region of the protein which is not interacting strongly with the membrane to flip to other conformations (Figure 8 a,c,g). The high RMSD of the 1U5E structure is due to missing loops in the PDB structure and thus splitting what is a dimer into 4 non-bonded chains, of which only one is tethered to the membrane, while the other three are free to move in bulk.

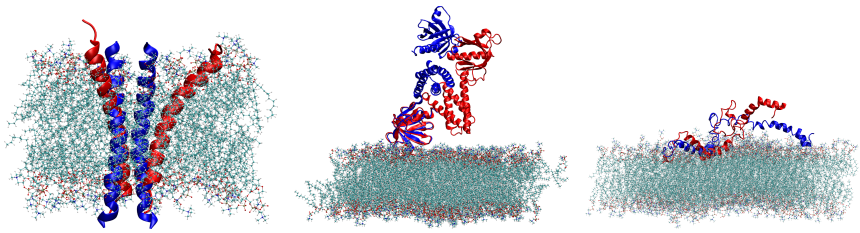
The above data are reassuring with respect to the stability of systems produced by the OPM building protocol described.

4 Conclusion

HTMD provides a unified framework for MD based discovery. It currently allows structural manipulation as well as all necessary functionality such as protonation, solvation, ionization, capping and more, for building systems in both CHARMM and AMBER formats. Additionally, through the use of predefined protocols and `Apps` it simplifies simulation setup and execu-



(a) 1Z9I 5.72 Å (b) 2L34 5.76 Å (c) 2KBV 6.14 Å (d) 2JO1 6.25 Å



(e) 2MEU 8.73 Å (f) 1U5E 9.31 Å (g) 2MP3 10.79 Å

Figure 8: The seven protein structures with RMSD higher than 5 Å after equilibration. Blue conformation is the starting conformation and red corresponds to the conformation after 40 ns of equilibration. a) and c) are shown in top-down view to better illustrate the change, while the rest of the figures are in side view.

tion on a variety of computational resources. Options are available to the user at every step to accommodate both starting and advanced users.

The power of HTMD is demonstrated on the building and simulation of most eukaryotic membrane proteins of the OPM. Many improvements are still possible on the given protocol. Protein-ligand interactions can be important for the stability of the membrane proteins. Parametrization of ligands found bound to the protein in the OPM would allow us to simulate them together with the rest of the system and monitor the dynamics between the proteins and the ligands. Unresolved loops of proteins can also be modeled automatically by integrating a loop-modeling software like Modeller[33] or Rosetta[34] into the pipeline, improving the protein models such as in the demonstrated case of 1U5E. The following step would then be the building and simulation of the remaining non-eukaryotic OPM proteins by using their corresponding membranes and compositions.

The demonstrated flexibility, simplicity and power of the software accommodates users with a wide range of expertises allowing them to apply MD to their problems with minimal coding. The integration of all necessary tools of the building and simulation pipeline into a single framework increases reproducibility and allows the easy automation of high-throughput simulations as in the example given of the OPM proteins where it is able to build hundreds of different proteins each with its own peculiarities in an automated and timely manner. We expect these developments to open the gates to high-throughput MD to many more scientists, allow for larger scale MD-based discovery and increase reproducibility, thus decreasing the amount of effort associated with this work.

Acknowledgement We thank Acellera Ltd for funding. GDF aknowledges support from MINECO (BIO2014-53095-P) and FEDER. We thank the volunteers of GPUGRID for donating their computing time for the simulations.

Supporting Information Available: This material is available free of charge via the Internet at `http://pubs.acs.org/`.

References

(1) Lindorff-Larsen, K.; Piana, S.; Dror, R. O.; Shaw, D. E. *Science* **2011**, *334*, 517–520.

(2) Jensen, M.; Jogini, V.; Borhani, D. W.; Leffler, A. E.; Dror, R. O.; Shaw, D. E. *Science* **2012**, *336*, 229–233.

(3) Buch, I.; Giorgino, T.; Fabritiis, G. D. *PNAS* **2011**,

(4) Kohlhoff, K. J.; Shukla, D.; Lawrenz, M.; Bowman, G. R.; Konerding, D. E.; Belov, D.; Altman, R. B.; Pande, V. S. *Nat. Chem.* **2014**, *6*, 15–21.

(5) Stanley, N.; Esteban-Martín, S.; De Fabritiis, G. *Nat Commun* **2014**, *5*, 5272.

(6) Reubold, T. F.; Faelber, K.; Plattner, N.; Posor, Y.; Ketel, K.; Curth, U.; Schlegel, J.; Anand, R.; Manstein, D. J.; Noé, F.; Haucke, V.; Daumke, O.; Eschenburg, S. *Nature* **2015**, *525*, 404–408.

(7) Case, D.; Betz, R.; Botello-Smith, W.; Cerutti, D.; Cheatham, T.; Darden, T.; Duke, R.; Giese, T.; Gohlke, H.; Goetz, A.; Homeyer, N.; Izadi, S.; Janowski, P.; Kaus, J.; Kovalenko, A.; Lee, T.; LeGrand, S.; Lin, C.; Luchko, T.; Luo, R.; Madej, B.; Mermelstein, D.; Merz, K.; Monard, G.; Nguyen, H.; Nguyen, H.; Omelyan, I.; Onufriev, A.; Roe, D.; Roitberg, A.; Sagui, C.; Simmerling, C.; Swails, J.; Walker, R.; Wang, J.; Wolf, R.; Wu, X.; Xiao, L.; York, D.; Kollman, P. AMBER 2016. 2016.

(8) 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. *J Comput Chem* **2009**, *30*, 1545–1614.

(9) Abraham, M. J.; Murtola, T.; Schulz, R.; Páll, S.; Smith, J. C.; Hess, B.; Lindahl, E. *SoftwareX* **2015**, *1–2*, 19–25.

(10) Plimpton, S. *Journal of Computational Physics* **1995**, *117*, 1–19.

(11) Harvey, M. J.; Giupponi, G.; De Fabritiis, G. *J. Chem. Theory Comput.* **2009**, *5*, 1632–1639.

(12) Phillips, J. C.; Braun, R.; Wang, W.; Gumbart, J.; Tajkhorshid, E.; Villa, E.; Chipot, C.; Skeel, R. D.; Kalé, L.; Schulten, K. *J. Comput. Chem.* **2005**, *26*, 1781–1802.

(13) Eastman, P.; Friedrichs, M. S.; Chodera, J. D.; Radmer, R. J.; Bruns, C. M.; Ku, J. P.; Beauchamp, K. A.; Lane, T. J.; Wang, L.-P.; Shukla, D.; Tye, T.; Houston, M.; Stich, T.; Klein, C.; Shirts, M. R.; Pande, V. S. *J. Chem. Theory Comput.* **2013**, *9*, 461–469.

(14) Bowers, K. J.; Chow, E.; Xu, H.; Dror, R. O.; Eastwood, M. P.; Gregersen, B. A.; Klepeis, J. L.; Kolossvary, I.; Moraes, M. A.; Sacerdoti, F. D.; Salmon, J. K.; Shan, Y.; Shaw, D. E. Scalable algorithms for molecular dynamics simulations on commodity clusters. Proceedings of the 2006 ACM/IEEE conference on Supercomputing. Tampa, Florida, 2006; p 84.

(15) Doerr, S.; Harvey, M. J.; Noé, F.; De Fabritiis, G. *J. Chem. Theory Comput.* **2016**, *12*, 1845–1852.

(16) Schrödinger Release 2016-1: Maestro. 2016.

(17) Molecular Operating Environment (MOE). 2016.

(18) Jo, S.; Kim, T.; Iyer, V. G.; Im, W. *J. Comput. Chem.* **2008**, *29*, 1859–1865.

(19) Hospital, A.; Andrio, P.; Fenollosa, C.; Cicin-Sain, D.; Orozco, M.; Gelpí, J. L. *Bioinformatics* **2012**, *28*, 1278–1279.

(20) 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. *J. Chem. Theory Comput.* **2016**, *12*, 405–413.

(21) Parton, D. L.; Grinaway, P. B.; Hanson, S. M.; Beauchamp, K. A.; Chodera, J. D. *PLOS Comput Biol* **2016**, *12*, e1004728.

(22) Shi, Y.; Xia, Z.; Zhang, J.; Best, R.; Wu, C.; Ponder, J. W.; Ren, P. *J. Chem. Theory Comput.* **2013**, *9*, 4046–4063.

(23) RDKit: Open-source cheminformatics. `http://www.rdkit.org`.

(24) 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, E. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.

(25) Donnini, S.; Tegeler, F.; Groenhof, G.; Grubmüller, H. *J Chem Theory Comput* **2011**, *7*, 1962–1978.

(26) Olsson, M. H. M.; Søndergaard, C. R.; Rostkowski, M.; Jensen, J. H. *J. Chem. Theory Comput.* **2011**, *7*, 525–537.

(27) Søndergaard, C. R.; Olsson, M. H. M.; Rostkowski, M.; Jensen, J. H. *J. Chem. Theory Comput.* **2011**, *7*, 2284–2295.

(28) Dolinsky, T. J.; Czodrowski, P.; Li, H.; Nielsen, J. E.; Jensen, J. H.; Klebe, G.; Baker, N. A. *Nucl. Acids Res.* **2007**, *35*, W522–W525.

(29) Dolinsky, T. J.; Nielsen, J. E.; McCammon, J. A.; Baker, N. A. *Nucl. Acids Res.* **2004**, *32*, W665–W667.

(30) Teixeira, V. H.; Vila-Viçosa, D.; Reis, P. B. P. S.; Machuqueiro, M. *J. Chem. Theory Comput.* **2016**, *12*, 930–934.

(31) Lomize, M. A.; Lomize, A. L.; Pogozheva, I. D.; Mosberg, H. I. *Bioinformatics* **2006**, *22*, 623–625.

(32) Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E. *Nucl. Acids Res.* **2000**, *28*, 235–242.

(33) Fiser, A.; Do, R. K.; Sali, A. *Protein Sci* **2000**, *9*, 1753–1773.

(34) Rohl, C. A.; Strauss, C. E. M.; Misura, K. M. S.; Baker, D. *Meth. Enzymol.* **2004**, *383*, 66–93.

SUPPLEMENTARY INFORMATION: Reproducible system building and simulation of membrane proteins with HTMD

S. Doerr,[1] Toni Giorgino,[2] M. J. Harvey,[3] and G. De Fabritiis[4,1,*]

[1] *Computational Biophysics Laboratory (GRIB-IMIM), Universitat Pompeu Fabra, Barcelona Biomedical Research Park (PRBB), C/ Doctor Aiguader 88, 08003 Barcelona, Spain*[†]

[2] *Institute of Neurosciences, National Research Council of Italy (IN-CNR), 35127 Padua, Italy*[†]

[3] *Acellera, Barcelona Biomedical Research Park (PRBB), C/ Doctor Aiguader 88, 08003 Barcelona, Spain*

[4] *Institució Catalana de Recerca i Estudis Avançats (ICREA), Passeig Lluis Companys 23, Barcelona 08010, Spain*

1

* Electronic address: gianni.defabritiis@upf.edu

† Contributed equally to this work

1 S. Doerr, M. J. Harvey, F. Noé, and G. De Fabritiis, J. Chem. Theory Comput. **12**, 1845 (2016), ISSN 1549-9618, URL `http://dx.doi.org/10.1021/acs.jctc.6b00049`.

