SISSA
Scuola
Internazionale
Superiore di
Studi Avanzati

Scuola Internazionale Superiore di Studi Avanzati

Doctoral Thesis

An unsupervised approach to the analysis of free energy landscapes and to protein design

Author:
Giulia Sormani

Supervisor:
Prof. Alessandro Laio

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

PhD course in Physics and Chemistry of Biological Systems
Molecular and Statistical Biophysics Group

19 November, 2020

Via Bonomea, 265
34136 Trieste – Italy
T +39 0403787111
E info@sissa.it
sissa.it

Contents

Ad Alberto, Agnese e Nicoló

Introduction

Over the past 20 years, data availability has exponentially increased in various fields. This growth is still expected to be accelerating in the future [1]. The term Big Data has been coined to refer to enormous datasets that, due to their size and their heterogeneity, pose important challenges in terms of acquisition, storage, and management [2]. Indeed, quoting from [3], *while computers get faster and produce more and more data, the processing power of human brains remains roughly constant* [4]. This has led to an increased interest in the development of machine learning techniques for efficient information filtering and processing. The objective of machine learning is to extract the relevant information from the data and make it available to the user in a form which is compatible with the "processing power" of human brain. There are two main categories of machine learning techniques. The first one is supervised learning in which a set of training data is used to infer information about new input data. The second one is unsupervised learning which tries to discover the underlying structure of the data without training the model on a ground truth, ideally without any human control. In this thesis we address two different problems of computational biochemistry by approaches that shares the philosophy of unsupervised learning: solving the problem with the minimum amount of human intervention.

The first field we investigate is the characterization of the free energy landscapes explored in Molecular Dynamics(MD) simulations of biomolecules. MD produces massive data sets containing the positions of all the atoms of the systems, which can be several thousands for simulations in explicit solvent. Moreover, these positions are stored a lot of times since the timescales of interest (often of the order of milliseconds) are reached trough small time steps (usually of around one femtosecond). A first

trivial selection of the data is performed decimating the frames with a fixed stride, since consecutive frames are correlated. Usually, the following step is the choice of a "feature representation" for the molecular system. A typical example of such a representation is the space defined by the position of all the C_α carbons of a protein. The number of coordinates defining this space is much smaller than the original one. At this point one has performed all the "trivial" steps, based on a "a priori" knowledge of the system such as "the solvent in my system doesn't matter" or "frames that are closer in time than x nanoseconds are correlated". The obtained representation of the simulation is a more compact representation, but it is assumed to contain all the important details. Still, this representation is far from being human-readable: the trajectory in the C_α space is a time series of a vector with $\mathcal{O}(1000)$ components. A possible approach to further reduce the dimension is provided by machine learning algorithms, which are becoming a popular tool for the analysis of MD simulations.
The par excellence unsupervised machine learning algorithm, in the field of biophysics, is Principal Component Analysis(PCA) [5]. This algorithm belongs to the category of the so called projection methods, whose aim is to provide a low dimensional representation of the data that is easy to interpret. In particular, PCA is a linear projection method. This means that it provides a correct projection if the manifold on which the data lie is an hyperplane. PCA has been widely used for the study of molecular systems [5–10]. However, assuming the existence of an hyperplane which can capture the relevant properties of a biomolecule is a strong assumption, which can easily fall short causing systematic errors in the predictions [11]. To overcome these limitations one can use a non-linear projection method such as Isomap [12] or Kernel-PCA [13]. These algorithms can also deal with data lying on curved and twisted manifolds. However, it's important to notice that the manifolds containing the data from MD simulations can have a complex topology; in other words: not only they are not hyperplanes but also they cannot be "ironed" to an hyperplan. This poses a constraint on the maximum level of dimensionality reduction that can be achieved. Let's consider the example of data points lying on the surface of a three dimensional sphere: even if the points lye on a surface it is topologically impossible to get a two-dimensional representation of them preserving the neighbourhood of all the data points.

In chapter 2 of this thesis we describe an approach, which allows to analyse a

MD simulation without performing any dimensionality reduction beyond the choice of the features describing the system. This approach has been explicitly designed to be applicable regardless of the topological complexity of the embedding manifold.

The first step of our approach is the calculation of the intrinsic dimension(ID) of the embedding manifold, as explained in section 2.2. Indeed, even if this manifold is topologically complex, it typically has a dimensionality which is much smaller than the number of the feature coordinates. This is a consequence of all the physical and chemical restraints which prevent the biomolecule from moving in many directions.

The following step, which is the core of our approach, is the calculation of the free energy of each data point, using the PAk estimator described in section 2.3.2. This algorithm, is akin to other algorithms from data science for estimating the density of points, such as the ones described in Ref. [14–17]. The distinctive feature of PAk, is that it estimates the density, or, equivalently the free energy, directly on the manifold on which the data actually lie, which is characterized by a well defined ID. The algorithm depends on the knowledge of this ID, but the collective variables that define this reduced space don't have to be explicitly specified.

The knowledge of the free energy of each data point is very useful for the study of a biomolecular system since all the relevant features can be obtained by the analysis of the gross topological features of the free energy landscape, in particular the number and the relative locations of the free energy minima. Also this kind of investigation can be performed exploiting machine learning techniques, specifically by using density based clustering algorithms [18–21]. The first clustering algorithm presented in this thesis is Density Peak (DP) clustering, described in section 2.4.1. In this algorithm the metastable states are obtained from a direct analysis of the free energy distribution: each free energy minimum corresponds to a cluster center. In chapter 3, we present an application of all the steps of our approach, using DP algorithm for the clustering step. The aim of this analysis is to identify the metastable states of the main protease of the coronavirus SARS-CoV-2 and, based on the structure of these states, proposing a possible strategy to block the action of this protein through allosteric inhibition.

The second clustering algorithm presented in this thesis is the $\hat{k}$-Peaks clustering,

described in section 4.4.2. This algorithm represents the main algorithmic development of this thesis. $\hat{k}$-Peaks clustering was developed while trying to characterize the free energy folding landscape of the Villin protein, as explained in chapter 4. Indeed, we realized that none among the clusters detected using the DP clustering mapped correctly the unfolded state, which is however a metastable state of the system. A posteriori, we understood the reason for this "failure": the unfolded state does not correspond to a free energy minima. It is indeed a kinetic trap stabilized by entropy and it corresponds to a large flat region of the free energy. To solve this issue we thus devised a new clustering algorithm in which a key role is played by the variable $\hat{k}$, which is the number of neighbours for which the density around each point can be considered constant. The idea is that there are two situations in which $\hat{k}_i$ assumes high values. The first one is in the free energy minima, where the high density of points leads to a high value of $\hat{k}_i$. The second one is in the flat regions of the free energy landscape, where the low variation of the density of points leads also to high values of $\hat{k}_i$. Therefore, we propose that in order to characterize the kinetics of a system in which at least one state is stabilized by conformational disorder it is convenient to look for the peaks of $\hat{k}_i$; which become the new cluster centers. In chapter 4 we describe how $\hat{k}$-Peaks clustering detects also metastable states stabilized by entropy. Summarizing: our approach allows to characterize the properties of a biomolecule from the analysis of a MD simulation without performing an explicit dimensional reduction beyond the initial choice of the feature space, which however in the applications we present is very high dimensional. This is a great advantage with respect to previous methods. The relevant states for describing the molecule, are directly obtained from the analysis of the free energy landscape, through a clustering technique. The first clustering algorithm developed in our group is able to detect the free energy minima and the connections among them. The main methodological novelty introduced in this thesis is a clustering algorithm able to detect also large flat regions of the free energy corresponding to entropic traps. This algorithm can thus be a very useful method to analyse systems that undergo a phase transition from a disordered state to an ordered one.

The second part of the thesis (chapter 5) is devoted to protein design.

Protein design is the so called inverse folding problem: it aims at identifying sequences of amminoacids compatible with a given protein scaffold. Our original idea was the design of an alien protein fold, which means a protein fold which satisfies the key structural requirements of existing proteins, but which has not yet been observed in nature. Several folds with these features where proposed in Ref. [22]. For such a goal, using an automated and unsupervised algorithm is mandatory.

Our plan was to use Rosetta Design, which is a popular computational design software package that takes as input a protein structure and gives as output the corresponding sequence. We thus first tested Rosetta Design, used as a black box, for the design of two small existing proteins, one belonging to the SH3-1 family, the other belonging to the Ubiquitin family. We wanted to check if the output sequences of the software were similar to the natural ones. We measured the similarity between the designed sequences and the sequences belonging to the natural families of these two proteins through standard bionformatics tools. Surprisingly, we found that the designed sequences were not recognized as belonging to their corresponding natural families, not even with a low statistical confidence. We thus realized that the impressive results that were obtained exploiting Rosetta Design [23–26], include as an essential part of the pipeline extensive human curation and the experimental validation of a large set of designed sequences. This is fully legitimate for practical purposes. However our tests indicate that Rosetta Design is not capable of producing meaningful sequences by a fully automated protocol, which exploits only an optimization algorithm, with no human curation of the results.

We therefore attempted to address this issue, improving the reliability of Rosetta Design and making it an unsupervised optimization algorithm according to the philosophy we always tried to follow in our work. We devised a Genetic Algorithm in which the design steps are combined with a progressive optimization of the agreement of the sequence with a database of natural sequences. The core idea is that the exploitation of the huge amount of information contained in natural sequences can drive the design towards the "correct" sequences. Importantly, along the optimization we don't give any information about the family membership of the input structure.

We applied the Genetic Algorithm for the design of the two proteins. The similarity of the obtained sequences to the natural ones is remarkably improved, if compared

with the similarity of the sequences obtained simply using Rosetta Design. Drawing the conclusions, we realized that the current algorithms for protein design are strongly based on human intervention. We thus proposed a possible direction that can be followed for making them fully unsupervised. The results we obtained are encouraging, but, as we will discuss in chapter 5, many problems are still unsolved, leaving a lot of room for future improvements.

Estimating Free-Energy Landscapes from Molecular Dynamics Simulations

This first chapter of the thesis is purely methodological. We present here an approach, for the analysis of free energy landscapes associated to MD simulations of bio-molecules which allows considering at the same time a very large of variables, for example the positions of all the C_α carbons of a protein. This approach will be then applied to the analysis of MD simulations of two different proteins in chapters 3 and 4.

A manner to quantitatively understand the relevant properties of a bio-molecule is the analysis of its free energy landscape, at a given temperature [27]. Indeed, free energy minima correspond to the metastable states of the system. The higher is the barrier between two minima, the smallest is the probability to observe a transition between the corresponding states [28]. Molecular dynamics(MD) simulations sample a probability distribution of the positions of all N atoms of the system [28–30]. We will denote this distribution as $\rho(x)$, where x are the coordinates of all the atoms. If the MD simulation is performed at a temperature T, then $\rho(x) \propto exp(-V(x)/K_BT)$, where $V(x)$ is the potential energy function.

In order to interpreter the results of a simulation, this distribution is often projected on a set of collective variables(CVs), which are functions of the coordinates of the system, and will be denoted as $S(x)$. The probability distribution $\rho(x)$ can thus be reduced to a function of the CVs by integrating $\rho(x)$ over all x, under the constraint $S(x) = s$:

$$\rho(s) = \int dx \rho(x) \delta(s - S(x)) \tag{2.1}$$

The free energy is then defined as

$$F(s) = -K_B T log(\rho(s)) \tag{2.2}$$

However, only an a priori knowledge of the system under study can lead to a meaningful choice of the set of CVs. Moreover, the projection on a single variable can bring to a description that is thermodynamically meaningful, but which does not capture the complexity of the kinetics. In particular, the height of a free energy barrier unavoidably depends on the chosen variable on which it is projected [31–33].
A more rigorous procedure for describing the kinetics is offered by Markov State Modeling (MSM) [34–36]. The core idea of this method is to describe the dynamics as a Markov process between a few metastable states. The most common procedure to obtain these states involves first grouping the conformations in a high number of microstates through conformational clustering (for example using k-means clustering [37] or the Ward algorithm [38]). The microstates are then grouped in Markov States, using dynamical clustering, for example through MPP [39] or PCCA [40]/PCCA+ [41]. These Markov States correspond to the metastable states of the system. MSM have been used to analyse a variety of biophysical process such as protein folding [3, 39, 42–44] or ligand binding [45–47].

The goal of this first chapter is to describe a protocol, developed in our group, which aims at obtaining a detailed description of the free energy landscapes associated to MD simulations of bio-molecules and of the kinetics on these landscapes. This analysis is performed in very high-dimensional spaces, taking into account at the same time several hundreds of different variables. This allows circumventing the problem of the choice of the CV.
The first step of our procedure, is the calculation of the free energy. To do so, it is first necessary to estimate the intrinsic dimension of the dataset(ID), which is the minimum number of variables that are needed to capture all the relevant features of a data landscape without significant information loss. We calculate the ID using the TWO-NN estimator described in section 2.2. Once the value of the ID is known, we evaluate the free energy of each frame and its uncertainty using the PAk estimator described in section 2.3.2. PAk allows to estimate the free energy in high dimensional

spaces, for example the space defined by the positions of all the C_α carbons of a protein. Importantly, PAk requires the knowledge of the ID of the space in which the data points are lying, but does not require knowing explicitly which variables define the reduced space.

The second step of our protocol is the analysis of the free energy landscape through a clustering algorithm: the Density-Peaks clustering described in section 2.4.1 or the $\hat{k}$-Peaks clustering described in section 4.4.2. The aim of both these algorithms is to build a topography of the dataset, which means finding the relevant states for describing the kinetic of the system, and the connections among them. The appropriateness of one of the algorithms or the other depends on the features of the system under consideration. Indeed, in Density Peaks clustering the states are defined by the free-energy minima. This algorithm can thus be used to study systems in which the relevant metastable states are enthalpic traps corresponding to the minima of the free energy in a high dimensional feature space. On the other hand, the salient feature of the $\hat{k}$-Peaks clustering algorithm, is the capability of identifying both flat regions of the free energy landscape (which correspond to entropic traps), and minima of the free-energy (which correspond to enthalpic traps). This algorithm is thus an efficient tool to study free energy landscape including metastable states stabilized by conformational disorder.

As we will see, the clusters obtained by both these procedures are very similar to the Markov States resulting from the dynamical clustering in Markov State modeling. Therefore, the procedure we developed allows also circumventing the explicit construction of a MSM. However, differently from the common procedures for building a MSM, the relevant kinetic states are here identified simply by analyzing the structure of the free energy landscape, without using kinetic information to optimize the partition, or for choosing the number of states.

In the last section of this chapter (2.5), we briefly summarize the theory of MSM, focusing on how the relevant timescales of the process under study can be obtained once the Markov States have been identified.

2.1 Feature Spaces and Metrics

Molecular dynamics(MD) simulations produce data sets very large both in the number of data point (namely the number of frames of the simulation), and in the number of simulated particles. Consider, for example, a MD simulation of the dynamics of a protein in explicit solvent. This system contains the atoms of the protein, whose number can range from $\simeq$100 to $\simeq$100000; moreover it also contains thousands of atoms of the solute. Taking into account the x-y-z coordinates of such a number of atoms, for each data point, becomes computationally infeasible. To analyse a MD simulation, before the application of any algorithm, it is thus convenient to choose a set of features defining a space in which the studied system can be characterized, without significant information loss. We denote the coordinates that define this spaces as $y(x)$, since they are a function of the atoms coordinates x. Their choice is based on a priori knowledge of the system, but as we will see, they are not the "classic" collective variables.

First of all, since the focus is on the protein, the atoms of the solvent are usually neglected. In the absence of an external force, we expect the protein's dynamic and thermodynamic to be invariant to translation and rotation of the protein. The chosen coordinates must thus satisfy these invariances. We here present three spaces, satisfying the above mentioned conditions, that are commonly used to represent proteins. For each of these spaces, we also define a metric (i.e we define the distance between couples of configurations), since this is needed in many of the algorithms used in our work.