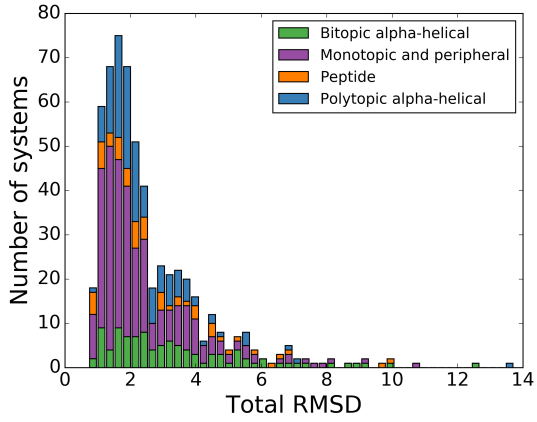


FIG. S1: RMSD of the last simulation frame after 40 ns of equilibration compared to starting structures.

```python
from htmd import *
from htmd.builder.builder import removeLipidsInProtein, minimalRotation, tileMembrane


def opmbuilder(pdbid, protfile, membfile, outdir, dbcursor=None):
    try:
        prot = Molecule(protfile)
    except:
        print('ERROR: File {} failed to load.'.format(protfile))
        return

    # Remove non-protein atoms
    prot.remove('not protein')

    # Explicitly remove ligands
    ligands = findLigands(pdbid)
    for l in ligands:
        try:
            idx2 = prot.atomselect('resname {}'.format(l), indexes=True)
            prot.remove(idx2)
            idx = np.hstack((idx, idx2))
            print('Removed resname {}'.format(l))
        except:
            pass

    # Mutate engineered residues back to their parents
    tomutate = findMutatedResidues(pdbid)
    for mut in tomutate:
        if tomutate[mut] != '--':
            try:
                prot.mutateResidue('resname {}'.format(mut), tomutate[mut])
            except:
                pass

    # Automatically detect protein segments
    prot = autoSegment(prot)

    # Build protein for charmm to add caps
    try:
        prot = prepareProtein(prot)
    except:
        raise RuntimeError('Protein preparation failed. This is probably due to too many missing
    ↪  backbone atoms.')

    prot.remove('not protein')

    # Build protein solo
    try:
        prot = charmm.build(prot, ionize=False, outdir='/tmp/build/')
    except:
        raise RuntimeError('Protein-only build failed. Check the logfile in /tmp/build/log.txt')

    # Rotate proteins around Z to have the maximum variance in the box XY diagonal
    r = minimalRotation(prot)
    prot.rotate([0, 0, 1], r)
    print('Rotated the protein by {} degrees around the Z axis.'.format(np.degrees(r)))

    # Move protein to [0,0] x,y coordinates
    meanpos = np.mean(prot.get('coords'), axis=0)
    prot.moveBy([-meanpos[0], -meanpos[1], 0])

    # Replicate the membrane
    membrane = Molecule(membfile)
    minc = np.min(prot.coords, axis=0).flatten()
    maxc = np.max(prot.coords, axis=0).flatten()
    buffer = 20  # Add 20 A of membrane around the protein
    memb = tileMembrane(membrane, minc[0]-buffer, minc[1]-buffer, maxc[0]+buffer, maxc[1]+buffer)

    # remove any lipids inside the protein hull
    memb, num = removeLipidsInProtein(prot, memb)

    # Append the membrane removing collisions
    system = prot.copy()
    system.append(memb, collisions=True, coldist=1.3)

    # Calculate box dimensions
    minz = np.min(system.coords, axis=0)[2] - 5
    maxz = np.max(system.coords, axis=0)[2] + 5
    minxy = np.min(memb.get('coords', 'water'), axis=0)
    maxxy = np.max(memb.get('coords', 'water'), axis=0)
    print([[minxy[0], minxy[1], minz[0]], [maxxy[0], maxxy[1], maxz[0]]])

    # Solvate the system
    system = solvate(system, minmax=[[minxy[0], minxy[1], minz[0]], [maxxy[0], maxxy[1], maxz[0]]])

    # Build the system
    try:
        system = charmm.build(system, outdir=outdir)
    except:
        raise RuntimeError('full build failed')
    return system
```

(a)

```python
def findMutatedResidues(pdbid):
    import requests
    from bs4 import BeautifulSoup
    tomutate = {}

    res =
    ↪  requests.get('http://www.rcsb.org/pdb/explore.do?structureId={}
    soup = BeautifulSoup(res.text, 'lxml')
    table = soup.find(id='ModifiedResidueTable')

    if table:
        trs = table.find_all('tr')

        for tr in trs:
            td = tr.find_all('td')
            if td:
                mutname = td[0].find_all('a')[0].text.strip()
                orgname = td[5].text.strip()
                print('{} was mutated to {}'.format(mutname,
    ↪  orgname))
                tomutate[mutname] = orgname
    return tomutate


def findLigands(pdbid):
    import requests
    from bs4 import BeautifulSoup
    ligands = []

    res =
    ↪  requests.get('http://www.rcsb.org/pdb/explore.do?structureId={}
    soup = BeautifulSoup(res.text, 'lxml')
    table = soup.find(id='LigandsTable')
    if table:
        trs = table.find_all('tr')

        for tr in trs:
            td = tr.find_all('td')
            if td:
                name = td[0].find_all('a')[0].text.strip()
                ligands.append(name)
    return ligands

def write_minim_equilibration(builddir, equildir, force=False):
    from htmd.protocols.equilibration_v1 import Equilibration
    folders = glob(os.path.join(builddir, '*', ''))
    for f in folders:
        if not os.path.exists(os.path.join(f, 'structure.pdb')):
            continue
        pdbid = os.path.basename(os.path.normpath(f))
        outfolder = os.path.join(equildir, pdbid)
        if not os.path.exists(outfolder):
            os.makedirs(outfolder)

        if os.path.exists(os.path.join(outfolder,
    ↪  'structure.pdb')) and not force:
            continue

        eq = Equilibration()
        runtime = 40  # in ns
        eq.numsteps = int(runtime * 1E6 / 4)  # divide by 4fs to
    ↪  get steps
        eq.temperature = 300
        eq.useconstantratio = True  # Need this for membrane
    ↪  simulations to keep the xy axis ratio fixed
        eq.write(f, outfolder)

def run_minimization(minimdir, force=False):
    from natsort import natsorted
    folderstmp = natsorted(glob(os.path.join(minimdir, '*', '')))

    if force:
        folders = folderstmp
    else:
        folders = []
        for f in folderstmp:
            if not os.path.exists(os.path.join(f, 'output.coor')):
                folders.append(f)

    print('Simulating {} folders.'.format(len(folders)))
    ac = AcemdBoinc()
    ac.submit(folders)
```

(b)

FIG. S2: The code of the OPM building protocol.

4.2 Dimensionality reduction methods for molecular simulations

S. Doerr, I. Ariz, M. J. Harvey and G. De Fabritiis.

Summary
In this work we tried to test and apply various dimensionality reduction methods to molecular simulation data. Dimensionality reduction methods can be critical when building a Markov model as they can allow the states to be placed on much better configurational space regions improving state definitions and thus the MSM quality. PCA is a technique that is widely used for dimensionality reduction of simulation data, however in the last years with the introduction of TICA and its demonstrated success in various systems, TICA has become the preferred method. Due to the resurgence of neural networks and auto encoders, which are unsupervised neural networks that can encode data in a lower dimensional space, we decided to apply them on simulation data and compare them to the other methods. Additionally, a variant of the k-means clustering technique which was shown to give good results in problems of dimensionality reduction [108] was tested as well. The four projection methods were then compared based on how good the MSM constructed on their projected data was able to estimate the slowest implied timescale of the system.

By testing the four dimensionality reduction methods on two simulated systems we find that it is difficult to choose a single best method and settings. Different methods seem to perform better or worse depending on the system, the amount of data and the number of dimensions. Here we also highlight the problem of deciding the number of dimensions to which to project, for which there is currently no satisfactory solution and manual testing has to be done by the user.

Dimensionality reduction methods for molecular simulations

S. Doerr,[†,§] I. Ariz,[†,§] M. J. Harvey,[‡] and G. De Fabritiis[*,¶]

†*Computational Biophysics Laboratory (GRIB-IMIM), Universitat Pompeu Fabra, Barcelona Biomedical Research Park (PRBB), C/ Doctor Aiguader 88, 08003 Barcelona, Spain*

‡*Acellera, Barcelona Biomedical Research Park (PRBB), C/ Doctor Aiguader 88, 08003 Barcelona, Spain*

¶*Institució Catalana de Recerca i Estudis Avançats (ICREA), Passeig Lluis Companys 23, Barcelona 08010, Spain*

§*Contributed equally to this work*

E-mail: gianni.defabritiis@upf.edu

Abstract

Molecular simulations produce very high-dimensional data-sets with millions of data points. As analysis methods are often unable to cope with so many dimensions, it is common to use dimensionality reduction and clustering methods to reach a reduced representation of the data. Yet these methods often fail to capture the most important features necessary for the construction of a Markov model. Here we demonstrate the results of various dimensionality reduction methods on two simulation data-sets, one of protein folding and another of protein-ligand binding. The methods tested include a k-means clustering variant, a non-linear auto encoder, principal component analysis and tICA. The dimension-reduced data is then used to estimate the implied timescales of the slowest process by a Markov state model analysis to assess the quality of the projection. The projected dimensions learned from the data are visualized to demonstrate which conformations the various methods choose to represent the molecular process.

1 Introduction

Molecular dynamics (MD) simulations allow one to simulate bio-molecules with increasingly good accuracy and in recent years have begun to provide meaningful predictions of experiments and insight into atomistic mechanisms, like the process of protein folding into native structures.[1] From the computational point of view, one of the primary challenges of MD simulations is the ability to sample experimentally relevant millisecond to second timescales. With the advent of general-purpose graphics processing units in 2009,[2] it has become possible to produce microseconds, and more recently milliseconds, of aggregated simulation data. This data is high dimensional with a common system size being of the order of ten to hundred thousand dimensions. The results are often analyzed using Markov state models (MSMs).[3] Discrete Markov state models require the definition of discrete states which are usually computed by clustering over a metric space. Depending on the metric used and the dimensionality of the space, the clustering might produce a poor discretization of states, hiding the slow dynamics and yielding a poor MSM from which it is impossible to compute the correct thermodynamic

variables.[3] As a consequence, it is important to use a proper metric space for each system and a proper discretization, i.e. one that captures the most relevant information about the simulated molecular process.

Choosing the most favorable reduced metric space for a system is difficult without *a priori* information, and clustering over high dimensional spaces can be very challenging.[4] In recent years, new algorithms that can learn complex functions have lead to methods which produce a lower dimensional representation of the data that have no significant loss of information.[5] Sparse coding,[6] auto encoders[7] and neighborhood embedding[8] have shown to be very effective in reducing the dimensionality of data while preserving important underlying features. Dimensionality reduction methods have also been developed specifically for molecular dynamics data by reweighing features with unsupervised methods,[9] by learning distance functions[10] and by using diffusion maps.[11]

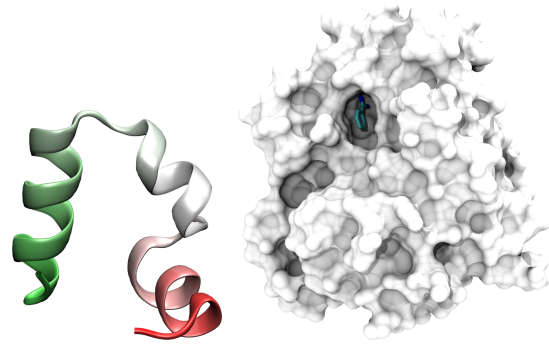
In this work we focus on comparing the performance of dimensionality reduction methods on biological simulation data. We resolve the folding of a protein and the binding of a ligand to a protein by simulation and try to find the projection that produces the best MSM using non-linear auto encoders, clustering and linear projection methods such as PCA and tICA.[12]

2 Methods

2.1 Data-sets

The data-sets used are from the folding and unfolding simulations of Villin as well as the ligand-binding simulations of Benzamidine to Trypsin.

Villin (see folded structure in Fig. 1a) is a tissue-specific protein which binds to actin. The part under study is a double norleucin mutant of the 35 amino acid long headpiece widely tested in MD simulations because of its fast folding properties. At the temperature of $300°K$ the non mutated protein domain has an experimental folding time of $4.3\mu s$.[13,14] Computational estimations of the double mutant at



(a) Villin (b) Benzamidine-Trypsin

Figure 1: a) folded structure of Villin. b) bound configuration of Benzamidine-Trypsin.