- Space of the backbone dihedral angles [48]: the configuration of a protein is defined by its Ψ-backbone dihedral angles [49]. The number of these angles is equal to the number of the residues of the protein minus one. The distance between two configurations at time t and t' is defined as

$$\theta_{t,t'} = \sum_i ((\psi_{i,t} - \psi_{i,t'}))^2 \tag{2.3}$$

  where $\psi_{i,t}$ is the value at time t of the i-th ψ dihedral angle. The notation $((\bullet))$ stands for 2π-periodicity within the brackets;

- Space of the contacts between residues [48]: the configuration of a protein is defined by a $n_{res} \times n_{res}$ contact map-matrix C, where n_{res} is the number of residues of the protein. Each matrix element C_{ij} is equal to one if residue i is in contact with residue j, otherwise it is equal to zero. The contacts are defined according to a cutoff distance. The contact-map distance between configuration t and t' is

$$d_{t,t'} = \sum_{(i,j)} \sqrt{(C_{ij}(t) - C_{ij}(t'))^2} \tag{2.4}$$

  where $C(t)$ is the contact matrix of configuration t.

- Space of the backbone atoms [48]: the configuration of a protein is defined by the X-Y-Z positions of the backbone atoms. The distance between configuration t and t' is the RMSD distance:

$$d_{t,t'} = \min_{R,t} \Big[\sum_i (\sqrt{(x_{i,t} - \tilde{x}_{i,t'})^2 + (y_{i,t} - \tilde{y}_{i,t'})^2 + (z_{i,t} - \tilde{z}_{i,t'})^2}) \Big] \tag{2.5}$$

  where the sum is over all backbone atoms, $(x, y, z)_{i,t}$ are the coordinates of the i-th backbone atom at time t and $\tilde{x}_{i,t'} = t + Rx_{i,t'}$, with t a translation vector and R a rotation matrix. In other words, before calculating the distance between two configurations, we look for their best superimposition obtained through a rigid rototranslation.

Let's remark that these coordinates are supposed to preserve all the relevant information of the trajectory. The choice of some feature coordinates is necessary to have a more compact numerical representation of the trajectory. However, this preliminary step is still supervised: some knowledge of the system is required to be able to select the degrees of freedom which are likely to be irrelevant (for example the coordinates of the solvent molecules).

2.2 Estimating the Intrinsic Dimension

Each configuration of a biomolecule, is now defined by the value of the coordinates defining the feature space. The number of these coordinates is however still high, if compared to the effective number of directions in which a molecule can move on

long timescales [50]. There are indeed many restraints both of chemical and physical nature that reduce the effective dimension of the manifold in which the trajectory lies. It is therefore important to be able to evaluate the so called intrinsic dimension of a dataset(ID), which is the minimum number of variables that are needed to capture all the relevant features of a data landscape without significant information loss. Moreover, the knowledge of the ID is a preliminary step for calculating the free energy of each data point through the PAk estimator, as we will discuss in section 2.3.2. Several approaches have been developed for the estimate of the ID of generic datasets [51–55]. In this work we use the TWO-NN estimator [55], which is summarized in the following.

Let i be a point of the dataset, and $r_1, .., r_k$ the sorted list of the distances between i and its first k neighbours. The volume of the hypersferical shell enclosed between two successive neighbours $l-1$ and l is given by:

$$v_l = \omega_d (r_l^d - r_{l-1}^d) \tag{2.6}$$

where d is the dimensionality of the embedding space and ω_d is the volume of the sphere with unitary radius in the d-dimensional space. It can be proved (see [55] for a derivation) that, if the density(ρ) is constant around point i, all the v_l are independently drawn from an exponential distribution:

$$P(v_l \in [v, v+dv]) = \rho e^{-\rho v} dv \tag{2.7}$$

This result is at the base of the TWO-nn estimator. Let's now consider two shells v_i and v_j, and let R be the quantity $\frac{v_j}{v_i}$, in the case of constant density, eq 2.7 allow us to compute exactly the probability distribution (pdf) of R:

$$\begin{aligned} P(R) &= \int_0^\infty dv_i \int_0^\infty dv_j \rho^2 e^{-\rho(v_i+v_j)} \delta\Big(\frac{v_j}{v_i} - R\Big) \\ &= \frac{1}{(1+R)^2} \end{aligned}$$

This pdf doesn't depend explicitly on the dimensionality d, which appears only in the definition of R. In order to work with equations explicitly depending on d we define the quantity $\mu \doteq r_2/r_1$ which is the distance between the second and the first

neighbours of point i. Fixing $i = 1$ and $j = 2$, R and μ are related by equation:

$$R = \mu^d - 1$$

This equation allows to find an explicit formula for the pdf of μ:

$$P(\mu) = d\mu^{-d-1}\chi_{[1,+\infty]}(\mu)$$

where $\chi_{[1,+\infty]}(\mu) = 1$ if $\mu \in [1, +\infty]$, 0 otherwise.
The cumulative distribution(cdf) of μ is then obtained by integration:

$$F(\mu) = (1 - \mu^{-d})\chi_{[1,+\infty]}(\mu) \tag{2.8}$$

Importantly, $F(\mu)$ depends explicitly on the intrinsic dimension d, but it is independent of the density ρ.
The value of the intrinsic dimension d can be estimated through the equation:

$$d = \frac{log(1 - F(\mu))}{log(\mu)} \tag{2.9}$$

Equation 2.9 allows estimating the ID of a dataset of N points. $F(\mu)$ is empirically estimated: μ_i is calculated for each point i of the dataset. Then the values of μ are sorted in ascending order, this gives $F^{emp}(\mu_{sorted_i}) = i/N$. Then, to each point i of the dataset, one associates a point in the R^2 plane having as coordinates $x_i = log(\mu_i)$ and $y_i = -log(1 - F^{emp}(\mu_i))$. The ID of the dataset is obtained fitting these N points in the x-y plane with the straight line $y = dx$ passing through the origin.

Equation 2.9 is theoretically exact if the density is locally constant, which means if the density is constant in the range of the first two nearest neighbours of each point. This happens in the limit of a dataset containing infinite points, however in Ref. [55] it is shown that in the case of finite datasets the TWO-NN estimator is numerically consistent. In general, a strong advantage of this estimator with respect to standard ID estimators [51–54], is the fact that the density is required to be constant only in

the range of the second neighbour. As a consequence inhomogeneities in the density and the space curvature of the manifold containing the data do not have a great impact on the measured ID.

2.3 Free Energy Estimate

As we mentioned in the introduction of this chapter, the free energy calculation, for each frame of the trajectory, is one of the two key steps in our protocol for analysing free energy landscapes.

In a MD simulations at a fixed temperature T, the probability distribution as a function of the coordinates defining the feature space(y) is given by equation:

$$\rho(y) = \int dx \rho(x) \delta(y - Y(x)) \tag{2.10}$$

where $\rho(x) = \frac{1}{Z} exp(-V(x)/K_B T)$, and $Y(x)$ is the function for obtaining the feature coordinates from the atoms positions. The free energy is then defined as

$$F(y) = -K_B T log(\rho(y)) \tag{2.11}$$

The strategy for estimating $F(y)$ we use in this work is an adaptation of algorithms from data science which estimates the probability of each point of a dataset [14–17,56]. The free energy is then obtained by equation 2.11. The probability of a single point, is estimated as its local density, which means that a point is more probable if it lies in a region of the embedding manifold in which there are many other points. Among popular density estimators there are the k-Nearest Neighbours estimator k-NN [17] and, its evolution, the Point Adaptive k-Nearest Neighbours estimator PAk [56], which we will briefly describe in the following.

2.3.1 The k-NN Density Estimator

The k-NN estimator measures the density of a point i of a dataset using the following equation:

$$\rho = \frac{k}{V_{i,k}} \tag{2.12}$$

where k in the number of considered neighbours of point i and $V_{i,k}$ is the volume occupied by these neighbours. The error associated to this measure is:

$$\varepsilon_\rho = \frac{\rho}{\sqrt{k}} \tag{2.13}$$

We here present the demonstration of equation 2.12, taken from Ref. [56].

We consider the same situation described in section 2.2, we will briefly recall it for clarity. $\{X_1, .., X_N\}$ are a set of N vectors in the $\mathbb{R}^D$ space, lying in a manifold of Intrinsic Dimension d (with $d \leq D$), which is constant for all the set. We consider a point i of the dataset, and we define $v_{i,l}$ the volume of the hyperspherical shell, enclosed between neighbours $l-1$ and l. As we said in section 2.2, if the density(ρ) is constant around a point i, the probability distribution of the volume $v_{i,l}$, is an exponential distribution:

$$P(v_{i,l}) = \rho e^{-\rho v_{il}} \tag{2.14}$$

Therefore, the log-likelyhood function of the parameter ρ, given the observation of the k-nearest neighbours distances from point i is

$$\mathcal{L}(\rho \mid \{v_{i,l}\}_{l\leq k}) \doteq \mathcal{L}_{i,k}(\rho) = log(\prod_{i=1}^{k} \rho e^{-\rho v_{il}}) = klog(\rho) - \rho \sum_{i=1}^{k} v_{il} = klog(\rho) - \rho V_{i,k} \tag{2.15}$$

where $V_{i,k}$ is the volume of the hypersphere centered at i containing k data points. By maximizing $\mathcal{L}$ respect to ρ, we find $\rho = k/V_{i,k}$ as in equation 2.12. The error on ρ is the asymptotic standard deviation of the parameter estimate [56]:

$$\varepsilon_\rho = \frac{\rho}{\sqrt{k}} = \frac{\sqrt{k}}{V_{i,k}} \tag{2.16}$$

Equation 2.12 has been derived under the assumption of constant density and the resulting estimate of ρ is strongly dependent on the k parameter. This parameter assumes thus a precise role: k is the number of neighbours for which the density is (or can be approximated as) constant. The choice of a global k can be difficult in the situation of highly non homogeneous dataset. Let's consider as an example the distribution of points in the x-y plane of figure 2.1 . The choice $k = 20$ which is

optimal for the blue point, is not a good one for the green point. Indeed, for the green point a much larger k could be chosen in order to have a better statistic. On the other hand the choice of $k = 375$, which is optimal for the the green point, leads to a wrong estimate for the blue one since the condition of constant ρ is not respected.

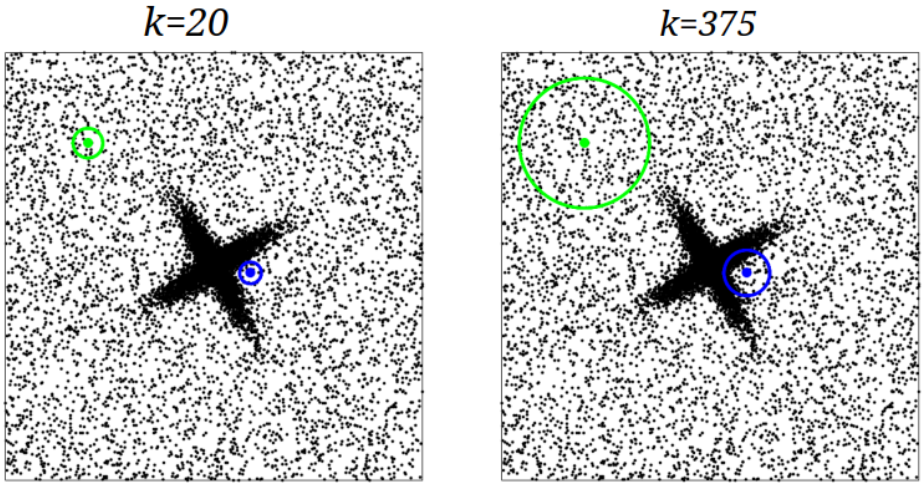


FIGURE 2.1: Example of the issues arising from the use of a fixed k in the k-NN algorithm. In both panels the same blue and green points are selected. In the left panel the circles around these points contain the first 20 neighbours, in the right panel the first 375.

Another clarification has to be made: in the original k-NN formulation the volume of the hypershells is not measured in the reduced space of the manifold in which the data lie, but in the configuration space ($\mathbb{R}^D$).

2.3.2 The PAk Estimator

The PAk estimator is "point-adaptive", meaning that the value of parameter k is optimized for each point of the dataset. k is chosen as the largest possible value for which there is a high level of confidence of respecting the hypothesis of constant density. In detail: for each point i, increasing the k value, the comparison of the maximum likelyhood of two models is performed:

- in the first model(M1), the densities of point i and of its $(k+1)$ nearest neighbour are considered different. Thus, the likelyhood of M1 ($\mathcal{L}_{M1}$, defined as in eq 2.15) has to be maximized with respect to two different parameters (the two densities ρ and ρ_1).

- in the second model(M2), the densities of point i and of its $(k+1)$ nearest neighbour are considered equal. Thus, the likelyhood of M2 ($\mathcal{L}_{M2}$, defined as in eq 2.15) has to be maximized with respect to a single parameter(ρ).

For each point i, the optimal k (denoted as $\hat{k}_i$) is chosen as the one for which the models M1 and M2 can be distinguished at a prefixed level of confidence. In detail, a statistical test [57] is performed to compare the two models, according to the procedure:

1. Evaluation of the difference $D_k = -2(\mathcal{L}_{M2} - \mathcal{L}_{M1})$

2. Search of the $\hat{k}_i$ for which

$$(D_k < D_{thr} \forall k \leq \hat{k}_i) \;\; \& \;\; (D_{\hat{k}_i+1} > D_{thr}) \tag{2.17}$$

   D_{thr} is chosen such that the p-value associated to the statement of distinguishable models is $p = 10^{-6}$

We will now present an example from Ref. [56], in which the whole procedure of k optimization is presented for the 2D points distributions shown in the upper panels of figure 2.2. The distribution of panel A is an uniform distribution of 2000 points. For this distribution, the value of D_k for a chosen point (the orange one) does not change increasing k (panel C). Moreover the value of D_k is always much lower than D_{thr}(green line). On the other hand, the distribution of panel B is obtained by the addition, on the 2000 uniformly scattered points, of 2000 points generated from a gaussian distribution. In panel D, we see that the value of D_k for the selected point (the orange one), increases with the increase of k and at $k \simeq 150$ it becomes bigger than D_{thr}.

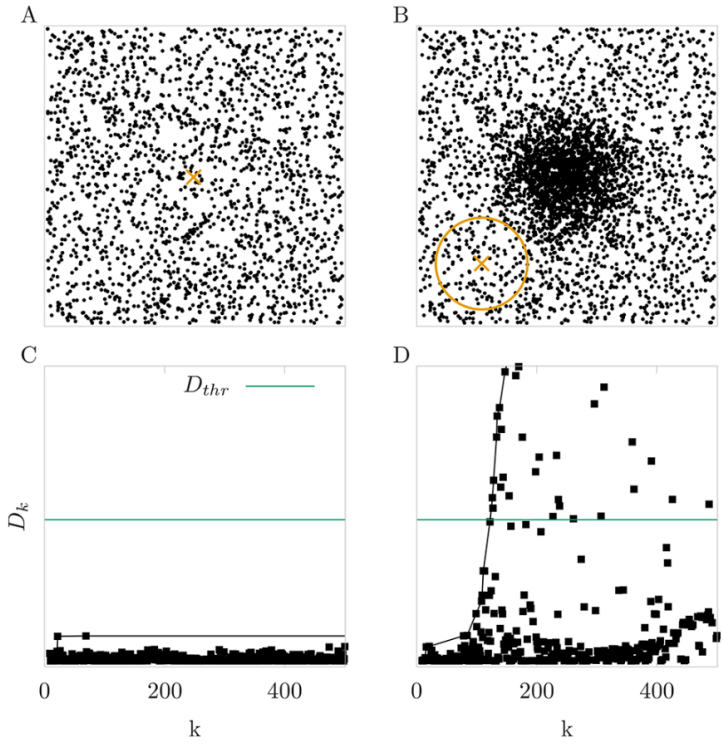


FIGURE 2.2: taken from Ref. [56]. a,b) Sample of 2000 points extracted from a uniform distribution and the same sample with 2000 additional points extracted from a Gaussian distribution. c,d) D_k evolution as a function of the value of k, for the two points highlighted in orange in panels a and b. The green line corresponds to the threshold(D_{thr}) presented in the text.

Let's now go back to the original aim, which was the free energy estimate, for each data point. Having found, for each point i the optimal $\hat{k}_i$, the corresponding free energy could be evaluated as

$$F = -log(\rho) = -log(\frac{\hat{k}_i}{V_{i,\hat{k}_i}}) \tag{2.18}$$

(Setting $K_B T = 1$). The estimate of the free energy in PAk is, in the truth, a bit more complicate than the one presented in equation 2.18. Indeed, the estimator 2.18 is affected by small sistematic errors [56]. This is due to the fact that, at the exit value $\hat{k}$, the models M1 and M2 are distinguishable with a high degree of confidence. The density at the $\hat{k}$-th neighbour is thus likely to be substantially different from the density at point i. To solve this bias, one uses a likelyhood model directly depending on the free energy(F) and on an extra parameter called a. This parameter aims at describing the linear trend in the free energy, moving away from the central point i.

The final estimator is thus given by equation:

$$log(\rho_i) = -\arg\max_{F,a}\left(-F\hat{k}_i + a\frac{\hat{k}_i(\hat{k}_i+1)}{2} + \sum_{l=1}^{\hat{k}_i} v_{i,l}e^{-F+al}\right) \tag{2.19}$$

From this last step also an expression of the the asymptotic standard deviation of F_i is obtained:

$$\varepsilon_i = \sqrt{\frac{4\hat{k}_i+2}{(\hat{k}_i-1)}\hat{k}_i} \tag{2.20}$$

Let's underline, once again, that in contrast to what happens in the original k-NN estimator, the volumes are measured on the manifold on which the data actually lie (of ID=d). Thus the PAk algorithm, depends on the knowledge of this ID, which is calculated as a first step using the TWO-NN estimator (described in section 2.2). The free energy is then estimated without the need of specifying explicitly the collective variables that define this reduced space. It is only necessary to define a metric, which means to define a proper distance between two data points. This metric is defined in the feature space introduced in section 2.1.

2.4 Dataset Topography

In the previous sections we illustrated how one can estimate the probability distribution from a MD trajectory, without performing an aggressive dimensional reduction by choosing a collective variable but using instead an extended feature space. The following step is obtaining useful information about the free energy landscape by the study of the peaks of this distribution (which correspond to the free energy minima) and of the saddles between them (which correspond to the transition states). This is performed using the density-based clustering technique described in section 2.4.1. Importantly, we also develope a new clustering technique, described in section 2.4.2, able to identify flat regions of the free energy landscape. These flat regions have indeed well defined kinetic properties as we will explain in the following.

2.4.1 Density-Peak Clustering

Reference [58] presents a procedure which aims at obtaining an automatic topography of the data set. The results of this analysis are: the location of the free energy

minima, the free energy value at the minima, the location and height of the saddle points separating these minima. The free energy minima are considered as centers of clusters containing similar data points.

The procedure is an extension of the Density Peak clustering algorithm (DP), described in Ref. [20], in which a peak of the density is characterized by two relevant features: it is surrounded by points with lower density and it is at a "sufficiently" high distance from other points with high density. The main improvement of the DP algorithm, is its combination with the PAk estimator presented in section 2.3.2, which makes DP unsupervised. We here summarize the steps of the procedure:

1. Free energy evaluation: for each point i the free energy(F_i) and its uncertainty(ε_i) are evaluated using PAk algorithm. These measures are based on the calculation of the radius $r_{\hat{k}_i}$ which determines the neighborhood in which the free energy can be considered constant.

2. Minima detection: all points which are a local minima of

$$g_i = F_i + \varepsilon_i \tag{2.21}$$

   are considered putative centers (using this equation points with a large error are penalised). A first selection is then made by checking two conditions:

   - $\delta_i > r_{\hat{k}_i}$ where δ_i is the distance to the nearest point with lower g
   - A center cannot belong to the neighbourhood of a point with lower g

3. Point Assignation: in order of increasing g, all the points are assigned to the same cluster of their nearest neighbour of lower g.

4. Saddle Points Detection: The saddle point between two clusters a and b is the one with the lowest g among the points which are at the border between a and b. A point i, belonging to a, is part of the border between cluster a and b if

   - $r_{ij} < r_{\hat{k}_i}$, where r_{ij} is the distance between point i and point j, which is the closest neighbour of i belonging to b
   - r_{ij} is the smallest distance to i, among the distances of all the points belonging to b.

5. Merging of the clusters which are not meaningful: cluster a is merged with cluster b if

$$(F_{ab} - F_a) < Z(\varepsilon_{F_a} + \varepsilon_{F_{ab}}) \tag{2.22}$$

where F_{ab} is the free energy of the saddle point between cluster a and b, with its uncertainty $\varepsilon_{F_{ab}}$; F_a is the free energy of the center of cluster a, with its uncertainty ε_{F_a}.

This means that clusters a and b are considered indistinguishable if the center of a or b has a free energy which is comparable, within its errors, with the free energy of the border between a and b. The constant Z sets the statistical confidence at which clusters are considered meaningful. Z is a free parameter of the procedure, but it has a well defined statistical meaning. Thus Z must be chosen according to the quality of the sampling of the probability distribution: the better is the sampling, the lower the value of Z that can be chosen without choosing as cluster centers spurious peaks.

2.4.2 $\hat{k}$-Peaks Clustering

As we will show in section 4.4, the procedure described in the previous paragraph is not an appropriate tool to perform clustering on systems in which some of the metastable states are stabilized by conformational disorder. For example, this is the case of MD simulations of protein folding, where the system performs a transition from a disordered state (corresponding to the unfolded protein) to an ordered one (corresponding to the folded protein). We do not expect to find local minima of the free energy corresponding to the unfolded states. Still these states are metastable in the sense that, on average, the system spends a long time in them before reaching a different state. In order to study these systems, we thus devised a new clustering technique able to detect both categories of metastable states. This technique is the most important algorithmic development presented in this thesis.

The new procedure is still strongly associated to the PAk free energy estimator. In particular a key role is played by $\hat{k}_i$ which is, for each frame i, the number of neighbours $\hat{k}_i$ for which the free energy can be considered constant within a given level of confidence. The idea is that there are two situations in which $\hat{k}_i$ assumes high

values. The first one is in the free energy minima, where the high density of points leads to a high value of $\hat{k}_i$. The second one is in the flat regions of the free energy landscape, where the low variation of the density of points leads also to high values of $\hat{k}_i$. Therefore, we propose that in order to characterize the kinetics of a system in which at least one state is stabilized by conformational disorder it is convenient to look for the peaks of $\hat{k}_i$. The approach for finding the clusters is similar to the one described in section 2.4.1, with the optimal number of neighbours $\hat{k}$ playing the role of the free energy in the original implementation. The centers of the clusters therefore are the local maxima of $\hat{k}$. For clarity we will present all the steps of the algorithm, even if some of them are identical to the ones presented in 2.4.1:

1. Estimate $\hat{k}_i$ and $r_{\hat{k}_i}$ for each frame. $r_{\hat{k}_i}$ is the radius of the neighborhood in which the free energy is approximately constant.

2. Estimate of the uncertainty σ_i of $\hat{k}_i$ as the standard deviation of $\hat{k}$ among the points which are inside the constant density neighborhood of point i.

3. Find the peaks of $g_i = \hat{k}_i - \sigma_i$. A local maximum of g_i (defined putative center) is a cluster center if two conditions are satisfied:

   - $\delta_i > r_{\hat{k}_i}$ where δ_i is the distance from the nearest point with higher g
   - it does not belong to the neighbourhood of a point with higher g

4. Point Assignation: in order of decreasing g, all the points are assigned to the same cluster of their nearest neighbour of higher g.

5. Saddle Points Detection: The saddle point between two clusters a and b is the one with highest g among the points which are at the border between a and b. A point i, belonging to a, is part of the border between cluster a and b if

   - $r_{ij} < r_{\hat{k}_i}$, where r_{ij} is the distance between point i and point j, which is the closest neighbour of i belonging to b
   - r_{ij} is the smallest distance to i, among the distances of all the points belonging to b.

6. Merging of the clusters which are not meaningful. In particular, cluster a is merged with cluster b if:

$$\hat{k}_a - \hat{k}_{ab} < Z(\sigma_{\hat{k}_a} + \sigma_{\hat{k}_{ab}}) \tag{2.23}$$

where $\hat{k}_a$ is the optimal number of neighbours of the center of cluster a, $\hat{k}_{ab}$ is the optimal number of neighbours of the saddle point between cluster a and b and $\sigma_{\hat{k}_a}$,$\sigma_{\hat{k}_{ab}}$ are the corresponding uncertainties. Z is a free parameter of our approach.

The results of $\hat{k}$-Peaks clustering are a central topic of this thesis and will be presented in chapter 4.4.2 .

2.5 Kinetics: Markov State Models

We here present the background theory regarding the study of the kinetics of MD simulations, which we perform, in our protocol, after the topography of data set has been built.

A useful method to obtain information about the kinetic of processes, from MD simulations, is the construction of Markov State Models(MSMs) [34–36]. In MSMs the kinetics is assumed to be a memoryless jump process between a number of states in which the trajectory has been previously mapped. Using MSMs is possible to describe the long-time dynamics of biomolecules [3, 39, 42–47]. We here present the fundamentals of MSMs theory (more details can be found in Ref. 34).

Let $x(t)$ be a time discrete Markov process in the Ω state space (in general containing both positions and velocities). Let's assume that $x(t)$ is ergodic, meaning that if the trajectory was infinitely long, all the states x would be visited an infinite number of times. The equilibrium probability density of the system, $\rho(x)$, is thus the fraction of time the system spend in state x during a infinitely long trajectory and it is unique.

Let $p(x,t)$ be the probability to observe the system in a certain configuration x. At time t, the time evolution of $p(x,t)$ is determined by the equation

$$p(x, t+dt) = \int dx' p(x', t) \Pi_{dt}(x' \Rightarrow x) \tag{2.24}$$

where $\Pi_{dt}(x' \Rightarrow x)$ is the Markov operator which is in practice a conditional probability: $\Pi_{dt}(x' \Rightarrow x) = P(x, t+dt \mid x', t)$. Indeed, for the kinetics obtained by molecular dynamics (also from MonteCarlo and Langevin dynamics) Π_{dt} is time independent, but it depends on the time lag dt. The Markov operator Π_{dt} has the following properties:

1. It is positive defined: $\Pi_{dt}(x \Rightarrow x') \geq 0 \ \forall x, x'$
2. It is normalized: $\int dx' \Pi_{dt}(x \Rightarrow x') = 1$
3. It satisfies the detailed balance condition: it exist a probability distribution at equilibrium $\rho(x)$ such that:

$$\Pi_{dt}(x' \Rightarrow x)\rho(x') = \Pi_{dt}(x \Rightarrow x')\rho(x) \tag{2.25}$$

   This means that the flux from x to x' is equal to the flux from x' to x. This last property is satisfied in MD, MonteCarlo and Langevin dynamics.

We are interested in finding the solution of equation 2.24. We search it as a combination of and eigenvectors (Ψ_i) of the operator Π_{dt} :

$$p(x,t) = \sum_i \Psi_i(x) c_i(t) \tag{2.26}$$

with (Ψ_i) defined by the equation:

$$\int dx' \Pi_{dt}(x' \Rightarrow x)\Psi_i(x') = \lambda_i \Psi_i(x') \tag{2.27}$$

where (λ_i) is the eigenvalue associated with $\Psi_i(x)$. If the conditions 1),2),3) and ergodicity are satisfied, then the Perron-Frobenius theorem [59] guarantees that it exists an eigenvalue $\lambda_0 = 1$ and all the others eigenvalues satisfy $\lambda_i \in (0, 1[$. We can write the solution of equation 2.24 as

$$p(x,t) = P^{eq}(x) + \sum_{i>0} e^{-\frac{t}{\tau_i}} \Psi_i(x) c_i(0) \tag{2.28}$$

where

$$\tau_i = -\frac{dt}{log(\lambda_i)} (i > 0) \tag{2.29}$$

represents the time associated to the i-th relaxation process in the system [34].

Let's analyse the solution 2.28, considering at first the long-time limit where $t \Rightarrow \infty$. In this situation only the term $\rho(x)$ survives. All other terms (whose corresponding eigenvalues are $\lambda_i < 1$) decay over time. The associated eigenfunctions describe the dynamical rearrangements taking place while the system relaxes toward the equilibrium distribution. Indeed, the closer λ_i is to 1, the higher is τ_i meaning that the slower is the corresponding relaxation process. We can thus introduce the concept of metastability [60]: a system is metastable if it possible to detect a large gap between a certain number of eigenvalues $\lambda_i \simeq 1$ and all the others which are smaller. The eigenvectors correspondent to the $\lambda_i \simeq 1$ are the ones sufficient to describe relaxation to equilibrium. All the other terms are rapidly going to zero.
In particular, the change of signs of $\psi_i(x)$ describes the relaxation process associated to time τ_i. If the sign of $\psi_i(x)$ is negative for x belonging to the set x_{neg} and positive for x belonging to the set x_{pos} the studied process is a transition from these two sets.

Since the operator Π_{dt} is not hermitian, the calculation of its eigenvectors and eigenvalues can be efficiently performed through the auxiliary hermitian operator h:

$$h(x \Rightarrow x') \doteq \Pi(x \Rightarrow x')\sqrt{\frac{\rho(x)}{\rho(x')}}$$

Indeed, it can be proved [34] that the two operators have the same eigenvalues, and that the eigenvectors of h (we will denote them as $\Phi(x)$) are related to the ones of Π (denoted as $\Psi(x)$), through the equation:

$$\Phi_\alpha(x) = \frac{\Psi_\alpha(x)}{\sqrt{\rho(x)}}$$

From the practical point of view a MSM can be built in two main steps:

1. mapping of the trajectory $x(t)$ (lying in the high dimensional continuous phase space), in a number n of clusters $C_1, .., C_n$. The temporal evolution the atom positions $x_1..x_N$, where N is the configuration number, is thus replaced with the temporal evolution of the cluster index $C_1..C_N$

2. Calculation of the counting matrix (C_{dt}). Each matrix element $C(ij)$ is estimated from the trajectory, counting the number of times the system is observed in cluster i at time t and in cluster j at time $t+dt$. Once normalized, this matrix estimates the transition matrix:

$$\hat{\Pi}(ij) = \frac{C(ij)}{C_i} \tag{2.30}$$

where C_i is the total number of times the trajectory was in state i. $\hat{\Pi}_{dt}$ is an estimate of the Markov operator Π_{dt}

Let's remark that, in the most common procedures to obtain a MSM, the first step involves first grouping the conformations in a high number of microstates through conformational clustering normally performed with k-means algorithm [37] or the ward algorithm [38]; and then grouping the microstates in the so called "Markov States" using dynamical clustering([39–41]). The Markov states correspond to the metastable states of the system, obtained in the second step. As we will see, in our protocol these steps are not necessary, and the Markov states are directly obtained from the analysis of the free energy landscape.