$360°K$ gave a folding time of $3.2\mu s$,[15] a folding free energy of -0.6 kcal/mol and a timescale of the order of $200ns$ and will be used as reference, as the same setup will be used here.

Benzamidine-Trypsin is a protein-ligand binding system, with an experimental free energy of $-6.2kcal/mol$[16] at $300°K$ and a timescale of the binding process of the order of $600ns$.

The structure of Villin was taken from Piana et. al.,[15] solvated in water and simulated using the CHARMM22* forcefield.[17] The Benzamidin-Trypsin setup was taken from Buch et. al.,[18] solvated and simulated using the AMBER 99SB force field.[19] Simulations were performed using ACEMD,[2] a molecular dynamics code for graphical processing units, on the GPUGRID distributed computing infrastructure.[20]

For Villin, 1562 simulations were used, each $120ns$ long, resulting in an aggregate simulation time of $187.4\mu s$ and 1,874,400 conformations at a sampling time of $0.1ns$. For Benzamidine-Trypsin, 488 simulations of $100ns$ were used for a total aggregate simulation time of $48.8\mu s$ and 488.000 configurations. To best demonstrate the performance of the dimensionality reduction methods in scarce-data regimes which are the norm in MD simulations, we bootstrapped the data-sets 20 times at various percentages of the total data-set, thus obtaining various subsampled data-sets at 20-100% of the total sim-

ulation data.

2.2 Preprocessing of the data

During simulations, the configuration of the system is represented by the positions and velocities of all atoms. For analysis purposes, however, a translation and rotation invariant representation is ideal. Therefore, for Villin we calculate and use the protein contact maps of the conformations and for Benzamidine-Trypsin we use the ligand-protein contact maps.

For Villin, the contact maps were produced from the original trajectories by calculating the distance between the backbone C_α of each amino acid to the C_α atoms of all other amino-acids. Each element of the resulting distance matrix was transformed into 1 if the distance was below 8Å and 0 otherwise. As contact maps are symmetric, only the upper triangular part of the matrix was considered. The upper triangular part was then expanded into a vector of $\frac{n_{res}(n_{res}-1)}{2}$ contacts. For Villin with 35 residues, this results in 595-element binary-valued contact maps. The contact map data-set of Villin is on average 80% sparse and fewer than 1% of the contact maps are duplicates.

For Benzamidine-Trypsin, the protein-ligand contact maps were produced by calculating the distance between the C_α atoms of the residues of Trypsin and two carbon atoms at opposite sides of the Benzamidine (as show in the SI of Doerr et al.[21]). The distances were then thresholded similarly to 8Å as in Villin to produce contacts, however in this case, the contacts are a one-dimensional vector of 446 contacts (2 ligand atoms times 223 protein residues). The contact maps of Benzamidin-Trypsin are on average 99% sparse.

2.3 Dimensionality reduction methods

The data-sets have proven challenging to analyze using standard clustering methods like k-means, k-centers, and others. In particular the folded state of Villin is not easily detected and therefore an MSM built on top of such clustering would lose any information on the folding process. A cause could be the high dimensionality of the data which can spread out clusters which exist on subspaces. A projection of the data on a lower dimensional space can lead to an improvement in the clustering and MSM constructed on top of it. In this work, four different methods are used for learning the features of a lower dimensional representation: a modification of k-means,[22] principal component analysis (PCA), a non-linear auto encoder and tICA.[12] The motivation for this choice is that k-means is an unsupervised method commonly used for clustering bio-molecular data; PCA is an optimal projection method in the linear regime; tICA, another linear method, extends the idea of PCA by using the time component of the simulations and auto encoders are a good extension to the non-linear regime when a sigmoid function is used as the activation function. Auto encoders are also known to learn PCA under certain conditions,[23] i.e. linear activation function and transposed weights between encoding and decoding. A fifth method, called t-SNE[8] was also considered due to its recent impressive success on various data-sets like the MNIST, NORB and NIPS as well as the Merck Viz Challenge. However, due to its high computational cost and memory requirements, we were not able to test it on our data-set.

2.4 k-means triangle

The k-means (triangle) method was taken from Coates et al.[22] Normal k-means clustering produces a hard assignment of each data point to a single cluster it corresponds to and can be represented by a binary $1 \times K$ vector (where K the number of clusters), which is 1 on the index of the closest cluster center and 0 elsewhere. k-means (triangle) on the other hand, after computing the cluster centers, represents each data point x as a $1 \times K$ dimensional vector v_x whose elements are calculated by

$$v_x(i) = max\{0, \mu(z) - z_i\}$$

where $c^{(i)}$ is the i-th cluster center, $z_i = \|x - c^{(i)}\|$ is the distance of data point x from cluster center $c^{(i)}$ and $\mu(z)$ is the mean of all z_i of x. In

other words, each data point gets represented by a vector of its distances to all cluster centers subtracted by their mean and thresholded at a minimum 0. This method proved superior in Coates et al.[22] compared to normal k-means clustering and several other methods. In this study k-means (triangle) was used to project the contact map data into the $1 \times K$ space defined by the K cluster centers.

2.5 PCA

Principal component analysis is one of the most widely used dimensionality reduction methods. By calculating the eigenvectors of the data covariance matrix, PCA can project the data on the principal components which are the dimensions of largest variance of the data-set, thereby minimizing the total squared reconstruction error of the projected data through a linear transformation of the input data. It enjoys wide application in the field of computational biology, implementations exist for most programming languages and it has a quick runtime. In this study we used PCA to project the contact map data onto the first principal components of PCA.

2.6 tICA

Time-lagged independent component analysis is a dimensionality reduction method recently rediscovered and applied very successfully to biological problems.[12,24] The reason for its great success in such problems is that the tICA projections are the linear transform of the input data which maximizes the auto-correlations of the output data. This means that it is able to identify and project the data on the slowest subspace which can be obtained through a linear transform. As the biologically most interesting processes in simulations are often transitions between metastable states separated by large barriers, tICA is able to project the data onto those slow processes and thus allows a finer discretization of the slow dynamics without losing information related to those slow processes. In this study we used tICA to project the contact map data on the first time-lagged independent components.

2.7 Auto encoder

An auto encoder is a neural network which tries to reconstruct a given input vector in its output layer after encoding it in one or more hidden layers. Therefore, input and output layers of auto encoders have the same number of units while the hidden layers often contain fewer units than the input and output, thus forcing the neural network to learn a lower dimensional representation of the data. The activation of an auto encoder unit is defined by

$$a_i^{(L+1)} = f\left(\sum_{j=1}^{S_L} W_{ij}^{(L)} a_j^{(L)} + b_i^{(L)}\right) \qquad (1)$$

where $a_i^{(L)}$ the i-th unit in layer L, S_L the number of units in layer L, $W_{ij}^{(L)}$ the weight matrix of layer L, $b_j^{(L)}$ the j-th bias term and f is the activation function. Various activation functions can be used in an auto encoder, however as we as we want to test a non-linear auto encoder, we choose to use a *sigmoid* function $f(z) = 1/(1+e^{-z})$. Additionally a sigmoid output layer aids us in mapping the reconstructed data to contact maps as its output values are within the range $[0, 1]$.

The most common optimization algorithm for auto encoders is gradient descent through the back propagation algorithm. Nevertheless, more elaborate algorithms have been used such as the conjugate gradient, or the Hessian-free algorithm used by,[25] which is a 2^{nd}-order optimization algorithm. Among these, the *L-BFGS* algorithm explained by[26] has been shown by[27] to be among the most efficient ones and was used here. For the purposes of this study, we built various shallow (single hidden layer) auto encoders in Theano.[28] The configuration for the 5-dimensional auto encoder can be seen in Figure 2. After forward propagating the examples through the auto encoder, a cost function, given in equation (3), is evaluated and the gradients for each layer are calculated by back-propagation.

The *L-BFGS* algorithm is then used for train-

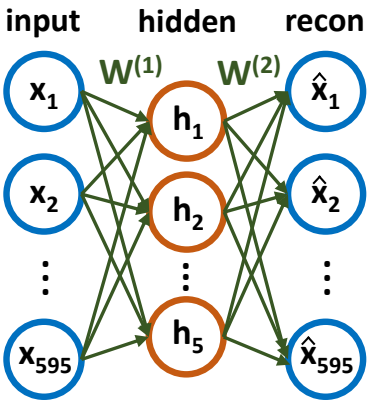


Figure 2: Auto encoder architecture for a 5 dimensional projection of the Villin data. x represents the input data in the input layer which consists of 595 units, $\hat{x}$ represents the reconstructed output in the output layer with 595 units and h represents the data representation in the hidden layer consisting of various number of hidden units depending on the auto encoder. W denotes the weights applied to the activation of the layers before them.

ing over 400 epochs. After training, the projected simulation data is obtained by removing the output layer of the auto encoder and taking the lower dimensional representation produced by the hidden layer for each simulation frame.

2.8 Markov state model

MSMs have been used to reconstruct equilibrium and kinetic properties in many molecular systems.[18,21,29] MSMs allow to extrapolate equilibrium properties of a dynamical system like MD, by many out-of-equilibrium trajectories. The trajectories have first to be discretized by assigning each frame to a given state. In this study, the projected data frames of each projection method were clustered and assigned to the closest of 1000 states produced by the mini-batch k-means algorithm of Scikit-learn,[30] thus producing discretized trajectories.

Using the discretized trajectories, a master equation can then be constructed by determining the frequency of transitions between states,

$$\frac{dP_i(t)}{dt} = \sum_{j=1}^{N} [k_{ij}P_j(t) - k_{ji}P_i(t)] \qquad (2)$$

where $P_i(t)$ is the probability of state i at time t, and k_{ij} are the transition rates from j to i. The master equation (Eq. 2) can be rewritten in a compact matrix form $d\mathbf{P}/dt = \mathbf{KP}$ where $K_{ij} = k_{ij}$ for $i \neq j$ and $K_{ii} = -\sum_{j \neq i} k_{ji}$. The formal solution is $\mathbf{P}(t) = \mathbf{T}\mathbf{P}(0)$ where $\mathbf{T} = p(i, t|j, 0)$ is the probability of being in state i at time t, given that the system was in state j at time 0. The transition matrix $\mathbf{T}$ is estimated from the simulation trajectory by counting how many transitions are observed between i and j and vice-versa and using a reversible maximum-probability estimator.[3]

From the the matrix T all the thermodynamics and kinetics properties of the system can be determined as well as a kinetic lumping of clusters using the PCCA+ method.[31] The implied timescale of the slowest process, which we will focus on in this study, can be calculated from the second eigenvalue of the transition probability matrix T as $t = \frac{\tau}{ln(\lambda(\tau))}$, where t the slowest timescale, τ is the lag-time at which the Markov model is constructed and $\lambda(\tau)$ the second eigenvalue of the transition probability matrix of the Markov model.

3 Results

We projected the Villin and Benzamidine-Trypsin data-sets with the four dimensionality reduction methods and analyzed the projected data using Markov models. To demonstrate the performance of the methods under scarce data conditions we calculated Markov models containing decreasing amounts of simulations and to reduce the effect of individual trajectories on the result, we bootstrapped the simulations used in the model over 20 iterations. From the 20 bootstrapped Markov models we calculate the slowest implied timescale over a range of lag-times. A converged Markov model should have the slowest timescale converged over a range of lag-times and thus there should be a

low standard deviation to the timescales calculated. Convergence of timescales is important for Markov models as it is an indication of Markovinianity of the model and is required for calculating consistent eigenvalues and eigenvectors and thus all other observables.

3.1 Benzamidine-Trypsin

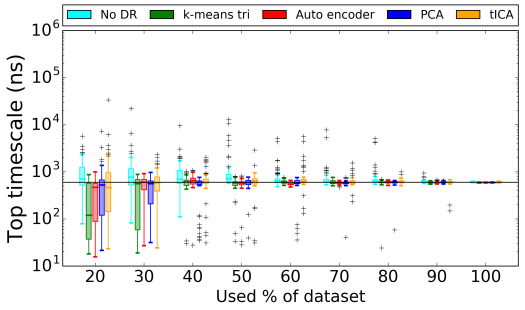


Figure 3: Top implied timescales for Markov models built for Benzamidine-Trypsin using 5 dimensional projections. Timescales were estimated at lag-times of 5 to 15ns. The black horizontal line indicates the reference timescale of 600ns.