2.5.1 Model Validation

The dynamics of a system, described through the positions and velocities of all the atoms(both of the solute and of the solvent), is certainly Markovian because the next conformation is simply a deterministic function of the current state of the system [36]. The Markovianity of the system, however, can be broken if the discrete partition of state space in not performed properly [36]. Indeed, grouping together conformations that don't belong to the same free energy basin can create states with large internal free energy barriers. Such states will violate the Markov property because a system that enters the state on one side of the barrier will behave differently from a system that enters on the other side, thus introducing history dependence. We here present two tests of the validity of a MSM.

The first test is the analysis of the dependence of the implicit timescales of the system, given by equation 2.29 from the time lag dt. Indeed, the estimator of the matrix Π_{dt} in equation 2.30, depends implicitly on the parameter dt. However, it can

be proved that, if the conformation space partition has been done in a correct way, there should be an interval of the time lag dt, in which all time scales are invariant [36].

The second test is based on the Chapman-Kolmogorov equation:

$$[\Pi_{dt}]^k = \Pi_{kdt} \tag{2.31}$$

This equation captures the fact that, if markovianity is satisfied, taking k steps with a MSM of lag time dt should be equivalent to taking a single step with a MSM of lag time of kdt . We thus expect to obtain the same transition matrix following these two these procedures

1. Directly counting from the trajectory the number of transitions between states, with a lag time of kdt

2. Scaling the transition matrix evaluated at lag time dt. The scaling of $\hat{\Pi}_{dt}$ is performed from its eigenvalues(λ) and left and right eigenvectors($\Psi^{left}, \Psi^{right}$):

$$[\Pi_{dt}]^k(ij) = \sum_{\alpha} \lambda_{\alpha}^k \Psi_i^{\alpha,left} \Psi_j^{\alpha,right} \tag{2.32}$$

where the sum is over all eigenvalues of $\hat{\Pi}_{dt}$.

The Free Energy Landscape of the SARS-CoV-2 Main Protease

In this second chapter of the thesis, we exploit the techniques introduced in chapter 2 to analyse the free energy landscape of the main protease of the coronavirus SARS-CoV-2. The aim of this analysis is identifying the metastable states of the protein and, based on the structure of these states, propose a possible strategy to block the action of this protein through allosteric inhibiton.

The severe acute respiratory syndrome, which has broken out in December 2019 (COVID-19), is caused by coronavirus 2 (SARS-CoV-2) [61, 62]. Its main protease (M^{pro} or $3CL^{pro}$) was the first protein of SARS-CoV-2 to be crystallized, in complex with a covalent inhibitor, in January 2020 [63]. It is essential in the viral life cycle since it operates at least eleven cleavage sites on large viral polyproteins that are required for replication and transcription [63, 64], so it is an attractive target for the design of antiviral drugs [65]. Since there is no known human protease having a cleavage specificity similar to the one of M^{pro}, it may be possible to design molecules that do not interact with human enzymes [63, 64].

M^{pro} is a homodimer. Each monomer has 306 residues and is composed of three domains. Domains I and II (residues 10-99 and 100-182, respectively) have an antiparallel β-barrel structure. The binding site of the substrate is enclosed between these β-sheets [64]. Domain III (residues 198-303) contains five α-helices and has a role in the regulation of the protein dimerization [64]. The two residues His^{41} and Cys^{145} form the catalytic dyad. The structure and way of functioning of the SARS-CoV-2 M^{pro} are similar to the ones of the SARS-CoV M^{pro} [66, 67]. This is expected,

due to a 96% sequence identity between them.

The most direct strategy to block the action of the M^{pro} is through small molecules that directly interact with the catalytic site. The first *in silico* trials were made with covalent inhibitors known to be interacting with the catalytic site of SARS-CoV M^{pro} such as N3 [63] or 11r [64]. Many efforts followed in the field of virtual screening. In this kind of studies, computational docking of millions of molecules is performed, the behaviour of the best candidates is usually then tested through MD simulation [68–73]. Another possible route that can be followed to stop the action of the M^{pro}, is allosteric inhibition [74, 75]. The idea is to block the protease in one of its metastable conformations, in which the catalytic dyad cannot regularly operate, inhibiting in this way the whole protein functionality. This approach, at least in principle, has several advantages. First of all, it offers the possibility to drug sites far from the catalytic pocket, thus enlarging the chance to discover active compounds and to obtain non-competitive inhibition. If an allosteric site is identified and targeted, using this strategy one can develop drugs which are highly specific since they do not bind in active sites, which are typically conserved in protein families [76]. Owing to these advantages, allostery has been established as a mechanism for drug discovery, for example to target G-protein-coupled receptors(GPCRs) [77, 78] or protein kinases [79–81].

In this chapter we describe a strategy to identify candidate binding sites for allosteric inhibition which is fully based on the analysis of a long molecular dynamics trajectory. We analyze a $100\mu s$ MD trajectory of the M^{pro} generated in the D. E. Shaw Lab [82]. We use the approach described in chapter 2 to search for possible metastable states of the protease, namely configurations which do not change significantly on the scale of several tens of ns. These configurations are important for developing drugs for allosteric inhibition, since they are already (marginally) stable, and by designing a ligand which increase their stability they can become kinetic traps [76].

The first step of our procedure is the estimate of the free energy of each data point, in its high dimensional space, using the PAk algorithm explained in section 2.3.2. The following step is the application of a clustering algorithm. At the simulated temperature, the M^{pro} explores the basin of the main free energy minumum corresponding

to native state, without performing large scale conformational changes. This is evident simply from a visual inspection of the MD trajectory. The possible metastable states of the system thus correspond to the local free energy minima on the "walls" of the global minimum, if deep enough. We look for these local minima using the unsupervised version of the Density Peak clustering algorithm, described in section 2.4.1. We apply the Density Peak clustering and not the $\hat{k}$-Peaks clustering, since we don't expect the presence of entropic traps corresponding to large flat region of the landscape.

We carry out our analysis in two different spaces: the space defined by all the ψ backbone dihedrals of the protease and the space defined by the contacts between pairs of residues which break or form during the dynamics. Both spaces consider the enzyme globally, not limiting the analysis to the catalytic dyad or to the binding pocket, which is essential to unveil possible allosteric states. Based on a characterisation of global and local properties of these states, we propose a few possible targets which could serve as binding sites for drug-like compounds with the purpose of allosteric inhibition.

3.1 Metric Spaces and Free Energy Estimate

We extract from the $100\mu s$ MD trajectory 10.000 equally spaced frames, one every $10ns$. Since the enzyme is a homodimer, we consider the 20.000 total frames of the two monomer trajectories as a sample of the conformational space of a single monomer. However, the trajectories of the two monomers are considered and analysed separately, in order to verify *a posteriori* whether the configurations they explore are similar or not.

In both metric spaces in which we perform our analysis, we neglect the 10 residues at the C-terminus of the peptide, since they are highly mobile in both monomers and might introduce noise in the analysis. The two metrics are:

- the *ψ-backbone-dihedral distance*, which is the distance defined in equation 2.3, here index i runs between 1 and 296 ;
- the *contact-map distance*, which is the distance defined in equation 2.4. However, we here consider only the contacts which vary significantly during the

simulation. To define these mobile contacts, we first compute the contact-map matrix C for each frame, restricted to residues 1-296. For each couple of residues ij we first evaluate the distances between all the couples of heavy atoms, with one atom belonging to i and the second one belonging to j. C_{ij} is then equal to $\sigma(d_{min})$ where d_{min} is the smallest distance between the couples of atoms, and σ is the sigmoidal function: $\sigma = (1-(d/r_0)^{10})/((1-(d/r_0)^{20}))$, with $r_0 = 4.5\mathring{A}$. Basically, $C_{ij}(t)$ is very close to zero if at time t no atom of residue i is close (in terms of r_0) to any atom of residue j, while is close to 1 if there is at least a couple of atoms of i and j closer than r_0. This procedure defines a total of $(296 \times 296)/2 = 43.660$ independent contacts. We consider as *mobile* the contacts which are completely formed ($C_{ij} > 0.8$) in at least 5% of the frames and completely broken ($C_{ij} < 0.2$) in at least 5% of the frames. Moreover, we neglect those contacts which have a value between 0.2 and 0.8 in more than 50% of the frames. This procedure selects 155 relevant mobile contacts for the first monomer (m1) and 184 for the second (m2). Most of these contacts are in common, as reasonable since the two monomers are chemically identical; the union of the two sets has 235 elements. Denoting by $\mathcal{M}$ the set of mobile contacts of a monomer, the contact-map distance between configuration t and t' is given by equation 2.4, whith $(i, j) \in \mathcal{M}$.

Our two metrics are both sensitive to local and global conformational changes in the peptide, but capture different details: the ψ coordinates keep track of the changes in the protein backbone; the mobile contacts metrics, instead, also keep track of the side-chains rearrangements, while neglecting fluctuations around the completely formed or completely unformed contacts. In summary, we are considering four different datasets, each of 10000 points: in two datasets (one for each monomer) the coordinates of the points are the 296 dihedral angles , in the other two datasets the coordinates of the points are the mobile contacts (155 for monomer1 and 184 for monomer2).

As we mentioned in the introduction, the free energy landscape of each dataset is characterized following the procedure, explained in section 2. First of all, the intrinsic dimension (ID) of the manifold containing the configurations is calculated using the TWO-NN estimator (described in section 2.2). In the spaces of the ψ dihedrals we

get an ID of 28 for m1 and of 26 for m2. In the spaces of the mobile contacts, we get an ID of 17 for both monomers. The free energy F of each configuration is then calculated using the PAk estimator (described in section 2.3.2). Finally, using Density Peak (DP) clustering in its unsupervised variant (described in section 2.4.1), we build a topography of the free energy landscape. We first find the free energy minima and we assign all the frames to one of these minima according to the DP procedure. The set of configurations assigned to a single free energy minimum defines a free energy basin. We then find the saddle point between each pair of basins. The *core set* (CS) of a basin is the set of configurations whose free energy is lower than the free energy of the lowest saddle point of the basin.

The described approach requires choosing the metric and a single metaparameter: the statistical confidence Z at which a basin is considered meaningful, introduced in 2.4.1. In our analysis Z is set to the value Z=1.4, which corresponds to a confidence level of approximately 85 %. This means that we expect to have nearly a 15% of artificially split free energy basins. We verified that, by varying Z around this value, the description does not change significantly: the most populated free energy basins remain approximately unchanged.

3.2 State Definition and Global Observables

In the following analysis we call a state a set of configurations which belong to the core set of the same free energy basins according to both metrics. If, for example, a given basin number found using the dihedral metric is split in two different basins according to the contact metric, in our analysis we will consider two states. As a consequence, our states are structurally uniform according to both metrics. We consider in our analysis only states with a population of at least 8 core state configurations. With this criterion, we identify 11 relevant states in the trajectory of m1 and 7 in the trajectory of m2, for a total of 18 metastable states.

First, we want to make sure that the metastable states detected analysing the m1 and m2 trajectories separately are the same as if we run the algorithm on the merged 20.000 configurations. We check it in the case of the mobile contacts metric. We find that all the clusters involve either only frames from the first monomer or from

the second. There is no relevant cluster that shares structures from both monomers, meaning that in terms of the contact map the configurations of m1 are different from the configurations of m2. Due to their chemical identity, in an ergodic simulation the configurations explored by the two monomers should be nearly identical. Therefore, the first important result of our analysis is that $100\mu s$ of MD simulation are not sufficient to explore ergodically all the configuration space, as recently claimed also by Cocina et al. [83]. This is also visible by looking at figure 3.1: most states are visited only 2-3 times. Consequently, the mean residence time cannot be meaningfully estimated. We instead compute, the maximum residence time, considering it a proxy of the metastability of each state. These times are shown in the upper panel of figure 3.2 and range from $0.20\mu s$ to $16.07\mu s$.

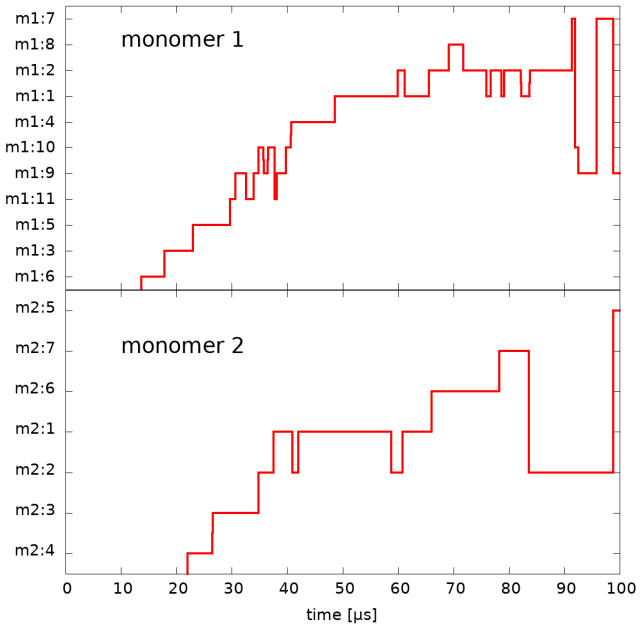


FIGURE 3.1: Trajectories for the two monomers in the space of the states. The frames that do not belong to a core set are relabeled by the state identifier of last visited core state; notice there is no label assigned to the first 10 to 20 μs indicating that no statistically meaningful metastable state is visited in the first part of the trajectory.

Then, to quantify the accessibility to the catalytic site, we estimate the average solvent accessible surface area (SASA) of the dyad, and what we call the *pocket doorway area* (PDA), which quantifies the opening of the catalytic pocket from the position of four selected Cα carbons (fully explained below, for a visual representation see figure 3.3). The SASA and PDA, for each of the 18 states, are presented in the

middle and lower panels of Figure 3.2. These two quantities are in general quite correlated, although not in all the states. Indeed, contrary to PDA, SASA is sensitive to what happens in the direct proximity of the catalytic residues, while neglecting more macroscopic rearrangements of the catalytic pocket.

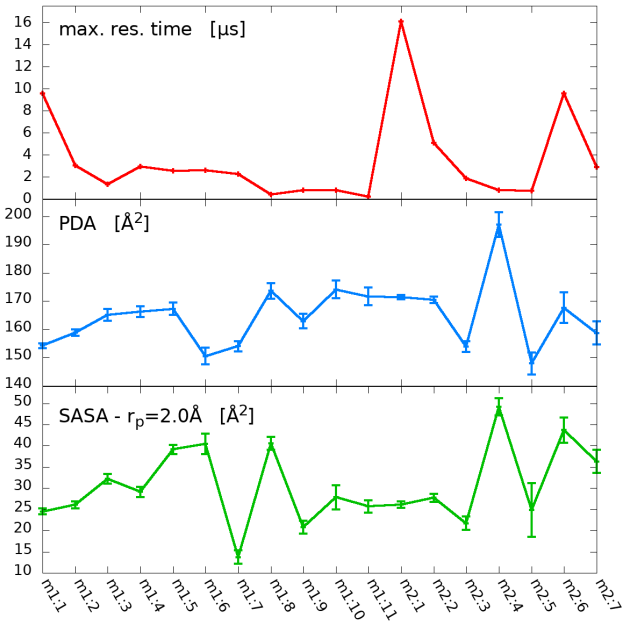


FIGURE 3.2: Global observables of the states. Top: the maximum residence time for each state, taken as the longest time interval over which the state label does not change. Middle: average PDA of the frames belonging to the core of a state. Bottom: average SASA of the catalytic dyad of the frames belonging to the core of a state; the SASA is computed choosing a probe radius $r_p = 2.0\mathring{A}$.

In the VMD visualisation of figure 3.3, we give a pictorial representation of the PDA and of the loops surrounding the binding pocket. The backbone is colored in dark blue and the residues surrounding the catalytic dyad in red. These residues are the ones that shape the enzyme's binding pocket. The most flexible loop surrounding the cavity are represented in light blue: the left and upper flap, the linker and right loop. The segments connecting the five Cα atoms so to define the three triangles whose total area we call PDA are presented in white dashed lines. The segment labels report distances in Å. In detail, PDA is defined as the sum of the area of the three triangles formed by the Cα carbons: Thr^{25}-Ser^{46}-Gly^{143}; Ser^{46}-Gly^{143}-Met^{165}; Gly^{143}-Met^{165}-Arg^{188}.