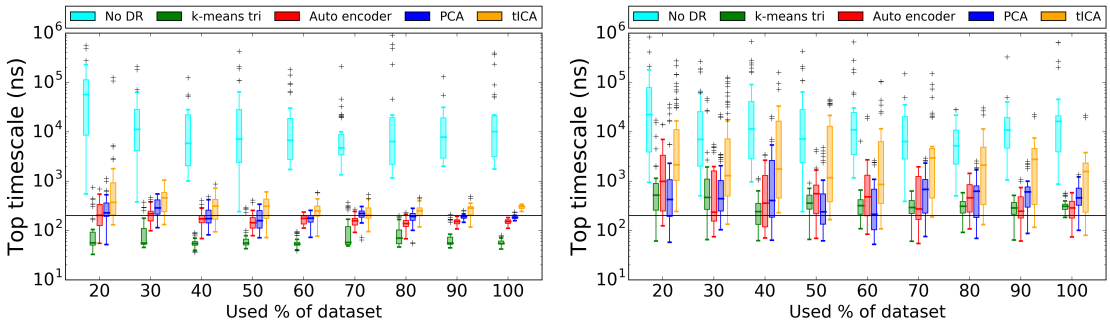
In Figure 3 we can see the performance of the dimensionality reduction methods reflected in the implied timescales of the Benzamidine-Trypsin data-set. We can see that on this data-set, even without dimensionality reduction we are able to obtain the correct timescale, with all methods showing very small errors, when using more than 50% of the dataset. We should note however, that increasing the range of lag-times as in SI Figure S1 to 40ns, we see that the full dimensional data, tICA and PCA, perform worse. In general this is not a big issue as Markov models are typically constructed at the shortest lag-time at which convergence is seen (in this case 10ns). However, it shows us that at larger lag-times, the slow process can become lost in the three aforementioned methods. Interestingly, k-means (triangle) and the auto encoder are not affected by this and keep consistently converged timescales over large lag-times. Therefore, dimensionality reduction methods can help in this case with keeping the timescales flat over larger lag-times.

Changing the number of dimensions on the other hand does not have a significant effect on Benzamidine-Trypsin. Results are shown for 50 dimensions in SI Figure S2 without any notable changes, indicating that the dimensionality reduction methods at the very least do not produce a worse projection than the starting data.

3.2 Villin

For Villin in Figure 4 it can be seen that using the full dimensional data is not an option as it overestimates the timescale by at least two orders of magnitude with huge uncertainty reflected in the error bars. This makes this system much more challenging than Benzamidine-Trypsin and stresses the importance of dimensionality reduction. In Figure 4a which shows the 5-dimensional projections; out of the four projection methods, PCA, tICA and the auto encoder dominate, estimating the timescale for Villin closest to the reference timescale, with k-means (triangle) underestimating the timescale by a factor of around 3. However, by increasing the number of projected dimensions to 20 as in Figure 4b, we see that with 20 dimensions the estimation errors of the timescales are much larger, with k-means (triangle), PCA and the auto encoder performing best.

Further investigating this, we tested at the 50% data-set various numbers of dimensions. The results can be seen in Figure 5, which shows that for Villin the number of projected dimensions is critical for the construction of a working Markov model. The best performance is obtained by low-dimensional tICA and the auto encoder, however, increasing dimensions gives increasingly wrong timescales for tICA. Especially so when compared to k-means (triangle), PCA and the auto encoder which are not as strongly affected and are more stable over varying dimensionalities. These results are consistent with[9] which shows that tICA is prone to larger errors when increasing dimensionality than other methods.



(a) 5 dimensions

(b) 20 dimensions

Figure 4: Top implied timescales for Markov models built for Villin using 5 and 20 dimensional projections. Timescales were estimated at lag-times from 25 to 30ns. The black horizontal line indicates the reference timescale of 200ns.

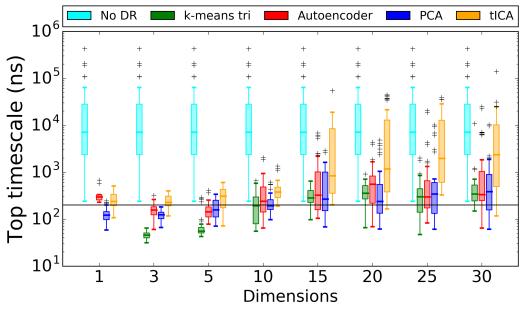
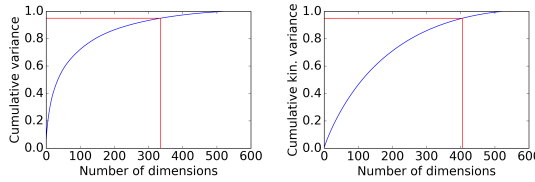


Figure 5: Top implied timescales for Villin using a varying number of dimensions on the 50% data-set. Timescales were estimated at lag-times of 25 to 30ns. The black horizontal line indicates the reference timescale of 200ns.

3.3 Automated dimensionality detection

As mentioned before, clustering methods have a hard time detecting clusters in high-dimensional spaces and the example of Villin consist further proof. A problem that arises from this, is the detection of the number of dimensions on which to project. As it strongly depends on the data-set in use, an automated method would be ideal to avoid having the user manually test multiple dimensions. Methods such as PCA and tICA, are able to calculate the percentage of variance described by the first N principal and independent components and this is often used to calculate the ideal number of dimensions on which to project. Therefore, the problem of number of dimensions can be reinterpreted as deciding a specific variance percentage to keep when projecting. Typical heuristics include using the first N dimensions which contain 95% of the variance. However, as can be seen in Figure 6, this would produce a large number of dimensions (between 300 and 400 dimensions) for both PCA and tICA, failing to produce a functioning Markov model. Therefore, to our knowledge, there is currently no automated method for dimensionality detection that would be able to produce a functioning Markov model for the Villin data-set.



(a) PCA

(b) tICA

Figure 6: Cumulative variance encoded in the PCA principal components and tICA independent components on the Villin data-set. Red lines show the 95% variance cutoff.

3.4 Learned structural features

To understand better how the dimensionality reduction methods work, it is interesting to visualize the features that the dimensionality reduction methods have learned. As we use protein contact maps for Villin, we are able to represent the weights, principal components, cluster centers and independent components respectively as two-dimensional images. We can see the different features that were learned by k-means (triangle) (Figure 7), the auto encoder (Figure 8), PCA (Figure 9) and tICA (Figure 10). Red is used for positive weights, white for values close to zero and blue for negative weights. Under the weight maps, we show the protein conformations that most strongly represent those maps. For k-means (tri) we show the conformation of the centroids of the clusters. For the other methods, as they have negative and positive weights, we show in each column the two conformations that correspond to the maximum and minimum value along that dimension. All methods learned interesting features, both local and global. Features with strong positive or negative values close to the matrix diagonal encode local protein secondary structures; in this case alpha helices. On the other hand, features further from the diagonal encode more distant (global) residue interactions, and features perpendicular to the diagonal encode for anti-parallel beta strands which are a relatively common occurrence during simulations of Villin. The fully folded conformation of Villin consists of 3 folded helices (see Fig. 1a), therefore the folded conformation is represented by contact maps similar to the 1st cluster center in Figure 7, the first auto encoder hidden unit in Figure 8, the second principal component in Figure 9 and the second independent component in Figure 10.

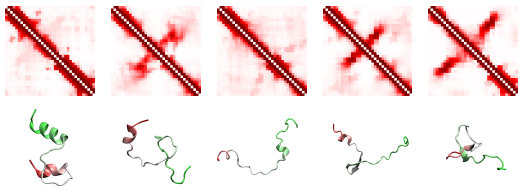


Figure 7: k-means (tri) cluster center contact maps on first row and centroid structures on second row.

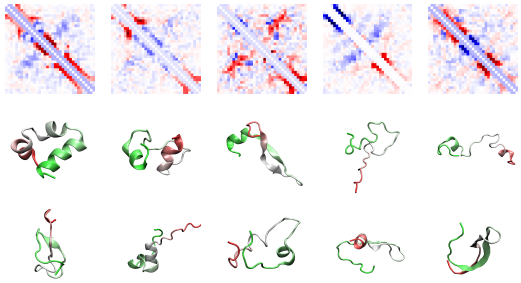


Figure 8: Auto encoder weights on first row and structures which maximally and minimally activate each hidden layer neuron on second and third row.

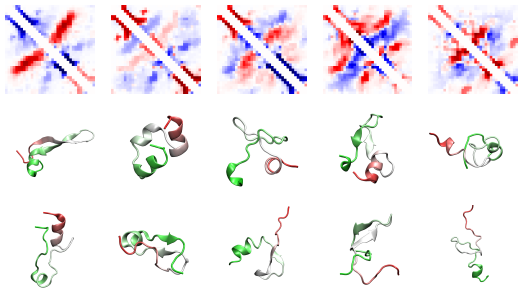


Figure 9: PCA principal components on first row and structures which correspond to the maximum and minimum values in each PC on second and third row.

4 Conclusions

In this paper we have used four methods for dimensionality reduction over two high-dimensional data-sets of protein folding and ligand binding trajectories. Benzamidine-Trypsin proved to be a trivial case in which dimensionality reduction is not necessary but can help improve the timescales for larger lag-times. On the other hand Villin proved much more complicated, where building a Markov model on the pure high-dimensional data was not able to reliably separate the underlying states to produce the correct kinetic quantities. However, a shal-

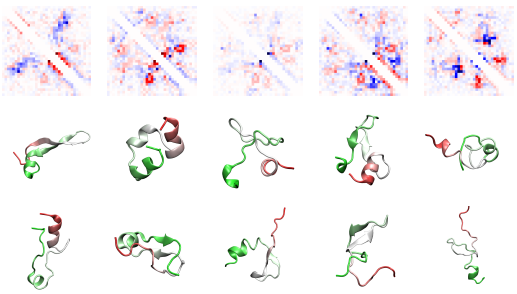


Figure 10: tICA independent components on first row and structures which correspond to the maximum and minimum values in each IC on second and third row.

low auto encoder, a modified k-means featurization method, PCA and tICA were capable of improving the Markov model significantly. TICA, PCA and the auto encoder provided the best performance, however tICA proved to be very sensitive to the number of used dimensions. Indeed all methods were affected by the increase of dimensionality, albeit much less than tICA, indicating that the number of dimensions can be more important for MSM construction than the choice of projection method.

Both tICA and the auto encoder perform well on Villin using very low-dimensional projections, however, this could hide some slow dynamics in more complicated systems. An important remaining problem is the choice of dimensionality. Heuristics used for automatically detecting the best number of dimensions in PCA and tICA, such as the variance encoded in the first components fail to give good results in the case of a Markov model analysis. Therefore, it remains an open question as to how best detect the dimensionality required to analyze the system and currently manual testing needs to be done for each system to determine the dimensionality that gives the most consistent results.

Another factor that should to be taken into account when comparing projection methods is their run-time, as analysis of simulations can become computationally expensive for large data-sets. In this aspect k-means (triangle) is faster than the other methods, projecting the full Villin data-set on 10 dimensions in tens of seconds compared to tICA taking around 1 minute to calculate the ICs and projections, PCA 2.5 minutes and auto encoders on GPUs around 40 minutes to train and project the data.

As the first application of auto encoders on the dimensionality reduction of contact map data, we believe that they provided interesting results, even reaching the performance of other established methods. However, we believe there is more potential than is demonstrated here, as we were not yet able to exhaustively test different configurations of the networks. The large number of options in the construction of an auto encoder as well as the number of free parameters, allow for a variety of tuning and different setups. As auto encoders can go beyond the linearity of the other methods, we believe that they can show increased potential in other configurations and deeper architectures. Additionally, as auto encoders can learn local structural features they can become generalized and could potentially be applied to different data-sets of the same type (i.e. folding of different proteins) without the need of retraining.