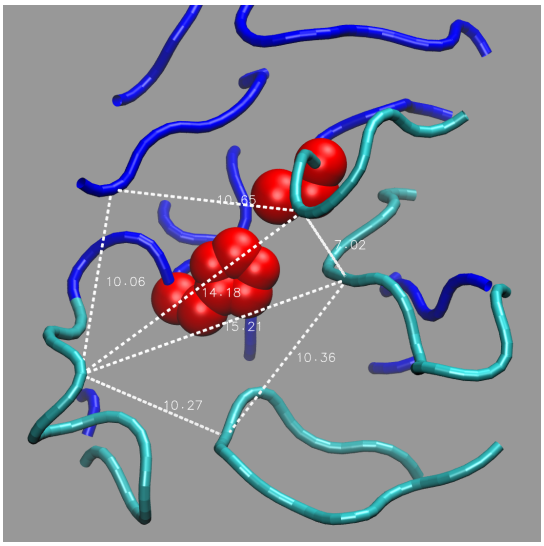


FIGURE 3.3: PDA explained through a VMD visualization

3.3 Description of the States

We then characterise the local differences in the states by analyzing in detail their contact structure and their backbone arrangement. In the case of the mobile contacts, we analyze the intra-monomer contacts which change significantly between at least two of the 18 states; furthermore, we also track the behaviour of few inter-monomer contacts that might reflect some changes in the metastable states' catalytic cativity [66, 84]. The contact structure of the selected states is summarised by the table in Figure 3.4a. As for the backbone, we analyze the ψ dihedral angles in the loops closing the cavity and other few dihedrals which change significantly in the various states (see Table 3.1). In the following, we first focus on the four loops surrounding the binding pocket, presenting the different conformations they adopt in the 18 states. We then fully characterize the eighteen detected states.

3.3.1 Loops Surrounding The Binding Pocket

As mentioned above, the catalytic dyad His^{41}-Cys^{145} is located in the pocket between the protein domains I and II. The access to this cavity is controlled by the flexible loop structures highlighted in Figure 3.4b. The two most flexible loops [85] involve residues from Ile^{43} to Pro^{52} (*left flap*) and from Phe^{185} to Tyr^{201} (*linker loop*). The left flap corresponds to the leftmost loop in Figure 3.4b, and opens and closes like a

small door. No conformers from the second dimer m2 have the left flap wide open, consequently contact Glu^{47}-Leu^{57} is never formed. The linker loop closes the cavity from below in Figure 3.4b and links domains II and III. All the m2 states have a loosely structured linker loop, with contact Arg^{131}-Thr^{199} almost never formed and contact Asp^{197}&Thr^{198}-Asn^{238} always formed.

The loop from Phe^{140} to Cys^{145} (we call it *upper flap*) is smaller and assumes mainly two conformations: tilted downwards (contacts Ans^{28}-Gly^{143}&Ser^{144} and Tyr^{118}-Asn^{142} not formed, dihedral ψ_{144} in β configuration), which hides the catalytic Cys^{145} or flat out (ψ_{144} in α configuration), which leaves more access to the dyad. All m2 states except m2:5 have the upper flap not tilted down and retracted with respect to the pocket, with contact Tyr^{118}-Asn^{142} almost always formed and contact Gly^{138}-His^{172} almost never formed. These two contacts are almost always mutually exclusive, with exception of states m1:6 and m2:5, in which both contacts are formed at the same time.

Last, the β-sheet loop from Met^{162} to Gly^{170} delimits the cavity from the right in Figure 3.4b (we call it *right loop*); it is the least flexible, but it interacts with the N-finger of the other monomer and is crucial for shaping the substrate binding pocket [86].

3.3.2 Structural Description of the Metastable States

We here present a description of all 18 metastable states in terms of their local contact structure and backbone arrangement and of the two observables SASA and PDA.

From the analysis of the maximum residence time it is clear that states 1 and 2 of both m1 and m2 are among the longest-lived metastable states. All four are in fact very similar to the crystallographic structure (PDB 6Y84): they all have the left flap and the linker loop in contact between each other (cont. Met^{49}-Gln^{189}); the left flap is closed (cont. Glu^{47}-Leu^{57} broken, cont. Thr^{25}-Cys^{44} formed) and the linker loop stretched towards it (cont. Leu^{167}-Arg^{188} broken), covering the lower part of the binding pocket. The contact and backbone structures of states m2:1 and m2:2 are almost identical and even a visual inspection with the software VMD confirms the two states can be considered in practice as the same metastable state (even the SASA and PDVA have compatible values within errorbars); the difference between

states m2:1,m2:2 and m1:1,m1:2 is the fact that the latter two have the F140-C145 loop (we call it *upper flap*) tilted downwards (contacts 28 vs 143-144 and 118 vs 142 not formed, dihedral 144 in β instead of α configuration), which hides the catalytic Cys^{145}, resulting in a slightly lower SASA and PDVA. The differences between m1:1 and m1:2, instead, are mostly in the linker loop, which in m1:2 is wider in proximity of the pocket (cont. 185-186 vs 192 not formed) and narrower towards the end (contacts 132 vs 196 and 197-198 vs 238 formed, 131 vs 199 not formed).

Two other states which are similar to each other in terms of contact structure are m2:6 and m2:7. The upper flap is not bent downwards (dihedral 144 in α configuration, as most of the states in m2), leaving some SASA for the catalytic Cys^{145}. In m2:7 the left flap is more stretched towards the linker loop, and the linker loop is open wider, granting slightly lower PDVA and SASA. In both cases, however, the catalytic dyad is quite accessible.

Then there are states m1:9 and m1:10 which are very similar in their contact and backbone structure, with the exception of the left flap, which is much more open in state m1:10. States m1:9 and m1:10 (especially the former) are then both structurally similar to m1:7: the only difference among the contacts is 132 vs 196, which is formed in m1:7 and not formed in m1:9 and m1:10, allowing the lower loop to be more flexible. In all three states the upper flap is tilted downwards; surprisingly, despite the fact that the left flap is wide open, two out of these three states are detected as closed by our observables. In m1:9 the side-chains of the residues in the loops surrounding the binding pocket are oriented towards the catalytic dyad, causing such state to rank among the lowest in SASA; moreover, cont. 285 vs 285* in this state is not completely formed (n configuration). State m1:7 ranks among the lowest in PDVA and as the lowest in SASA; the reason lies in the sidechains of the lower and left flaps, in particular of Thr^{45} and Gln^{189}, which form a contact and effectively close the access to the reactive site.

Another couple of similar states is that of m1:4 and m1:11: they characterised by a very open left flap (cont. 47 vs 57 formed) and the upper flap still tilted downwards. They rank among the most open in PDVA but not very high in SASA, due to the upper flap and to sidechains orientation (especially in m1:11). State m1:4 is among the only three states in which the contact of the dimer interface (cont. 285 vs 285*)

is a little looser than in the others.

The remaining states do not present close similarities to others in terms of contact structure; we describe them in approximate order of decreasing openness of the catalytic pocket. The most open state according to both PDVA and SASA is m2:4; its upper flap is not tilted downwards and is retracted from the pocket, distancing from the β-sheet M162-G170 loop (we call it *right loop*), leaving cont. 138 vs 172 not formed; the left flap is very open (although the dihedrals of this loop are quite variable among the configurations of such state); the linker loop is slightly contracted and wide (cont. 131 vs 199 and 132 vs 196 not formed), not stretching towards the left flap as in other closed or partly-closed states; all of the above play to leave the catalytic dyad well exposed.

State m1:8 also ranks very high in PDVA and in SASA, despite the upper flap tilted downwards. The left flap is very open, although dihedrals 43-46 are not all in α configuration; their particular arrangement ($\alpha\beta\alpha c$), however, grants that the biggest sidechains of the left flap are not oriented towards the binding pocket. The linker loop is not strerched towards the left flap, but rather down, towards the interface with the solvent; it is quite open (dihedral 189 in c instead of β configuration) in proximity of the pocket and all its sidechains do not obstruct the access to the cavity (in particular those of Arg^{188} and Gln^{189}, responsible for a low SASA in other states).

State m1:5 is characterised by an having the left flap open (although less than e.g. state m1:4 and m1:11), with cont. 47 vs 57 formed, and the upper loop not tilted. The right loop leans slightly towards the tip of linker loop (Arg^{188}), causing cont. 138 vs 172 to be broken and cont. 167 vs 188 to be formed between the sidechain of Leu^{167} and the backbone of Arg^{188}. All other contacts far from the pocket are formed. The linker loop leans towards the left flap rather than down.

In state m1:3 the position of the upper flap and of the right loop are approximately the same as in m1:5. The linker loop stretches a bit more toward the left flap, causing contacts 132 vs 196 and 197-198 vs 238 to be broken. The left flap is closed, forming contact 49 vs 189 with the linker loop. The lower part of the pocket results closed, but the catalytic dyad is left quite exposed from above, which yields a central position in both SASA and PDVA ranks.

Also state m1:6 leaves the pocket quite accessible from the top and covered from

the bottom. The linker loop is quite open, while the left flap is closed and stretched towards it. The peculiar shape of the left flap brings the α-carbons of Ser^{46} and Arg^{188} very clpse together, which results in a very low PDVA (second lowest in the ranking).

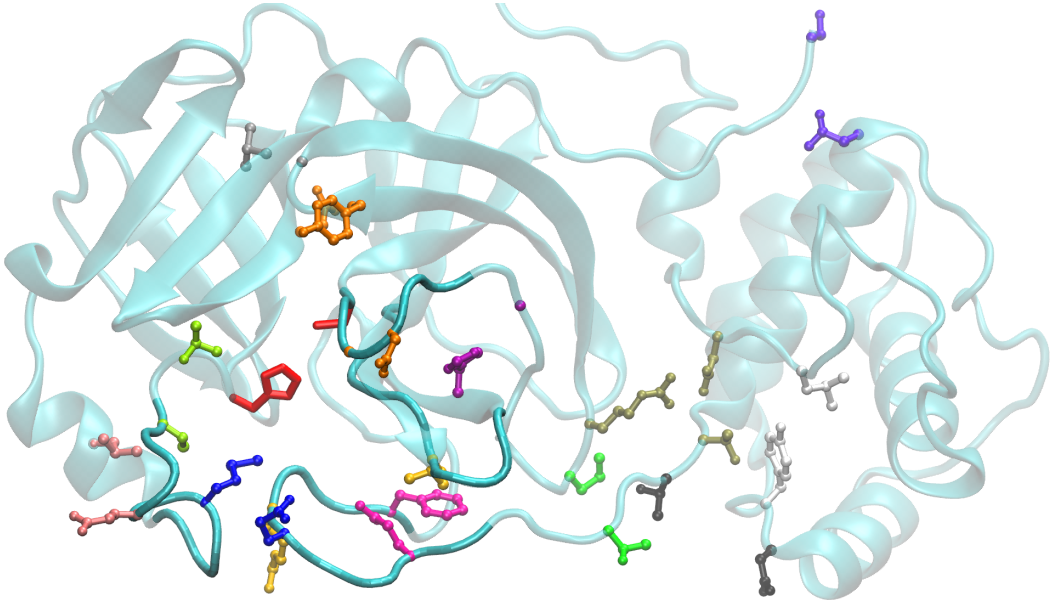
State m2:3 ranks as the third lowest in both SASA and PDVA. Cys^{145} is not well covered, but on the other hand His^{41} is less accessible than in most other states. As most m2 states, m2:3 has the upper flap flat and cont. 138 vs 172 not formed. The linker loop is not stretched, leaving the contacts with residue Arg^{131} unformed or partly unformed. The left flap is really closed and stretched towards the linker loop and its dihedrals are arranged in such a way that cont. 49 vs 189 is not formed; however, these two most mobile loops have a contact between Glu^{47} and Gln^{189}.

Finally, state m2:5 is the one with the lowest PDVA and is among the lowest-ranked in SASA. Its conformation is quite peculiar: the linker loop is all retracted and coiled (it is the only state of m2 forming cont. 167 vs 188). The left flap is all stretched towards the linker loop (cont. 49 vs 189 formed), which, with the contribution of the sidechains, almost completely covers the catalytic His^{41}. The upper flap, rather than being flat or tilted down, is oriented upwards, causing a deformation in the II domain which allows cont. 138 vs 172 to be formed. Remarkably, m2:5 is one of the three states with cont. 285 vs 285^* not tightly formed.

Let's remark that the most closed states are the most relevant when looking for strategies to perform allosteric inhibition, since in these states the catalitic pocket is less accessible. Among them we mention states m1:7,m1:9,m2:3,m2:5.

state ID	Thr^{25} - Cys^{44}	Asn^{28} - Tyr^{118}	Asn^{28} - Gly^{143} & Ser^{144}	Glu^{47} - Leu^{57}	Met^{49} - Gln^{189}	Tyr^{118} - Asn^{142}	Leu^{167} - Arg^{188}	Phe^{185} & Val^{186} - Gln^{192}	Leu^{141} - Gly^{2*}	Gly^{2} - Ser^{214}	Val^{18} - Gly^{120}	Arg^{131} - Thr^{199}	Arg^{131} - Asp^{289}	Pro^{132} - Thr^{196}	Gly^{138} - His^{172}	Asp^{197} & Thr^{198} - Asn^{238}	Tyr^{239} - Leu^{287}	Ala^{285} - Ala^{285*}
m1:1	1	1	0	0	1	0	0	1	0	n	1	1	1	0	1	0	1	1
m1:2	n	1	0	0	1	0	0	0	0	n	1	0	1	1	1	1	1	1
m1:3	0	1	1	0	1	1	0	1	1	1	1	1	1	0	0	0	1	1
m1:4	0	1	0	1	0	0	0	1	0	1	1	1	1	0	1	0	1	n
m1:5	0	1	1	1	0	1	1	1	1	1	1	n	1	1	0	1	1	1
m1:6	n	1	1	0	0	1	0	1	1	n	1	0	n	0	1	1	1	1
m1:7	0	1	0	1	0	0	0	1	0	1	1	0	1	1	1	1	1	1
m1:8	0	1	0	0	0	0	0	1	0	0	1	1	1	0	1	0	1	1
m1:9	0	1	0	1	0	0	0	1	0	1	1	0	0	0	1	1	1	n
m1:10	0	1	0	1	0	0	0	1	0	n	1	0	n	0	1	1	1	1
m1:11	0	1	0	1	0	0	0	1	0	1	1	1	1	0	1	0	1	1
m2:1	1	1	1	0	1	1	0	1	n	1	1	0	n	1	0	1	0	1
m2:2	1	1	1	0	1	1	0	1	n	1	1	0	n	1	0	1	0	1
m2:3	0	0	0	0	0	1	0	1	n	1	0	0	n	1	0	1	1	1
m2:4	n	0	0	0	0	n	0	1	1	1	n	0	1	0	0	1	1	1
m2:5	0	0	0	0	n	1	1	0	n	1	n	n	1	0	1	1	0	n
m2:6	1	1	1	0	n	1	0	1	n	1	1	0	0	1	0	1	0	1
m2:7	n	1	1	0	1	1	0	1	n	1	1	0	0	n	0	1	0	1

(A)



(B)

FIGURE 3.4: a) Selected intra-monomer contacts and inter-monomer contacts (marked with a star(*)). In the case of inter-monomer contacts, the residue of the monomer which is excluded by the metric that defines a state is marked with a star(*). For each contact (columns) the average over the configurations of a given state is reported in the corresponding row. Such contacts are divided into two subgroups by a double vertical line: on the left those between residues belonging to the flexible loops which control the access to the binding pocket and on the right other contacts. For readability, the entries take only three possible labels: 0 when the average over the configurations belonging to a state is < 0.3, namely the contact is not formed; 1 when the average is > 0.7, namely the contact is formed; n in all other case. Contacts whose label does not vary in any of the states of a given monomer are reported in light gray colour. b) In the picture, a VMD [87] representation of monomeric M^{pro} in state m1:1; on the left hand side the enzyme binding pocket, which encloses the catalytic dyad (in red); all other highlighted residue couples refer to the contact with the corresponding colour in the table.