5 Appendix

$$J(W,b) = \left[\frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}\left\|\hat{x}_i - x_i\right\|^2\right)\right] + \frac{\lambda}{2}\sum_{l=1}^{2}\sum_{i=1}^{s_l}\sum_{j=1}^{s_l+1}\left(W_{ij}^{(l)}\right)^2 + \beta\sum_{j=1}^{s_2} KL(p||\hat{p}_j) \quad (3)$$

where m the number of training examples, x_i training example i, $\hat{x}_i$ the reconstruction of x_i in the last layer of the auto encoder, λ the weight decay parameter, s_l the number of units in layer l, $W^{(l)}$ the weight matrix of layer l, β the sparsity penalty weight and $KL(p||\hat{p}_j)$ the Kullback-Leibler (KL) divergence between p the desired sparsity of the hidden units and $\hat{p}_j$ the mean sparsity of hidden unit j over all

training data. In our setup we used $\lambda = 0.003$, $p = 0$ and $\beta = 3$.

Acknowledgement GDF aknowledges support from MINECO (BIO2014-53095-P) and FEDER. We thank the volunteers of GPUGRID for donating their computing time.

Supporting Information Available: This material is available free of charge via the Internet at `http://pubs.acs.org/`.

References

(1) Lindorff-Larsen, K.; Piana, S.; Dror, R. O.; Shaw, D. E. *Science* **2011**, *334*, 517–520.

(2) Harvey, M. J.; Giupponi, G.; De Fabritiis, G. *J. Chem. Theory Comput.* **2009**, *5*, 1632–1639.

(3) Prinz, J.-H.; Wu, H.; Sarich, M.; Keller, B.; Senne, M.; Held, M.; Chodera, J. D.; Schütte, C.; Noé, F. *J. Chem. Phys.* **2011**, *134*, 174105–174105–23.

(4) Kriegel, H.-P.; Kröger, P.; Zimek, A. *ACM Trans. Knowl. Discov. Data* **2009**, *3*, 1:1–1:58.

(5) Wang, J.; He, H.; Prokhorov, D. V. *Procedia Computer Science* **2012**, *13*, 120–127.

(6) Olshausen, B. A.; Field, D. J. *Nature* **1996**, *381*, 607–609.

(7) Hinton, G. E.; Salakhutdinov, R. R. *Science* **2006**, *313*, 504–507.

(8) van der Maaten, L.; Hinton, G. **2008**,

(9) Blöchliger, N.; Caflisch, A.; Vitalis, A. *J. Chem. Theory Comput.* **2015**, *11*, 5481–5492.

(10) McGibbon, R. T.; Pande, V. S. *J. Chem. Theory Comput.* **2013**, *9*, 2900–2906.

(11) Boninsegna, L.; Gobbo, G.; Noé, F.; Clementi, C. *J. Chem. Theory Comput.* **2015**, *11*, 5947–5960.

(12) Pérez-Hernández, G.; Paul, F.; Giorgino, T.; De Fabritiis, G.; Noé, F. *J. Chem. Phys.* **2013**, *139*, 015102.

(13) Kubelka, J.; Eaton, W. A.; Hofrichter, J. *J. Mol. Biol.* **2003**, *329*, 625–630.

(14) Kubelka, J.; Chiu, T. K.; Davies, D. R.; Eaton, W. A.; Hofrichter, J. *Journal of Molecular Biology* **2006**, *359*, 546–553.

(15) Piana, S.; Lindorff-Larsen, K.; Shaw, D. E. *Proc. Natl. Acad. Sci. U.S.A.* **2012**, *109*, 17845–17850.

(16) Mares-Guia, M.; Shaw, E. *J. Biol. Chem.* **1965**, *240*, 1579–1585.

(17) Lindorff-Larsen, K.; Maragakis, P.; Piana, S.; Eastwood, M. P.; Dror, R. O.; Shaw, D. E. *PLoS ONE* **2012**, *7*, e32131.

(18) Buch, I.; Giorgino, T.; Fabritiis, G. D. *PNAS* **2011**,

(19) Hornak, V.; Abel, R.; Okur, A.; Strockbine, B.; Roitberg, A.; Simmerling, C. *Proteins* **2006**, *65*, 712–725.

(20) Buch, I.; Harvey, M. J.; Giorgino, T.; Anderson, D. P.; De Fabritiis, G. *J. Chem. Inf. Model.* **2010**, *50*, 397–403.

(21) Doerr, S.; De Fabritiis, G. *J. Chem. Theory Comput.* **2014**, *10*, 2064–2069.

(22) Coates, A.; Ng, A. Y.; Lee, H. An analysis of single-layer networks in unsupervised feature learning. International Conference on Artificial Intelligence and Statistics. 2011; pp 215–223.

(23) Bourlard, H.; Kamp, Y. *Biol. Cybern.* **1988**, *59*, 291–294.

(24) Schwantes, C. R.; Pande, V. S. *J. Chem. Theory Comput.* **2013**, *9*, 2000–2009.

(25) Martens, J. Deep learning via Hessian-free optimization. Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel. 2010; pp 735–742.

(26) Liu, D. C.; Nocedal, J. *Mathematical Programming* **1989**, *45*, 503–528.

(27) Le, Q.; Ngiam, J.; Coates, A.; Lahiri, A.; Prochnow, B.; Ng, A. On optimization methods for deep learning. Proceedings of the 28th International Conference on Machine Learning (ICML-11). New York, NY, USA, 2011; pp 265–272.

(28) Bergstra, J.; Breuleux, O.; Bastien, F.; Lamblin, P.; Pascanu, R.; Desjardins, G.; Turian, J.; Warde-Farley, D.; Bengio, Y. Theano: a CPU and GPU Math Expression Compiler. Proceedings of the Python for Scientific Computing Conference (SciPy). Austin, TX, 2010; Oral Presentation.

(29) Bisignano, P.; Doerr, S.; Harvey, M. J.; Favia, A. D.; Cavalli, A.; De Fabritiis, G. *J Chem Inf Model* **2014**, *54*, 362–366.

(30) 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, E. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.

(31) Cordes, F.; Weber, M.; Schmidt-Ehrenberg, J. *Metastable Conformations via successive Perron-Cluster Cluster Analysis of dihedrals*; 2002.

Dimensionality reduction methods for molecular simulations

S. Doerr,[1] I. Ariz,[1] M. J. Harvey,[2] and G. De Fabritiis[3,1,*]

[1] *Computational Biophysics Laboratory (GRIB-IMIM), Universitat Pompeu Fabra, Barcelona Biomedical Research Park (PRBB), C/ Doctor Aiguader 88, 08003 Barcelona, Spain*[†]

[2] *Acellera, Barcelona Biomedical Research Park (PRBB), C/ Doctor Aiguader 88, 08003 Barcelona, Spain*

[3] *Institució Catalana de Recerca i Estudis Avançats (ICREA), Passeig Lluis Companys 23, Barcelona 08010, Spain*

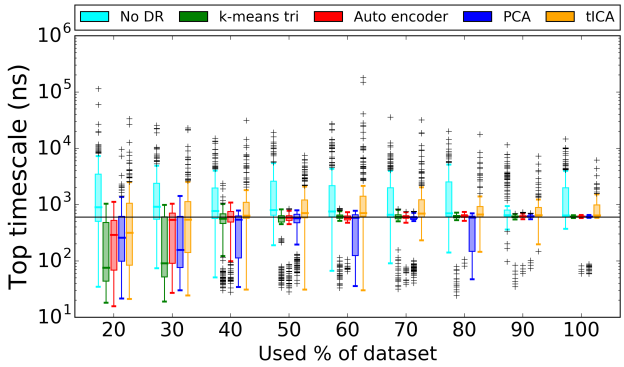


FIG. S1: Top implied timescales for Markov models built for Benzamidine-Trypsin using **5** dimensional projections. Timescales were estimated at lag-times of **5 to 40ns**. The black horizontal line indicates the reference timescale of $600ns$.

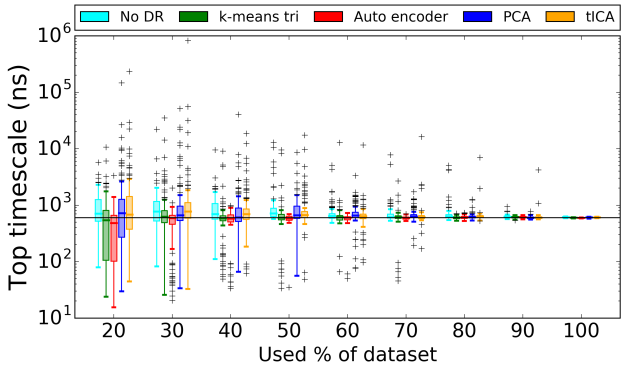


FIG. S2: Top implied timescales for Markov models built for Benzamidine-Trypsin using **50** dimensional projections. Timescales were estimated at lag-times of **5 to 15ns**. The black horizontal line indicates the reference timescale of $600ns$.

* Electronic address: gianni.defabritiis@upf.edu

† Contributed equally to this work

Chapter 5

DISCUSSION

In the work presented in this thesis, we highlighted the applications of high-throughput MD simulations as well as the importance and possibilities of learning from them. Slower processes than ever before can be resolved using MD in combination with adaptive sampling and Markov state models, and the support provided by a unified framework on which to perform these investigations magnifies the potential of the field. Here we discuss the implications and future prospects of the previously presented work.

Analyzing high-throughput MD

High-throughput MD and Markov state models provide a very strong framework for investigating metastable states and kinetics as demonstrated on the case of AmpC β-Lactamase and carboxy-thiophene. This study provided a good starting point for the thesis. It demonstrated the weaknesses of crystallographic methods, through misleading binding poses as well as the significance of the fact that crystallography is a static method, which in the case of the oxyanion hole represented a wide configurational minimum in the form of a single configuration. It also demonstrated the power of MD but also the pitfalls involved in the construction of Markov state models. Markov state models can be very accurate at estimating

the slowest process of the system and the two metastable minima it connects. Investigating however the second, third etc. slow processes related to other metastable minima can prove tricky. Additionally, undersampled states can give misleading kinetics and require caution and experience for the scientist to notice the underlying issue. What initially looked like a deep binding pocket with an order of magnitude stronger binding than the tunnel pose, turned out to be caused by a lack of sampling of the unbinding event in the Markov model. The lack of sampling of unbinding transitions, in turn, forced us to perform a manual adaptive sampling, where new simulations were restarted from the deep pocket pose to generate unbinding events and thus finally obtain a converged estimate of kinetics from the Markov model. Improving the sampling in those regions and thus allowing the discovery and correct description of further metastable states calls for the introduction of a new sampling methodology as manual adaptive sampling is not feasible on the large scale of high-throughput MD studies.

Another outstanding issue of Markov state models are the various parameters and options that must be tuned by their users. Good or bad choices of parameters can make the difference between a working model and one that is unable to resolve any valid information. Various efforts have been made to automatize Markov model construction [109, 110, 111], without however producing an established method. Another issue with Markov models comes from the configurational space discretization. As shown in this work, dimensionality reduction methods can improve greatly the quality of Markov models, they do however leave at least one more parameter up for tuning which is the number of dimensions. Currently, no automatic way of determining these has been shown to work well.

Intelligent sampling

The sampling methodology presented here for solving the problem of under-sampled metastable minima and more generally speeding up the

exploration of the phase space is adaptive sampling. It allows faster sampling of the phase space of up to an order of magnitude as demonstrated in our two publications [95, 97]. Adaptive sampling needs to be flexible enough to be applied to various biological systems and without prior knowledge of the system. Even though the increase in speed by the adaptive method implemented in our first adaptive work [95] was substantial, further research showed that is was not generally applicable on various other biological systems. Benzamidine-Trypsin proved to be a system in which most metastable minima connect with very low barriers to the crystal bound pose. Therefore, an adaptive scheme based on the highest unbinding time was able to follow the gradient of the funnel very quickly into the binding pocket. Such a scheme however would not work on systems with deep metastable poses, well separated from the global minimum, from which the system would have to escape. Indeed applications on systems of protein folding proved to not provide a great advantage over naive sampling. Therefore, in our second work demonstrating adaptive sampling in HTMD [97] we moved from the specialized methodology of the first work to a more generalized adaptive sampling methodology which worked for both protein-ligand binding and protein folding systems as it works by sampling the least sampled metastable states of the system, thus allowing for faster barrier-crossing.