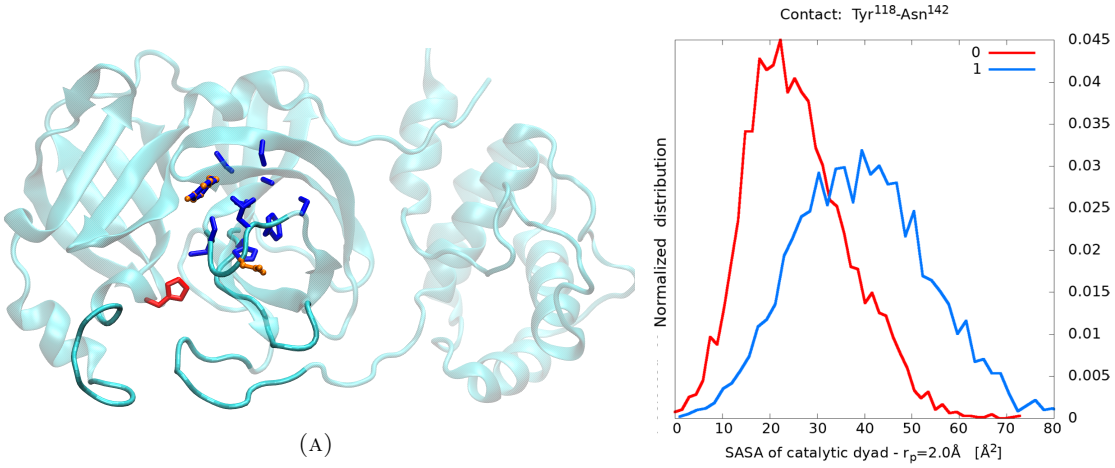
state ID	Ile^{43}-Pro^{52} loop	Phe^{140}-Cys^{145} loop	Phe^{185}-Tyr^{201} loop	2	61	153	154	168
m1:1	αββαααααcβ	βββαββ	ββcββcβββββββββα	β	β	c	α	α
m1:2	αββαααααcβ	ββcαββ	ββcββββββcββββα	β	β	c	α	α
m1:3	ααααacααcβ	cαβααβ	ββcβcββββββcββββα	β	β	β	c	α
m1:4	αααβcβαβc	cβcαββ	ββcβcβcβββββββββα	β	β	β	c	α
m1:5	αααβcβαcβ	cαβααβ	ββcβcβcβββββcββββα	β	β	β	α	α
m1:6	αββαααβαcβ	cαβααβ	ββcββcβββββββββcα	c	β	β	c	α
m1:7	αααβcββcβ	βββαββ	ββcβcβcβββββcββcα	β	β	β	α	α
m1:8	αβαcβcαβββ	βββαββ	ββcββcβββββββββα	β	β	c	α	α
m1:9	αααβcββcβ	ββcαββ	ββcββββββββββα	β	β	β	α	α
m1:10	ααβαβαββcβ	ββcαββ	ββcββββββββββα	β	β	β	α	α
m1:11	αααβcββcβ	cβcαββ	ββcββββcββββα	β	β	β	α	α
m2:1	αββαααααcβ	βββααβ	ββcβcβββββcββββα	β	β	β	c	α
m2:2	αββαααααcβ	βββααβ	ββcβcβββββββββα	β	β	c	α	α
m2:3	αββββααββββ	βββcαβ	ββcβcβββββcββcα	α	α	c	α	α
m2:4	βββββαβαβα	cββααβ	ββcββββcββββββα	α	β	c	α	α
m2:5	αββββαββββα	βββββ	cαβαααααααββββββα	β	β	c	α	β
m2:6	αββαβαcacβ	βββcαβ	ββcβcβββββcββββα	c	β	c	α	α
m2:7	αββββαββββ	βββααβ	ββcββββcββββcββββα	β	β	c	α	α

TABLE 3.1: Selected ψ backbone dihedral angles. The first three column refer to the three most flexible loops, which are the ones controlling the access to the catalytic pocket. The remaining columns refer to other isolated dihedrals, selected due to their high variability throughout the 18 states. For each row, the average over the configurations of the corresponding state is considered. For a better readability, we adopt a ternary labelling: if $-\pi/2 < \psi < pi/6$ the angle is labeled as α; if $\psi < -11/12\pi$ or $\psi > \pi/2$ as β; in all other cases the angle is labeled as c. Dihedrals whose label does not vary in any of the states of a given monomer are reported in light gray colour.

3.4 Candidate Pockets for Allosteric Inhibition

Our analysis shows that the accessibility to the catalytic dyad is reflected in the forming and breaking of few relevant contacts around the reactive cavity. For example, cont. Glu^{47}-Leu^{57} is not formed when the left flap is closed, a condition common to most states in which the catalytic dyad is not accessible. Similarly, the catalytic site (in particular Cys^{145}) is less exposed when the upper flap it tilted downwards, i.e. when cont. Tyr^{118}-Asn^{142} is not formed. The druggability analysis software PockDrug [88] finds one pocket in correspondence of the residues of each of the two contacts (respectively called left pocket and upper pocket) and assigns to them a druggability probability of 0.68 ± 0.08 and 0.95 ± 0.03. Targeting these two regions with drug-like compounds, blocking the formation of the mentioned contacts, might

prove a successful strategy for the inhibition of the catalytic activity. The distribution of SASA over all configuration in which contact Tyr^{118}-Asn^{142} is not formed is significantly shifted towards lower SASA values than in the cases in which the contact is formed (see Figure 3.5b).

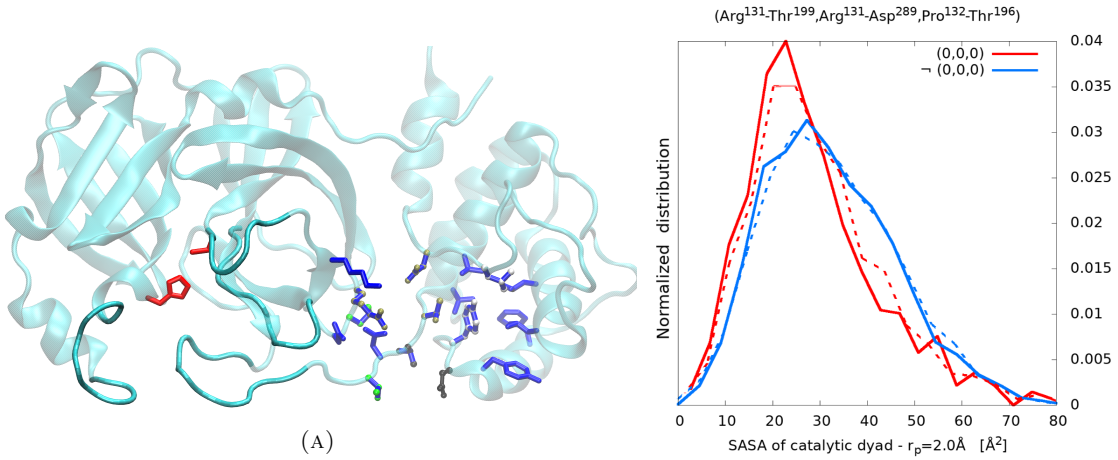


(A) (B)

FIGURE 3.5: (a) VMD [87] representation of monomeric M^{pro} in state m1:9; in red the catalytic dyad; in dark blue the residues involved in the upper pocket found by the software PockDrug [88]. (b) SASA distibutions over configurations in which the contact Tyr^{118}-Asn^{142} is formed or not: 0 indicates that the contact is surely not formed, 1 indicates that the contact is surely formed.

Our analysis on the relevant contacts also unveils the presence of another interesting pocket far from the catalytic site, in the interface region between domains II and III (right hand side of the table in Figure 3.4a). The five relevant contacts in this region are: Arg^{131}-Thr^{199}, Arg^{131}-Asp^{289}, Pro^{132}-Thr^{196}, Asp^{197}&Thr^{198}-Asn^{238}, Tyr^{239}-Leu^{287}. This region, which we call *distal pocket* has been previously identified and screened for docking and has been predicted as a potential druggable target [89, 90]. It has also been suggested as a target for allosteric inhibition of the catalytic activity [91, 92]. Coherently, the predicted druggability score, from the software PockDrug, is 0.65 ± 0.08. With the aim of verifying the presence of allosteric effects involving the distal pocket, we focus on the above mentioned contacts in this region. We compute the distribution of the PDA and of the SASA restricted to the frames in which the contact pattern described above is present or not. Despite all considered residues being far from the binding pocket, the distributions of the PDA and of the SASA are sizably different in the two conditions (see Figure 3.6b). This

suggests that if these five contacts could be forced to be formed or broken according to the desired pattern, e.g. by a drug-like compound, one could influence the PDA and the SASA, controlling indirectly the access to the reactive site. Comparing the table in Figure 3.4a and Figure 3.2, a good candidate for allosteric drugging seems to be the contact pattern of state m1:9: $(0, 0, 0, 1, 1)$. Interestingly, the PDA and SASA distributions obtained by selecting only the first three of the five contacts, namely $(0, 0, 0)$, do not differ significantly from those with all five contacts involved (see e.g. Figure 3.6b).



(A) (B)

FIGURE 3.6: (a) VMD [87] representation of monomeric M^{pro} in state m1:9; in red the catalytic dyad; in dark blue the residues involved in the distal pocket found by the software PockDrug [88]. (b) SASA distributions over configurations with selected contact patterns; 0 indicates a contact surely not formed, 1 indicates a contact surely formed.

Another confirmation of the viability of the distal pocket as a target comes from Xchem crystallographic fragment screening [90]. This study was carried out in the lab of UK's national synchrotron (Diamond Light Source), where, for the first time, the M^{pro} with unliganded active site [90] was crystallized. The XChem crystallographic fragment screening is indeed performed against this specific crystal structure [93]. Among the hits that were identified, three are particularly interesting. Fragment Mpro-x0390, classified as "high confidence", is in contact with atoms from five different residues, among which four are involved in the relevant contacts mentioned above. Fragment Mpro-x0464, also classified as "high confidence", is in contact with eleven residues, among which six are involved in the relevant contacts. Fragment Mpro-x1163, classified as "correct ligand but with weak density", is in contact with

nine residues, among which five are involved in the relevant contacts.

3.5 Conservation of Relevant Residues

We finally analyse the conservation of the residues involved in all the proposed contact patterns in the sequences of proteins belonging to the same family as M^{pro}. Conserved residues are a better target, since the same compound can bind to all the proteins of the family. We thus perform a multiple sequence alignment of our sequence (from PDB 6Y84 [94]) with all the sequences of the Pfam [95] seed of the corresponding family, Coronavirus endopeptidase C30 (Pfam entry PF05409). Similarly to ref. [96], we find that many of the residues involved in the proposed target sites are conserved in all or most of the sequences and furthermore all of them are conserved in the sequence of Human SARS coronavirus (SARS-CoV). These results are presented in table 3.2. The first row contains the amino acid 1-letter code of relevant residues in the Human SARS-CoV2 $3CL^{pro}$ (from PDB 6Y84) , the following ones contains the corresponding residues in the other proteins of the seed of the same Pfam family, obtained via multiple sequence alignment. The sequence IDs reported as column headers refer to the sequence of Human SARS-CoV2 M^{pro}. We see that all relevant contacts are conserved between the M^{pro} of Human SARS-CoV2 and Human SARS-CoV. Particularly stable within the protein sequences appear to be the residues corresponding to: Tyr^{118}, Arg^{131}, Asp^{289}, Leu^{287}. Furthermore, quite recurrent are Asn^{142}, Thr^{196}, Asp^{197}.

Source Organism	**47**	**57**	**118**	**142**	**131**	**132**	**196**	**197**	**198**	**199**	**238**	**239**	**287**	**289**
Human SARS-CoV2	E	L	Y	N	R	P	T	D	T	T	N	Y	L	D
Human SARS-CoV	E	L	Y	N	R	P	T	D	T	T	N	Y	L	D
Murine coronavirus	A	L	Y	C	R	S	Q	D	Y	T	G	F	L	D
Human coronavirus 229E	T	E	Y	N	R	T	A	N	Q	M	G	F	L	D
Feline coronavirus	T	E	Y	A	R	S	T	N	V	M	S	F	L	D
Avian infectious bronchitis virus	S	V	Y	A	R	S	P	D	N	L	G	F	F	D
Thrush coronavirus HKU12	K	I	Y	N	Q	T	T	F	Q	Y	S	F	F	C

TABLE 3.2: Selection of the relevant residues from the MSA of the M^{pro} sequence from PDB 6Y84, with the sequences belonging to the seed of its Pfam family(Coronavirus endopeptidase C30).

3.6 Discussion

In this chapter we presented an application of the approach described in chapter 2 which aims at detecting the metastable states of a biomolecule through the study of the free energy landscape associated to its MD simulation. This approach allowed us to identify 18 putative metastable states of the M^{pro} of SARS-CoV-2. We characterised these states in terms of their structural differences, identifying some contacts which are selectively formed or broken in the different states.

Based on this analysis we propose some possible target sites for the design of drug-like molecules, some of which directly in contact with the flaps regulating the access to the enzyme's active site, some located in the distal pocket at the interface between domains II and III of the monomers. We provide evidence of allosteric effects connected to such pocket and we propose as drug target simply three contacts whose inhibition is correlated to a reduction in the access to the catalytic site; a more refined drug design could yield even stronger catalytic inhibition. We show that all three proposed target sites lie in pockets with high druggability score according to the software PockDrug. A summary of the results of our analysis is shown in fig 3.7, where we highlight the three pockets as detected by the sofware Pockdrug.

We find that all residues involved in the proposed target sites are conserved between the M^{pro} of Human SARS-CoV and Human SARS-CoV-2 and that many of them are conserved in most sequences in the seed of the Pfam family to which they both belong. We interpret this as a comforting indication for the validity of our proposed targets. Moreover, the conservation of all such residues might suggest that mutations are unlikely, thus hopefully the displayed allosteric mechanisms are resistant to possible future mutations. A further possible interesting way to validate the viability of the predicted pockets as potential drug targets, especially of the distal pocket, would be analysing the effect of mutations in that region on the catalytic activity.

We believe that our analysis brings insight on the molecule's conformational changes which might prove useful for the design of farmaceutical inhibitors. Our approach is useful especially for understanding (and eventually controlling) the global dynamics of a protein, since treats the region of the catalytic cavity and any other part of the

protein within the same framework. Moreover, differently from other popular techniques such as MSM, we are able to extract information from a trajectory which is not at convergence. Indeed, we do not expect to have found all the metastable states of the M^{pro} and we have not estimated the transition times between these states. We stress that the same kind of procedure can easily be applied to any other candidate target proteins, due to its generality.

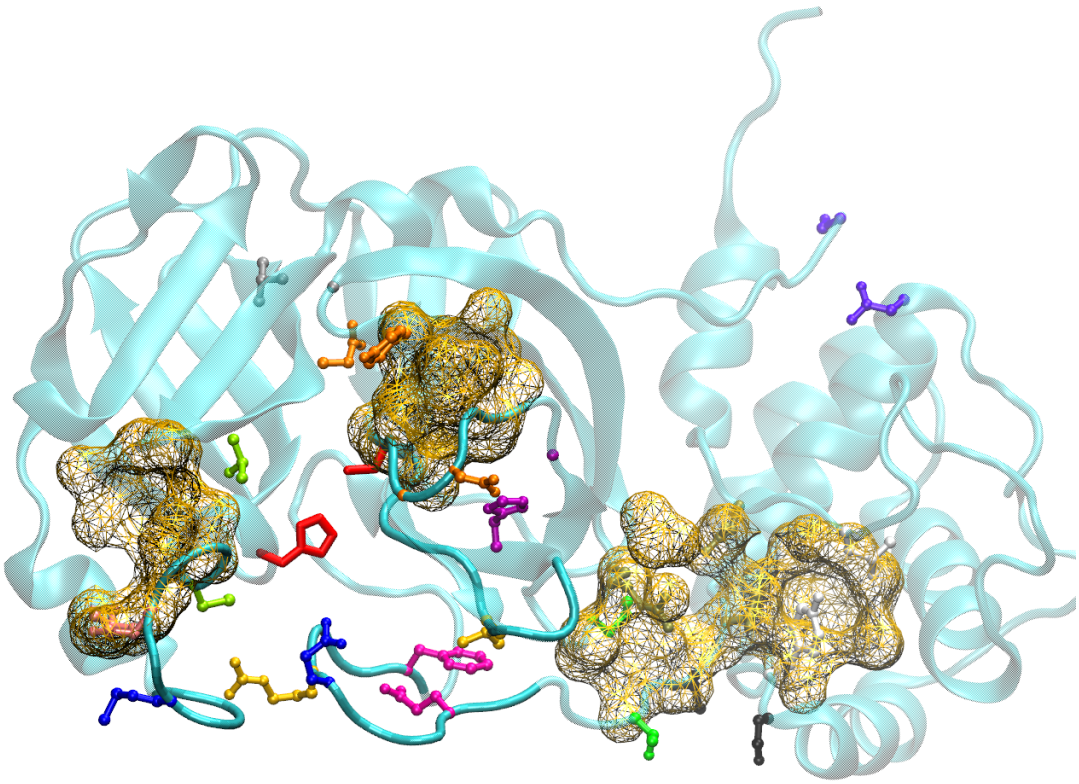


FIGURE 3.7: VMD visualisation of monomer m1. The three pockets as detected by the software Pockdrug are highlighted with the wired representation. The residues in licorice are the ones which we considered in our analysis.

The Folding Free-Energy Landscape of the Villin Protein

In this third chapter of the thesis, we apply the approach described in chapter 2 to characterize the folding free energy landscape of the Villin protein by analysing a $122\mu s$ MD trajectory from Ref. [97]. At the simulated temperature, the protein performs several transitions from the unfolded state to the folded state. We first tried to detect the relevant states of this system using the DP clustering algorithm described in section 2.4.1. However, we realised that none among the detected clusters mapped correctly the unfolded state, which is however a metastable state for this system. The reason of this failure is a posteriori rather easy to understand: in the high-dimensional feature space that we consider in our analysis, the unfolded state is not a free energy minimum, but corresponds to a large region in the landscape where the free energy is approximately flat. To addres this problem, we developed a new clustering algorithm which generalize standard DP clustering. This technique is able to detect also the metastable states stabilized by entropy like the unfolded state of a protein.

Protein folding is possibly the most biologically relevant and studied conformational transition in biomolecules. Proteins organize themselves into a specific three-dimensional structure through an impressively complex conformational change, which can be described as a sequence of elementary reactions [98, 99]. These reactions involve the atoms of the protein as well as the ones of the surrounding solvent. Indeed, the main driving force of folding is thought to be the burial of hydrophobic residues in the core of the protein [100].

The possibly most famous and inspiring paradigm in the field is the folding funnel [99, 101]. At the basis of this paradigm there are empirical observations of simple kinetic patterns shared by many proteins, despite the complexity of folding process. This simplicity is owed to the global organization of the energy landscape of proteins [27, 99, 102], which resembles a funnel: the native state is at its bottom, while the non-native local minima can be thought as small ripples on the walls of the funnel. Going down along the funnel the number of possible states must decrease and the number of native contacts must increase. A schematic representation of the folding funnel is shown in figure 4.1.

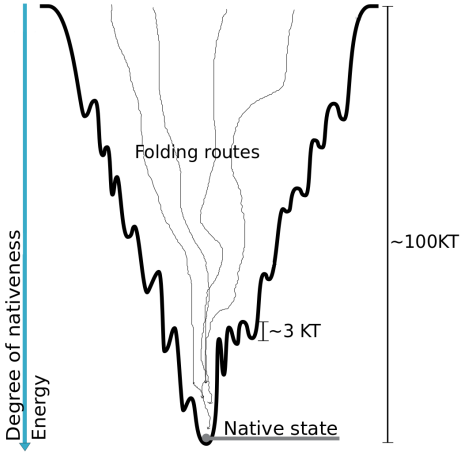


FIGURE 4.1: Schematic representation of the folding funnel from Ref. [99]

The funneled organization of the energy landscape is a result of evolution [99, 102]. Indeed, random heteropolymers are expected to collapse or to form structures resembling a random coil, if by chance there is the formation of stabilizing contacts. If still more stabilizing contacts are present, but without careful placement, the energy landscape will be characterized by a variety of structures having very low energy, but being globally different. This system would exhibit a complex kinetic, with the presence of many traps; the exact ground state would arise by chance from competition between conflicting energy contributions. Such a situation would be unfavourable to genetic evolution of an organism: a single mutation in the sequence would usually cause a structurally different kinetic trap to become the new ground state, causing the loss of the functionality of the protein. On the other hand, evolution has likely

selected sequences according to the principle of "minimal frustration" [102, 103]: the interactions between components are not in conflict between each other, but they cooperate to reach the low energy structure, which is important for its functionality. This principle does not require the elimination of all possible alternative conformations, but their stability should be small enough to be able to escape from them in a time scale much smaller than the one of folding.

These requirements are in agreement with the scenario of the folding funnel with small alternative minima (barriers of $\simeq K_B T$) on the edges. Many folding routes can be followed in order to reach the global minimum of the energy: sidechains may order before or after the mainchain , a specific secondary structure may form before or after another one [104–106]. The dominant routes are the ones in which there is a fast gain of energy paying a low entropy cost [102]. Indeed, a mayor obstacle to folding is the loss of the configurational entropy that characterize the unfolded state. An important feature required by funneling is that all the folding routes have to pass through a common region of the conformational space which acts a sort of bottleneck [107, 108].

In the following, we characterize explicitly the folding free energy landscape of a double mutant of the Villin headpiece (shortly "Villin") [97, 109]. In detail, we analyse the $122\mu s$ of the trajectory from Ref. [97], using the tools presented in chapter 2. We perform the analysis in two different spaces: the space of the backbone Ψ dihedral angles and the space of the backbone atoms, both defined in section 2.1. Since we get analogous results using the two different metrics, we first present a detailed description of the results using the Euclidean distance between the Ψ dihedral angles, summarizing in section 4.6 the ones obtained using the RMSD distance between backbone atoms.

The Intrinsic Dimension of the space of the backbone Ψ dihedrals, estimated by the TWO-NN algorithm described in section 2.2, is approximately 12. We show that the manifold in which the data are embedded is curved and topologically complex, which implies that it is not possible to obtain an explicit expression of these 12 coordinates. However, by using the PAk estimator described in section 2.3.2 one can compute the free energy as an implicit function of these coordinates. Our results are fully consistent with the funnel theory, but interpreting the free energy as an

efficacious conformational energy. Indeed, the free energy of each configuration decreases monotonically with the fraction of native contacts. The depth of the native minimum is $\simeq 15\ k_BT$, and all the barriers on the funnel are of a few k_BT. We then analyze the folding kinetics on the funnel, by the $\hat{k}$-Peaks clustering algorithm described in section 4.4.2. This approach allows locating within the same framework metastable states stabilized by entropy and by energy. We find five relevant states, neatly mapping different regions of the funnel: three with a high fraction of native contacts, and two unfolded. Our model predicts four relevant relaxation times. The slowest is associated with the folding-unfolding transition, the second one, only two times smaller, is associated with an internal relaxation in the unfolded state.

4.1 Intrinsic Dimension

By using the TWO-NN estimator, we estimate the intrinsic dimension(ID) of the manifold on which the data lie. In figure 4.2, we show for each data point i the two quantities $y_i = -log(1 - F(\mu_i))$ vs $x_i = \mu_i$ (see section 2.2), where μ is the ratio between the distance of the second neighbour and the distance of the first neighbour and $F(\mu)$ is the empirical cumlate(cdf) of this quantity. As explained in section 2.2, the best fit of these points in the plane, is a line passing through the origin whose slope is the ID of the manifold on which the trajectory lies ($y = IDx$). Indeed, the red line has a slope of $\simeq 12$, which means that the ID of the system is approximately 12. Qualitatively this means that, on average, from every configuration the system can move in 12 linearly independent directions. A similar value is obtained by using the RMSD metric (see section 4.6).

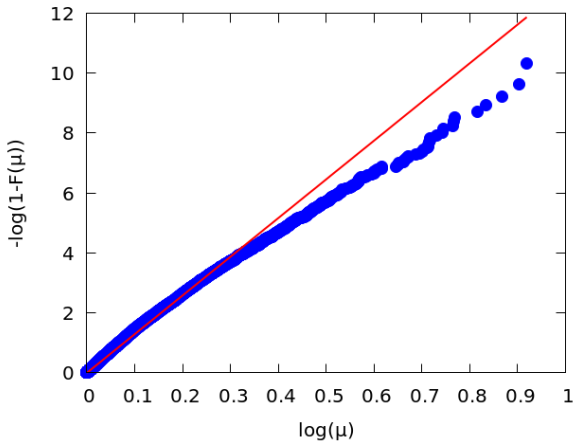


FIGURE 4.2: $-log(1 - F(\mu))$ vs μ for each frame. μ is the ratio between the distance of the second neighbour and the distance of the first neighbour. $F(\mu)$ is the empirical cumlate(cdf) of this quantity. The slope of the red line is $\simeq 12$, this value corresponds to the ID of the system.

4.2 Isomap Projection

In order to investigate the structure of the 12-dim manifold, we first performed an analysis of the trajectory using Isomap, a non linear dimensional reduction method, presented in Ref. [12]. The main idea at the base of this algorithm is to search a low dimensional representation of the data which best preserve the geodesic distances between data points on the original manifold. In practice, a covariance matrix is obtained from the geodesic distances between all pairs of points, the projection is then performed on the directions given by top d eigenvectors of this matrix. A clear gap in the eigenvalues spectrum after the d-th eigenvalue is an indication that the dimensional reduction including the components before the gap is meaningful. Importantly, this approach is consistent only if the manifold containing the data is topologically equivalent to a hyperplane. More details can be found in Ref. [12].

In order to reduce the computational cost of the projection, we undersampled the trajectory by using one every four frames. Then, the Isomap projection has been performed using the Scikit-learn implementation [110], using the five nearest neighbors for defining the geodesic distance. From panel b) of figure 4.3 we see that the projection on the two top eigenvectors, fails to discriminate between folded and unfolded frames. Data points corresponding to different Q values, which are presented with different colors, are overlapping. Therefore a dimensional reduction

to two variables is not meaningful. Moreover, in panel a), we see that it is hard to spot a single gap in the spectrum of the Isomap covariance matrix. Thus, of the reduction of the 12-dim manifold to the space defined by the first twelve eigenvectors is not justified.

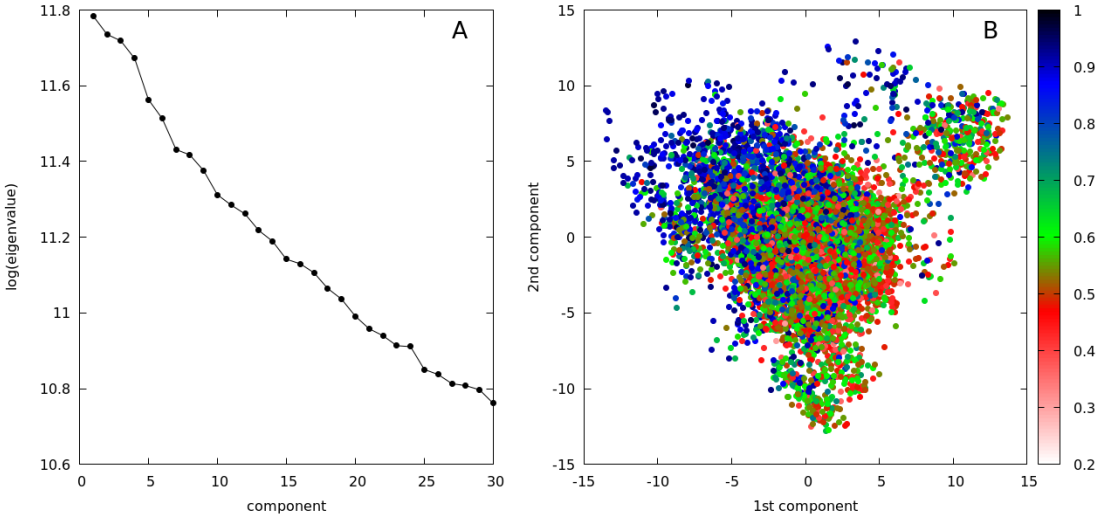


FIGURE 4.3: ISOMAP analysis of the dihedral distances for the Villin trajectory. a) spectrum of the eigenvalues of the projection b) scatter plot of the two first components of the ISOMAP projection colored according to their Q value.

4.3 Description of the Free Energy Landscape

For each point of the dataset (i.e. for each frame of the trajectory), we evaluated the free energy and its uncertainty using the PAk estimator, described in section 2.3.2. We observe a strong anti-correlation between the free energy (F) and the fraction of native contacts (Q): the folded state is the free energy minimum (see figure 4.4, panel a)). The free energy is a monotonic function of Q. The free energy landscape can be thought of as a funnel in twelve dimensions, with a global minimum corresponding to the crystallographic structure and a wide area corresponding to the unfolded region. Moreover, we see in panel b) that there are few states with free energy of $\simeq -17$ (corresponding to the transition region): the funnel has a bottleneck with lower number of available states in the intermediate region.

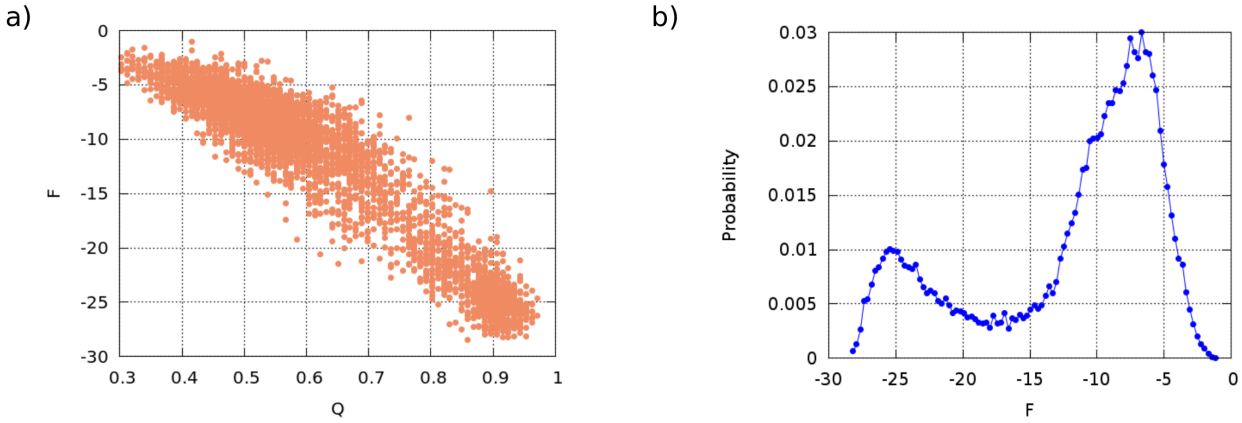


FIGURE 4.4: a) Free energy(F) vs fraction of the fraction of native contacts(Q) for each trajectory frame. Q is evaluated comparing the contact matrix in a structure with the contact matrix of the crystallographic structure(PDB_id 2F4K), two heavy atoms form a contact if their distance is less than 4.5 Å. b) The probability distribution of the free energy. At intermediate free energy ($F \sim -17$, corresponding to $Q \sim 0.75$ there are fewer states than at high and low free energy

4.4 Kinetic Attractors On The Funnel

4.4.1 Density Peaks Clustering

We then attempted analyzing the free energy landscape using the unsupervised version of the DP clustering, described in section 2.4. In this approach, each free energy minimum corresponds to a cluster, and the connections among clusters are obtained measuring the height of the free energy barriers between the minima.