As can be expected, introducing experimental knowledge on the system as in [96] can provide another order of magnitude speedup over a generalized population-based sampling method while losing some generality. Future work is focused on using predicted information on the system such as docked poses or predicted structural information for the folded conformation to guide adaptive sampling without the need for experimental information. Additionally, there is currently no theoretically proven optimal generalized methodology for adaptive sampling of the phase space and it remains an open question.

Infrastructure for supporting molecular discovery

The HTMD software developed by us for aiding molecular discovery through simulations has shown great applicability inside our lab and is already starting to be used outside of it as well. As a framework that tries to provide all the functionality for molecular discovery, it can fall into the usual pitfalls of academic software. Duplicating existing functionality, not reaching a wider audience and thus becoming abandoned academic software being major issues. On the side of functionality duplication we try to minimize it as much as possible through collaborations with other groups developing related software. This is also why HTMD has made a switch from Matlab to Python since a large part of scientific software is being currently developed for Python, allowing for direct integration of established projects. This switch allowed HTMD for example to use the pyEMMA [102] software under the hood for Markov model construction increasing its power, stability and reducing development effort. The remaining two problems of development and audience are connected and we hope that by reaching a wider audience and through more collaborations inside academia as well as the industry, the software will stay alive and actively developed.

Chapter 6

CONCLUSIONS

1. High-throughput MD is shown to be a powerful tool for investigating the slow processes of biological systems. In conjunction with Markov state models it is able to resolve the kinetics between multiple metastable states, detect previously unseen binding pockets and conformational changes and provide hints on misleading experimental information.

2. Adaptive sampling is demonstrated to provide a superior sampling mode for high-throughput molecular simulations. An order of magnitude less required simulation time to achieve the same observables without biasing justifies making a generalized adaptive sampling scheme the default sampling mode for modern MD.

3. Configurational space discretization is an important challenge in the construction of Markov state models. Various dimensionality reduction methods are demonstrated to be able to improve the models, while introducing auto encoders to the field of simulation dimensionality reduction. An open issue remains the ideal number of dimensions which as shown can be of great importance in the construction of Markov models.

4. A new software framework, which provides all required tools for molecular discovery using MD simulations is presented. The HTMD

software allows increased control over experiments, increasing reproducibility and allowing scientists to perform high-throughput experiments on much larger scales than before.

Chapter 7

LIST OF COMMUNICATIONS

Articles

1. P. Bisignano, S. Doerr, M. J. Harvey, A. D. Favia, A. Cavalli, and G. De Fabritiis. Kinetic Characterization of Fragment Binding in AmpC β-Lactamase by High-Throughput Molecular Simulations. Journal of Chemical Information and Modeling 54. 362-366 (2014).
2. S. Doerr and G. De Fabritiis. On-the-Fly Learning and Sampling of Ligand Binding by High-Throughput Molecular Simulations. Journal of Chemical Theory and Computation 10. 2064-2069 (2014).
3. S. Doerr, M. J. Harvey, Frank Noé and G. De Fabritiis. HTMD: High-Throughput Molecular Dynamics for Molecular Discovery. Journal of Chemical Theory and Computation 12. 1845-1852 (2016).

Posters

1. HTMD: Simulation-based molecular discovery. Stefan Doerr, Matt J. Harvey, Gianni De Fabritiis. Molecular and Chemical Kinetics 2015. September 7, 2015. CECAM-DE-MMS, Zuse Institute Berlin, Germany.

Chapter 8

APPENDIX: OTHER PUBLICATIONS

This section lists a publication in which I have contributed a minor part.

8.1 Reversible protein-protein association and binding in all-atom molecular dynamics

Nuria S. Platter, Stefan Doerr, Gianni De Fabritiis and Frank Noé. Manuscript in preparation.

Abstract
Protein-protein association is a basis for intra- and extra-cellular processes, including the assembly of biomolecular structure and machines, transport, signal transduction and inhibition. While robust techniques for the determination of kinetics, affinities and complex structures are available, our ability to resolve microscopic mechanisms in single protein complexes has as yet been limited. Using the tightly binding complex between the ribonuclease barnase and its inhibitor barstar, we here demonstrate that protein-protein association and dissociation can now be followed in fully atomistic detail using milliseconds of unbiased high-

throughput molecular dynamics simulations. By reweighting the trajectory swarm with Hidden Markov models, a model of the equilibrium kinetics is obtained that is consistent with available kinetic, thermodynamic and structural data, and yields insight into previously elusive details. Initial binding occurs into a structurally diverse ensemble of non-native encounter and mis-bound states. The complex then rearranges by means of surface and orientational search on the timescale of 10 to 100 s to the natively bound complex. The encounter complex funnels into a transition state of binding, in which barnase and barstar are anchored close to the binding configuration by one or a few key amino acids, which still permits some conformational flexibility. Final docking involves a small barrier that is overcome on the microseconds timescale, and is probably due to desolvation. In the bound state, the complex fluctuates between a more stable tightly-bound X-ray structure, and a less-stable loosely-bound structure on the microsecond to 100 microsecond timescale. The bound state is stable for minutes, then spontaneous denaturation to the encounter complex and rebinding occurs repeatedly before dissociation occurs on the timescales of many hours. These microscopic insights shed new light on the mechanisms of protein-protein association and suggest strategies to understand the role of mutants and find small molecule inhibitors.

Bibliography

[1] Sham HL, Kempf DJ, Molla A, Marsh KC, Kumar GN, Chen CM, et al. ABT-378, a highly potent inhibitor of the human immunodeficiency virus protease. Antimicrob Agents Chemother. 1998 Dec;42(12):3218–3224.

[2] Varghese JN. Development of neuraminidase inhibitors as anti-influenza virus drugs. Drug Dev Res. 1999 Mar;46(3-4):176–196. Available from: http://onlinelibrary.wiley.com/doi/10.1002/(SICI)1098-2299(199903/04)46:3/4<176::AID-DDR4>3.0.CO;2-6/abstract.

[3] Wood JM, Maibaum J, Rahuel J, Grütter MG, Cohen NC, Rasetti V, et al. Structure-based design of aliskiren, a novel orally effective renin inhibitor. Biochem Biophys Res Commun. 2003 Sep;308(4):698–705.

[4] Bantia S, Ananth SL, Parker CD, Horn LL, Upshaw R. Mechanism of inhibition of T-acute lymphoblastic leukemia cells by PNP inhibitor–BCX-1777. Int Immunopharmacol. 2003 Jun;3(6):879–887.

[5] Oliver WR, Shenk JL, Snaith MR, Russell CS, Plunket KD, Bodkin NL, et al. A selective peroxisome proliferator-activated receptor delta agonist promotes reverse cholesterol transport. Proc Natl Acad Sci USA. 2001 Apr;98(9):5306–5311.

[6] Karplus M. Behind the folding funnel diagram. Nat Chem Biol. 2011 Jul;7(7):401–404. Available from: http://www.nature.com/nchembio/journal/v7/n7/full/nchembio.565.html.

[7] Wensley BG, Batey S, Bone FAC, Chan ZM, Tumelty NR, Steward A, et al. Experimental evidence for a frustrated energy landscape in a three-helix-bundle protein family. Nature. 2010 Feb;463(7281):685–688. Available from: http://www.nature.com/nature/journal/v463/n7281/full/nature08743.html.

[8] Gebhardt JCM, Bornschlögl T, Rief M. Full distance-resolved folding energy landscape of one single protein molecule. PNAS. 2010 Feb;107(5):2013–2018. Available from: http://www.pnas.org/content/107/5/2013.

[9] Santoso Y, Joyce CM, Potapova O, Reste LL, Hohlbein J, Torella JP, et al. Conformational transitions in DNA polymerase I revealed by single-molecule FRET. PNAS. 2010 Jan;107(2):715–720. Available from: http://www.pnas.org/content/107/2/715.

[10] Eisenmesser EZ, Millet O, Labeikovsky W, Korzhnev DM, Wolf-Watz M, Bosco DA, et al. Intrinsic dynamics of an enzyme underlies catalysis. Nature. 2005 Nov;438(7064):117–121. Available from: http://www.nature.com/nature/journal/v438/n7064/full/nature04105.html.

[11] Min W, Luo G, Cherayil BJ, Kou SC, Xie XS. Observation of a Power-Law Memory Kernel for Fluctuations within a Single Protein Molecule. Phys Rev Lett. 2005 May;94(19):198302. Available from: http://link.aps.org/doi/10.1103/PhysRevLett.94.198302.

[12] Neubauer H, Gaiko N, Berger S, Schaffer J, Eggeling C, Tuma J, et al. Orientational and Dynamical Heterogeneity of Rhodamine

6G Terminally Attached to a DNA Helix Revealed by NMR and Single-Molecule Fluorescence Spectroscopy. J Am Chem Soc. 2007 Oct;129(42):12746–12755. Available from: http://dx.doi.org/10.1021/ja0722574.

[13] Gansen A, Valeri A, Hauger F, Felekyan S, Kalinin S, Tóth K, et al. Nucleosome disassembly intermediates characterized by single-molecule FRET. PNAS. 2009 Sep;106(36):15308–15313. Available from: http://www.pnas.org/content/106/36/15308.

[14] Wright PE, Dyson HJ. Intrinsically disordered proteins in cellular signalling and regulation. Nat Rev Mol Cell Biol. 2015 Jan;16(1):18–29. Available from: http://www.nature.com/nrm/journal/v16/n1/full/nrm3920.html.

[15] Noé F, Krachtus D, Smith JC, Fischer S. Transition Networks for the Comprehensive Characterization of Complex Conformational Change in Proteins. J Chem Theory Comput. 2006 May;2(3):840–857. Available from: http://dx.doi.org/10.1021/ct050162r.

[16] Fischer S, Windshügel B, Horak D, Holmes KC, Smith JC. Structural mechanism of the recovery stroke in the Myosin molecular motor. PNAS. 2005 May;102(19):6873–6878. Available from: http://www.pnas.org/content/102/19/6873.

[17] Kobitski AY, Nierth A, Helm M, Jäschke A, Nienhaus GU. Mg2+-dependent folding of a Diels-Alderase ribozyme probed by single-molecule FRET analysis. Nucl Acids Res. 2007 Mar;35(6):2047–2059. Available from: http://nar.oxfordjournals.org/content/35/6/2047.

[18] Jäger M, Zhang Y, Bieschke J, Nguyen H, Dendle M, Bowman ME, et al. Structure–function–folding relationship in a WW domain. PNAS. 2006 Jul;103(28):10648–10653. Available from: http://www.pnas.org/content/103/28/10648.

[19] Ostermann A, Waschipky R, Parak FG, Nienhaus GU. Ligand binding and conformational motions in myoglobin. Nature. 2000 Mar;404(6774):205–208. Available from: http://www.nature.com/nature/journal/v404/n6774/full/404205a0.html.

[20] Dror RO, Dirks RM, Grossman JP, Xu H, Shaw DE. Biomolecular Simulation: A Computational Microscope for Molecular Biology. Annual Review of Biophysics. 2012;41(1):429–452. Available from: http://www.annualreviews.org/doi/abs/10.1146/annurev-biophys-042910-155245.

[21] Kendrew JC, Bodo G, Dintzis HM, Parrish RG, Wyckoff H, Phillips DC. A three-dimensional model of the myoglobin molecule obtained by x-ray analysis. Nature. 1958 Mar;181(4610):662–666.

[22] Carpenter EP, Beis K, Cameron AD, Iwata S. Overcoming the challenges of membrane protein crystallography. Curr Opin Struct Biol. 2008 Oct;18(5):581–586.

[23] Acharya KR, Lloyd MD. The advantages and limitations of protein crystal structures. Trends in Pharmacological Sciences. 2005 Jan;26(1):10–14. Available from: http://www.sciencedirect.com/science/article/pii/S0165614704003050.

[24] Cousido-Siah A, Petrova T, Hazemann I, Mitschler A, Ruiz FX, Howard E, et al. Crystal packing modifies ligand binding affinity: the case of aldose reductase. Proteins. 2012 Nov;80(11):2552–2561.

[25] Hashem Y, des Georges A, Fu J, Buss SN, Jossinet F, Jobe A, et al. High-resolution cryo-electron microscopy structure of the Trypanosoma brucei ribosome. Nature. 2013 Feb;494(7437):385–389.

[26] Zhao G, Perilla JR, Yufenyuy EL, Meng X, Chen B, Ning J, et al. Mature HIV-1 capsid structure by cryo-electron microscopy and all-atom molecular dynamics. Nature. 2013 May;497(7451):643–646. Available from: http://www.nature.com/nature/journal/v497/n7451/full/nature12162.html.

[27] Greber BJ, Bieri P, Leibundgut M, Leitner A, Aebersold R, Boehringer D, et al. The complete structure of the 55S mammalian mitochondrial ribosome. Science. 2015 Apr;348(6232):303–308. Available from: http://science.sciencemag.org/content/348/6232/303.

[28] Kleckner IR, Foster MP. An introduction to NMR-based approaches for measuring protein dynamics. Biochim Biophys Acta. 2011 Aug;1814(8):942–968.

[29] Kang C, Li Q. Solution NMR study of integral membrane proteins. Current Opinion in Chemical Biology. 2011 Aug;15(4):560–569. Available from: http://www.sciencedirect.com/science/article/pii/S1367593111000949.

[30] Pellecchia M, Bertini I, Cowburn D, Dalvit C, Giralt E, Jahnke W, et al. Perspectives on NMR in drug discovery: a technique comes of age. Nat Rev Drug Discov. 2008 Sep;7(9):738–745. Available from: http://www.nature.com/nrd/journal/v7/n9/abs/nrd2606.html.

[31] Pellecchia M, Sem DS, Wüthrich K. Nmr in drug discovery. Nat Rev Drug Discov. 2002 Mar;1(3):211–219. Available from: http://www.nature.com/nrd/journal/v1/n3/abs/nrd748.html.

[32] Chung HS, McHale K, Louis JM, Eaton WA. Single-Molecule Fluorescence Experiments Determine Protein Folding Transition Path Times. Science. 2012 Feb;335(6071):981–984. Available from: http://science.sciencemag.org/content/335/6071/981.

[33] Stigler J, Ziegler F, Gieseke A, Gebhardt JCM, Rief M. The Complex Folding Network of Single Calmodulin Molecules. Science. 2011 Oct;334(6055):512–516. Available from: http://science.sciencemag.org/content/334/6055/512.

[34] Pirchi M, Ziv G, Riven I, Cohen SS, Zohar N, Barak Y, et al. Single-molecule fluorescence spectroscopy maps the folding landscape of a large protein. Nat Commun. 2011 Oct;2:493. Available from: http://www.nature.com/ncomms/journal/v2/n10/full/ncomms1504.html.

[35] Lindorff-Larsen K, Piana S, Dror RO, Shaw DE. How Fast-Folding Proteins Fold. Science. 2011 Oct;334(6055):517–520. Available from: http://www.sciencemag.org/content/334/6055/517.

[36] Voelz VA, Bowman GR, Beauchamp K, Pande VS. Molecular simulation of ab initio protein folding for a millisecond folder NTL9(1-39). J Am Chem Soc. 2010 Feb;132(5):1526–1528. Available from: http://www.ncbi.nlm.nih.gov/pubmed/20070076.

[37] Bowman GR, Voelz VA, Pande VS. Atomistic Folding Simulations of the Five-Helix Bundle Protein 685. J Am Chem Soc. 2011 Feb;133(4):664–667. Available from: http://dx.doi.org/10.1021/ja106936n.

[38] Buch I, Giorgino T, De Fabritiis G. Complete reconstruction of an enzyme-inhibitor binding process by molecular dynamics simulations. Proceedings of the National Academy of Sciences. 2011;108(25):10184–10189. Available from: http://www.pnas.org/content/108/25/10184.abstract.

[39] Shaw DE, Maragakis P, Lindorff-Larsen K, Piana S, Dror RO, Eastwood MP, et al. Atomic-Level Characterization of the Structural Dynamics of Proteins. Science. 2010 Oct;330(6002):341

–346. Available from: http://www.sciencemag.org/content/330/6002/341.abstract.

[40] Dror RO, Pan AC, Arlow DH, Borhani DW, Maragakis P, Shan Y, et al. Pathway and mechanism of drug binding to G-protein-coupled receptors. PNAS. 2011 Aug;108(32):13118–13123. Available from: http://www.pnas.org/content/108/32/13118.

[41] Kohlhoff KJ, Shukla D, Lawrenz M, Bowman GR, Konerding DE, Belov D, et al. Cloud-based simulations on Google Exacycle reveal ligand modulation of GPCR activation pathways. Nat Chem. 2014 Jan;6(1):15–21. Available from: http://www.nature.com/nchem/journal/v6/n1/full/nchem.1821.html?WT.mc_id=TWT_NatureChemistry.

[42] Nygaard R, Zou Y, Dror RO, Mildorf TJ, Arlow DH, Manglik A, et al. The Dynamic Process of 2-Adrenergic Receptor Activation. Cell. 2013 Jan;152(3):532–542. Available from: http://www.cell.com/article/S0092867413000111/abstract.

[43] Plattner N, Noé F. Protein conformational plasticity and complex ligand-binding kinetics explored by atomistic simulations and Markov models. Nat Commun. 2015 Jul;6:7653. Available from: http://www.nature.com/ncomms/2015/150702/ncomms8653/full/ncomms8653.html.

[44] Silva DA, Bowman GR, Sosa-Peinado A, Huang X. A Role for Both Conformational Selection and Induced Fit in Ligand Binding by the LAO Protein. PLoS Comput Biol. 2011;7(5):e1002054. Available from: http://dx.doi.org/10.1371/journal.pcbi.1002054.

[45] Bisignano P, Doerr S, Harvey MJ, Favia AD, Cavalli A, De Fabritiis G. Kinetic Characterization of Fragment Binding in AmpC -Lactamase by High-Throughput Molecular Simulations. J Chem

Inf Model. 2014 Feb;54(2):362–366. Available from: http://dx.doi.org/10.1021/ci4006063.

[46] Sadiq SK, Noé F, De Fabritiis G. Kinetic characterization of the critical step in HIV-1 protease maturation. Proceedings of the National Academy of Sciences. 2012;109(50):20449–20454. Available from: http://www.pnas.org/content/109/50/20449.abstract.

[47] Silva DA, Weiss DR, Avila FP, Da LT, Levitt M, Wang D, et al. Millisecond dynamics of RNA polymerase II translocation at atomic resolution. PNAS. 2014 May;111(21):7665–7670. Available from: http://www.pnas.org/content/111/21/7665.

[48] Zhu F, Hummer G. Pore opening and closing of a pentameric ligand-gated ion channel. PNAS. 2010 Nov;107(46):19814–19819. Available from: http://www.pnas.org/content/107/46/19814.

[49] Jensen M, Jogini V, Borhani DW, Leffler AE, Dror RO, Shaw DE. Mechanism of Voltage Gating in Potassium Channels. Science. 2012 Apr;336(6078):229–233. Available from: http://www.sciencemag.org/content/336/6078/229.

[50] Bernèche S, Roux B. Energetics of ion conduction through the K+ channel. Nature. 2001 Nov;414(6859):73–77. Available from: http://www.nature.com/nature/journal/v414/n6859/full/414073a0.html.

[51] Köpfer DA, Song C, Gruene T, Sheldrick GM, Zachariae U, Groot BLd. Ion permeation in K+ channels occurs by direct Coulomb knock-on. Science. 2014 Oct;346(6207):352–355. Available from: http://science.sciencemag.org/content/346/6207/352.

[52] Reubold TF, Faelber K, Plattner N, Posor Y, Ketel K, Curth U, et al. Crystal structure of the dynamin tetramer.

Nature. 2015 Sep;525(7569):404–408. Available from: http://www.nature.com/nature/journal/v525/n7569/full/nature14880.html.

[53] Arkhipov A, Yin Y, Schulten K. Membrane-Bending Mechanism of Amphiphysin N-BAR Domains. Biophysical Journal. 2009 Nov;97(10):2727–2735. Available from: http://www.sciencedirect.com/science/article/pii/S0006349509014350.

[54] Blood PD, Voth GA. Direct observation of Bin/amphiphysin/Rvs (BAR) domain-induced membrane curvature by means of molecular dynamics simulations. PNAS. 2006 Oct;103(41):15068–15072. Available from: http://www.pnas.org/content/103/41/15068.

[55] Stanley N, Esteban-Martín S, De Fabritiis G. Kinetic modulation of a disordered protein domain by phosphorylation. Nat Commun. 2014 Oct;5:5272. Available from: http://www.nature.com/ncomms/2014/141028/ncomms6272/abs/ncomms6272.html.

[56] Brooks BR, Brooks CL, Mackerell AD, Nilsson L, Petrella RJ, Roux B, et al. CHARMM: the biomolecular simulation program. J Comput Chem. 2009 Jul;30(10):1545–1614. Available from: http://www.ncbi.nlm.nih.gov/pubmed/19444816.

[57] Case DA, Betz RM, Botello-Smith W, Cerutti DS, Cheatham TE, Darden TA, et al.. AMBER 2016. University of California, San Francisco; 2016.

[58] Karplus M, McCammon JA. Molecular dynamics simulations of biomolecules. Nat Struct Mol Biol. 2002 Sep;9(9):646–652. Available from: http://www.nature.com/nsmb/journal/v9/n9/full/nsb0902-646.html.

[59] Lindorff-Larsen K, Piana S, Palmo K, Maragakis P, Klepeis JL, Dror RO, et al. Improved side-chain torsion potentials for the Amber ff99SB protein force field. Proteins. 2010 Jun;78(8):1950–1958.

[60] Best RB, Buchete NV, Hummer G. Are current molecular dynamics force fields too helical? Biophys J. 2008 Jul;95(1):L07–09.

[61] Hornak V, Abel R, Okur A, Strockbine B, Roitberg A, Simmerling C. Comparison of multiple Amber force fields and development of improved protein backbone parameters. Proteins. 2006 Nov;65(3):712–725.

[62] Best R, Mittal J, Feig M, MacKerell A. Inclusion of Many-Body Effects in the Additive CHARMM Protein CMAP Potential Results in Enhanced Cooperativity of -Helix and -Hairpin Formation. Biophys J. 2012 Sep;103(5):1045–1051. Available from: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3433603/.

[63] Lindorff-Larsen K, Maragakis P, Piana S, Eastwood MP, Dror RO, Shaw DE. Systematic validation of protein force fields against experimental data. PLoS ONE. 2012;7(2):e32131.

[64] Ohmura I, Morimoto G, Ohno Y, Hasegawa A, Taiji M. MDGRAPE-4: a special-purpose computer system for molecular dynamics simulations. Phil Trans R Soc A. 2014 Aug;372(2021):20130387. Available from: http://rsta.royalsocietypublishing.org/content/372/2021/20130387.

[65] Shaw DE, Grossman JP, Bank JA, Batson B, Butts JA, Chao JC, et al. Anton 2: Raising the Bar for Performance and Programmability in a Special-purpose Molecular Dynamics Supercomputer. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. SC '14. Pis-

cataway, NJ, USA: IEEE Press; 2014. p. 41–53. Available from: http://dx.doi.org/10.1109/SC.2014.9.

[66] Shaw DE, Deneroff MM, Dror RO, Kuskin JS, Larson RH, Salmon JK, et al. Anton, a special-purpose machine for molecular dynamics simulation. In: Proceedings of the 34th annual international symposium on Computer architecture. ISCA '07. New York, NY, USA: ACM; 2007. p. 1–12. Available from: http://doi.acm.org/10.1145/1250662.1250664.

[67] Pande VS, Baker I, Chapman J, Elmer SP, Khaliq S, Larson SM, et al. Atomistic protein folding simulations on the submillisecond time scale using worldwide distributed computing. Biopolymers. 2003 Jan;68(1):91–109.

[68] Buch I, Harvey MJ, Giorgino T, Anderson DP, De Fabritiis G. High-Throughput All-Atom Molecular Dynamics Simulations Using Distributed Computing. J Chem Inf Model. 2010 Mar;50(3):397–403. Available from: http://dx.doi.org/10.1021/ci900455r.

[69] Pérez-Hernández G, Paul F, Giorgino T, Fabritiis GD, Noé F. Identification of slow molecular order parameters for Markov model construction. The Journal of Chemical Physics. 2013 Jul;139(1):015102. Available from: http://scitation.aip.org/content/aip/journal/jcp/139/1/10.1063/1.4811489.

[70] Prinz JH, Wu H, Sarich M, Keller B, Senne M, Held M, et al. Markov models of molecular kinetics: Generation and validation. The Journal of Chemical Physics. 2011 May;134(17):174105–174105–23. Available from: http://jcp.aip.org/resource/1/jcpsa6/v134/i17/p174105_s1?isAuthorized=no.

[71] Sarich M, Noé F, Schütte C. On the Approximation Quality of Markov State Models. Multiscale Model Simul. 2010

Jan;8(4):1154–1177. Available from: http://epubs.siam.org/doi/abs/10.1137/090764049.

[72] Djurdjevac N, Sarich M, Schütte C. Estimating the Eigenvalue Error of Markov State Models. Multiscale Model Simul. 2012 Jan;10(1):61–81. Available from: http://epubs.siam.org/doi/abs/10.1137/100798910.

[73] Prinz JH, Chodera JD, Noé F. Spectral Rate Theory for Two-State Kinetics. Phys Rev X. 2014 Feb;4(1):011020. Available from: http://link.aps.org/doi/10.1103/PhysRevX.4.011020.

[74] Noé F, Schütte C, Vanden-Eijnden E, Reich L, Weikl TR. Constructing the equilibrium ensemble of folding pathways from short off-equilibrium simulations. PNAS. 2009 Nov;106(45):19011–19016. Available from: http://www.pnas.org/content/106/45/19011.

[75] Noé F, Doose S, Daidone I, Löllmann M, Sauer M, Chodera JD, et al. Dynamical fingerprints for probing individual relaxation processes in biomolecular dynamics with simulations and kinetic experiments. PNAS. 2011 Mar;108(12):4822–4827. Available from: http://www.pnas.org/content/108/12/4822.

[76] Amadei A, Linssen ABM, Berendsen HJC. Essential dynamics of proteins. Proteins. 1993 Dec;17(4):412–425. Available from: http://onlinelibrary.wiley.com/doi/10.1002/prot.340170408/abstract.

[77] Schwantes CR, Pande VS. Improvements in Markov State Model Construction Reveal Many Non-Native Interactions in the Folding of NTL9. J Chem Theory Comput. 2013 Apr;9(4):2000–2009. Available from: http://dx.doi.org/10.1021/ct300878a.

[78] Schwantes CR, Shukla D, Pande VS. Markov State Models and tICA Reveal a Nonnative Folding Nucleus in Simulations of NuG2. Biophysical Journal. 2016 Apr;110(8):1716–1719. Available from: http://www.cell.com/article/S0006349516301072/abstract.

[79] Schwantes CR, Pande VS. Modeling Molecular Kinetics with tICA and the Kernel Trick. J Chem Theory Comput. 2015 Feb;11(2):600–608. Available from: http://dx.doi.org/10.1021/ct5007357.

[80] Noé F, Clementi C. Kinetic Distance and Kinetic Maps from Molecular Dynamics Simulation. J Chem Theory Comput. 2015 Oct;11(10):5002–5011. Available from: http://dx.doi.org/10.1021/acs.jctc.5b00553.

[81] Zuckerman DM. Equilibrium Sampling in Biomolecular Simulations. Annual Review of Biophysics. 2011;40(1):41–62. Available from: http://dx.doi.org/10.1146/annurev-biophys-042910-155255.

[82] Hansen HS, Hünenberger PH. Using the local elevation method to construct optimized umbrella sampling potentials: calculation of the relative free energies and interconversion barriers of glucopyranose ring conformers in water. J Comput Chem. 2010 Jan;31(1):1–23.

[83] Buch I, Sadiq SK, De Fabritiis G. Optimized Potential of Mean Force Calculations for Standard Binding Free Energies. J Chem Theory Comput. 2011 Jun;7(6):1765–1772. Available from: http://dx.doi.org/10.1021/ct2000638.

[84] Hamelberg D, Mongan J, McCammon JA. Accelerated molecular dynamics: a promising and efficient simulation method for biomolecules. J Chem Phys. 2004 Jun;120(24):11919–11929.

[85] Grant BJ, Gorfe AA, McCammon JA. Ras Conformational Switching: Simulating Nucleotide-Dependent Conformational Transitions with Accelerated Molecular Dynamics. PLOS Comput Biol. 2009 Mar;5(3):e1000325. Available from: http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1000325.

[86] Laio A, Parrinello M. Escaping free-energy minima. PNAS. 2002 Oct;99(20):12562–12566. Available from: http://www.pnas.org/content/99/20/12562.

[87] Dickson BM. Approaching a parameter-free metadynamics. Phys Rev E. 2011 Sep;84(3):037701. Available from: http://link.aps.org/doi/10.1103/PhysRevE.84.037701.

[88] Limongelli V, Bonomi M, Parrinello M. Funnel metadynamics as accurate binding free-energy method. Proc Natl Acad Sci USA. 2013 Apr;110(16):6358–6363.

[89] Marinari E, Parisi G. Simulated Tempering: A New Monte Carlo Scheme. EPL. 1992;19(6):451. Available from: http://stacks.iop.org/0295-5075/19/i=6/a=002.

[90] Hansmann UHE. Parallel tempering algorithm for conformational studies of biological molecules. Chemical Physics Letters. 1997 Dec;281(1–3):140–150. Available from: http://www.sciencedirect.com/science/article/pii/S0009261497011986.

[91] Sugita Y, Okamoto Y. Replica-exchange molecular dynamics method for protein folding. Chemical Physics Letters. 1999 Nov;314(1–2):141–151. Available from: http://www.sciencedirect.com/science/article/pii/S0009261499011239.

[92] Chen J, Brooks CL, Khandogin J. Recent advances in implicit solvent-based methods for biomolecular simulations. Curr Opin Struct Biol. 2008 Apr;18(2):140–148.

[93] Monticelli L, Kandasamy SK, Periole X, Larson RG, Tieleman DP, Marrink SJ. The MARTINI Coarse-Grained Force Field: Extension to Proteins. J Chem Theory Comput. 2008 May;4(5):819–834. Available from: http://dx.doi.org/10.1021/ct700324x.

[94] de Jong DH, Singh G, Bennett WFD, Arnarez C, Wassenaar TA, Schäfer LV, et al. Improved Parameters for the Martini Coarse-Grained Protein Force Field. J Chem Theory Comput. 2013 Jan;9(1):687–697. Available from: http://dx.doi.org/10.1021/ct300646g.

[95] Doerr S, De Fabritiis G. On-the-Fly Learning and Sampling of Ligand Binding by High-Throughput Molecular Simulations. J Chem Theory Comput. 2014 May;10(5):2064–2069. Available from: http://dx.doi.org/10.1021/ct400919u.

[96] Zimmerman MI, Bowman GR. FAST Conformational Searches by Balancing Exploration/Exploitation Trade-Offs. J Chem Theory Comput. 2015 Dec;11(12):5747–5757. Available from: http://dx.doi.org/10.1021/acs.jctc.5b00737.

[97] Doerr S, Harvey MJ, Noé F, De Fabritiis G. HTMD: High-Throughput Molecular Dynamics for Molecular Discovery. J Chem Theory Comput. 2016 Apr;12(4):1845–1852. Available from: http://dx.doi.org/10.1021/acs.jctc.6b00049.

[98] Poli R, Kennedy J, Blackwell T. Particle swarm optimization. Swarm Intell. 2007 Aug;1(1):33–57. Available from: http://link.springer.com/article/10.1007/s11721-007-0002-0.

[99] Tabu Search—Part I. ORSA Journal on Computing. 1989 Aug;1(3):190–206. Available from: http://pubsonline.informs.org/doi/abs/10.1287/ijoc.1.3.190.

[100] Tabu Search—Part II. ORSA Journal on Computing. 1990 Feb;2(1):4–32. Available from: http://pubsonline.informs.org/doi/abs/10.1287/ijoc.2.1.4.

[101] Abraham MJ, Murtola T, Schulz R, Páll S, Smith JC, Hess B, et al. GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. SoftwareX. 2015 Sep;1–2:19–25. Available from: http://www.sciencedirect.com/science/article/pii/S2352711015000059.

[102] Scherer MK, Trendelkamp-Schroer B, Paul F, Pérez-Hernández G, Hoffmann M, Plattner N, et al. PyEMMA 2: A Software Package for Estimation, Validation, and Analysis of Markov Models. J Chem Theory Comput. 2015 Nov;11(11):5525–5542. Available from: http://dx.doi.org/10.1021/acs.jctc.5b00743.

[103] Beauchamp KA, Bowman GR, Lane TJ, Maibaum L, Haque IS, Pande VS. MSMBuilder2: Modeling Conformational Dynamics on the Picosecond to Millisecond Scale. J Chem Theory Comput. 2011 Oct;7(10):3412–3419. Available from: http://dx.doi.org/10.1021/ct200463m.

[104] McGibbon R, Beauchamp K, Harrigan M, Klein C, Swails J, Hernández C, et al. MDTraj: A Modern Open Library for the Analysis of Molecular Dynamics Trajectories. Biophysical Journal. 2015 Oct;109(8):1528–1532. Available from: http://www.sciencedirect.com/science/article/pii/S0006349515008267.

[105] Humphrey W, Dalke A, Schulten K. VMD: Visual molecular dynamics. J Mol Graphics. 1996 Feb;14(1):33–38. Available

from: http://www.sciencedirect.com/science/article/pii/0263785596000185.

[106] Jo S, Kim T, Iyer VG, Im W. CHARMM-GUI: A web-based graphical user interface for CHARMM. J Comput Chem. 2008 Aug;29(11):1859–1865. Available from: http://onlinelibrary.wiley.com/doi/10.1002/jcc.20945/abstract.

[107] Vanommeslaeghe K, Hatcher E, Acharya C, Kundu S, Zhong S, Shim J, et al. CHARMM general force field: A force field for drug-like molecules compatible with the CHARMM all-atom additive biological force fields. J Comput Chem. 2010 Mar;31(4):671–690.

[108] Coates A, Ng AY, Lee H. An analysis of single-layer networks in unsupervised feature learning. In: International Conference on Artificial Intelligence and Statistics; 2011. p. 215–223.

[109] Sultan MM, Kiss G, Shukla D, Pande VS. Automatic Selection of Order Parameters in the Analysis of Large Scale Molecular Dynamics Simulations. J Chem Theory Comput. 2014 Dec;10(12):5217–5223. Available from: http://dx.doi.org/10.1021/ct500353m.

[110] Bacallado S, Chodera JD, Pande V. Bayesian comparison of Markov models of molecular dynamics with detailed balance constraint. J Chem Phys. 2009 Jul;131(4). Available from: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2730706/.

[111] Bowman GR, Beauchamp KA, Boxer G, Pande VS. Progress and challenges in the automated construction of Markov state models for full protein systems. The Journal of Chemical Physics. 2009 Sep;131(12):124101. Available from: http://scitation.aip.org/content/aip/journal/jcp/131/12/10.1063/1.3216567.