With a merging parameter of $Z = 2.3$ (see equation 2.22), we find 17 clusters. In panels a),b),c) of figure 4.5, we represent the three most populated clusters (denominated CL1, CL2, CL3), which together account for 76% of all trajectory frames. The three plots on the left show the Q probability distribution for these three clusters, the blue arrows point the Q value of the structures which are the centers of the clusters. On the right side of the plots, some representative structures are shown, taken from the peaks of the corresponding probability distribution. For cluster 1 both structures are taken from the major probability peak, for cluster 2 the left structure is taken from the left peak, the right structure from the right one, for cluster 3 both structures are taken from the major peak. We see that:

- Cluster 1 is mainly unfolded, but its center is partially folded (it is at the tail of the Q distribution). The representative structures show a low presence of secondary structure.

- Cluster 2 is a mixed cluster: it contains both folded and unfolded structures. The Q distribution has two peaks: one for $Q \sim 0.5$, the other for $Q \sim 0.8$. Thus, this cluster contain structures that can be really different from each other: for example, the left representative structure, which is mainly unfolded, comes from the first peak. The structure on the right, which is mainly folded comes from the second peak. The cluster center is folded.

- Cluster 3 contains mainly folded configurations, except from a small unfolded tail. The representative structures (one of them is the cluster center) are similar to the crystallographic structure(PDB_id 2F4K).

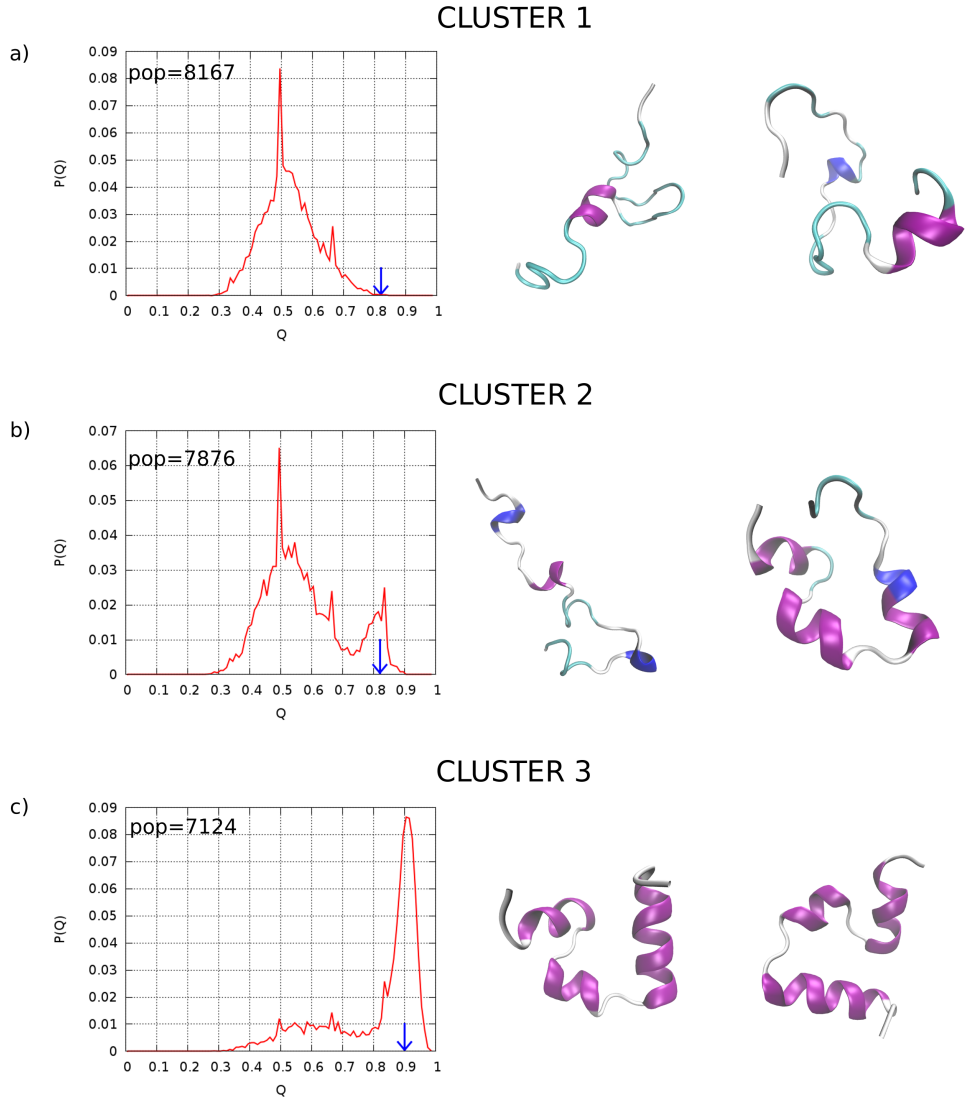


FIGURE 4.5: Panels a),b),c): clusters description, for the three most populated clusters ($CL1$, $CL2$, $CL3$) the probability distribution of the fraction of native contacts($P(Q)$) is shown, the blue arrows indicate the Q value of each cluster center. On the right, some representative structures of each cluster are shown.

We conclude that, this procedure identifies the folded state, but it is hindered by some serious pitfalls. The key problem is that there are no free energy minima corresponding to the unfolded state: the position of the clusters centers is shown with blue dots in the F vs Q plane in figure 4.6. There are no centers with $Q < 0.6$. Indeed, the unfolded state is composed by configurations that are significantly different from each other, with very little or no secondary structure. Clearly, an algorithm attempting to find free energy minima in the space of the C_α positions is not an appropriate tool for studying such a system.

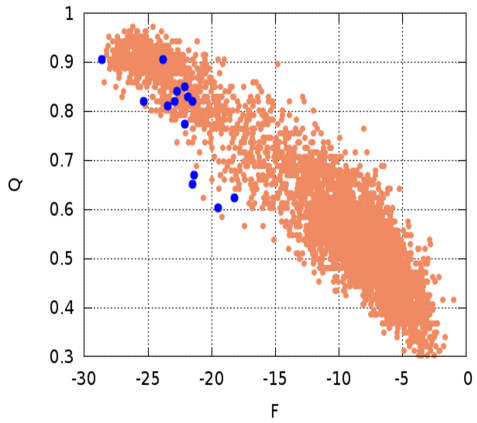


FIGURE 4.6: Free energy(F) vs fraction of native contacts(Q) for each trajectory frame, the blue points represent the 17 clusters centers

4.4.2 $\hat{k}$-Peaks Clustering

In order to address this problem, we developed the procedure called $\hat{k}$-Peaks clustering, described in section 2.4.2. This procedure allows performing clustering on systems in which some of the metastable states are stabilized by conformational disorder.

With the aim of testing the new algorithm, we devise a toy model in which the free energy distribution has a funnel shape (figure 4.7, panel a)). We generated $\simeq$15'000 points in the x,y plane ($x, y \subset [-1, 1]$), from a density distribution given by the sum of a narrow Gaussian centered at the origin and of a uniform distribution. We then applied both the unsupervised version of the Density Peak algorithm and the $\hat{k}$-Peaks algorithm, giving as input the coordinates of these $\simeq$15'000 points. The results are compared in figure 4.7. In panel b) we show, for each point, the dependence of its free energy on an order parameter s, defined as a function of the distance from the origin which is close to 1 if the distance is small, and close to zero if the distance is large. Clearly, the free energy has a single minimum, thus the Density Peak algorithm finds a single cluster. On the other hand, the optimal number of neighbors ($\hat{k}$), as a function of s (panel d)), has two peaks, one at the center of the gaussian($s \sim 1$), the other at the maximum distance from this center($s \sim 0$). Two clusters are thus found by the $\hat{k}$-Peaks algorithm: the points assignations to these two clusters are shown with two different colors in panel d). These results show the ability of the algorithm to localize within the same framework a metastable state stabilized by the

(free) energy, and a metastable state stabilized by conformational disorder, namely a flat area of the (free) energy landscape.

Encouraged by the results obtained with the toy model, we applied the $\hat{k}$-Peaks algorithm to study the Villin trajectory.
We fixed the value of the merging parameter to $Z = 0.2$ (see equation 2.23). If Z is increased, the description becomes less detailed; if Z is increased, it becomes more detailed. We verified that the description does not change significantly if Z is lowered to 0.1 or increased to 0.3, indicating that our results are robust with respect to the choice of this parameter. Even if Z is set to zero the most populated clusters are the same as the ones in $Z = 0.2$, but there are several additional clusters with very small populations, or which are explored only once during the dynamics, and are therefore likely to be numerical artifacts. We choose to present the results fixing $Z = 0.2$ since this value allows a detailed description of the system, maintaining a high level of statistical significance: the relevant states are visited a significant number of times (> 14) and each state has a significant population.
In panel e of figure 4.7, the points of the five biggest clusters are shown in the $\hat{k}$ vs Q plane. These clusters alone contain 93% of all the trajectory frames. In this representation, we see the presence of several peaks of $\hat{k}$ as a function of Q, both for low and high values of Q. The crystallographic state is easily identified in cluster 5, which contains the frames with the highest Q. There are other two peaks with a high value of Q, corresponding to cluster 2 and 4. These two clusters specifically select a region with $0.75 < Q < 0.85$. The unfolded region is mainly represented by clusters 1 and 3. These two clusters almost do not contain any structure with $Q > 0.75$. There is a significant overlap in Q value between the two unfolded clusters, but this is not surprising: Q is a good reaction coordinate for describing the folding process, not the dynamics within the unfolded state. Also the value of Q of cluster 2 and 4 is overlapping with the value in cluster 5: indeed, as we will see, these two clusters correspond to defective folded states, with only a few non-native contacts.

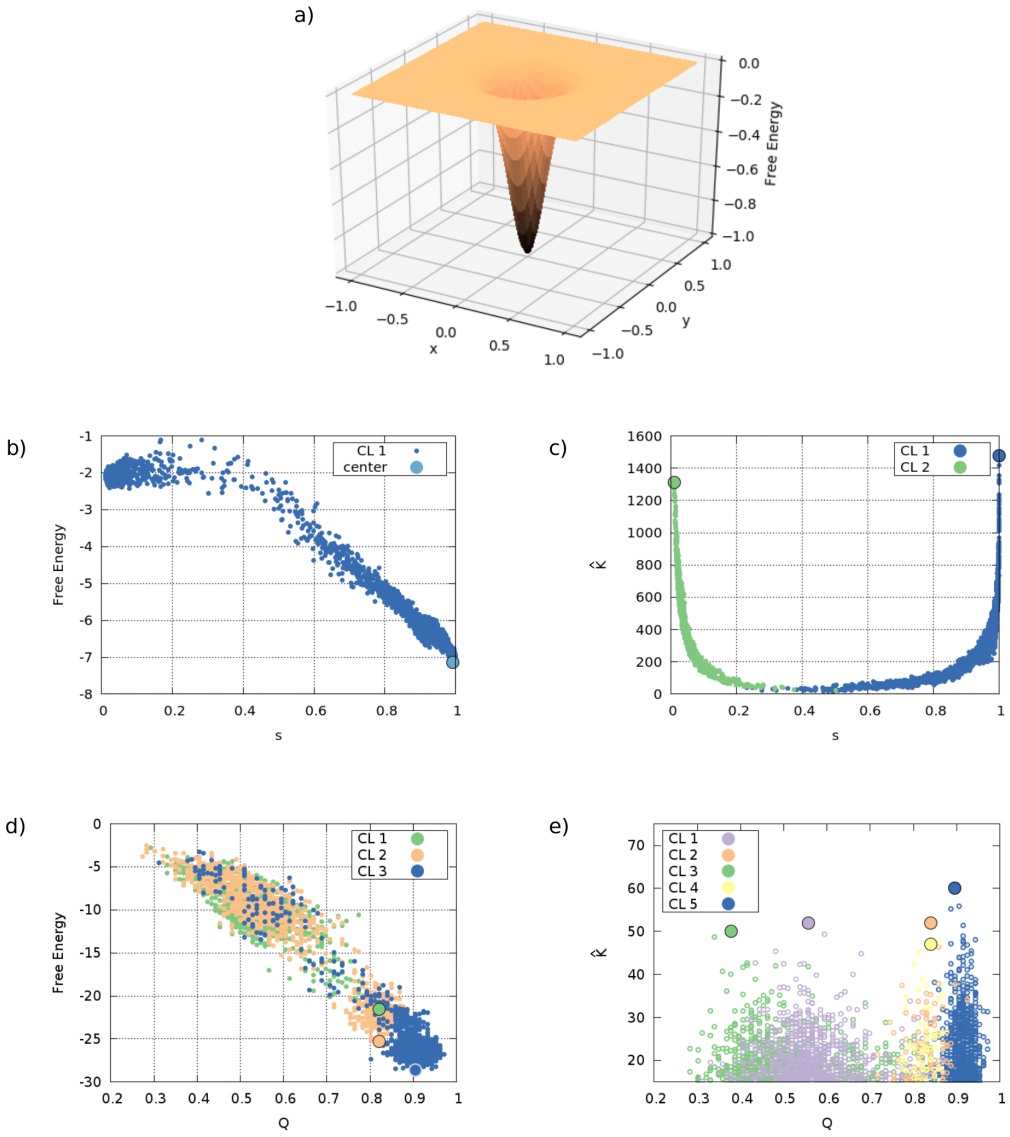


FIGURE 4.7: Comparison between DP clustering and $\hat{k}$-Peaks Clustering. : a) A funnel-shaped two-dimensional free energy distribution; Panel b) and c): the results of the comparison for the toy model in panel a). b) Free energy of each point vs an order parameter $s = \frac{1-(d/0.3)^3}{1-(d/0.3)^6}$, where d is the distance from the origin. The cluster analysis, performed with DP clustering [58] finds only one cluster. c) Optimal value of nearest neighbors of each point($\hat{k}$) vs s. The cluster analysis, performed with $\hat{k}$-Peaks Clustering, finds two clusters. Panel d) and e): the results of the comparison for the Villin trajectory. d) The fraction of native contacts Q vs the free energy F for the frames belonging to the 3 most populated clusters found with DP clustering. e) The fraction of native contacts Q vs the optimal number of neighbors($\hat{k}$) for the 5 most populated clusters found with $\hat{k}$-Peaks Clustering. In panels b),c),d),e) the clusters centers are shown as points with bigger radius.

In figure 4.8 we present the average values of the dihedral angles (Ψ) and their

variance for the core set structures of each cluster. A structure i is assumed to belong the cluster core if its value of $\hat{k}$ is sufficiently high ($\hat{k}_i \geq 25$) and if the following or the previous configuration satisfying the first condition belongs to the same cluster. The first condition selects the frames which are within the lower part of the basin defining the cluster. The second condition discards isolated configurations classified as core states. Once the core set of the clusters are determined, the remaining frames are assigned to the cluster of the previous visited core state. At the end of this procedure, we discard all the clusters that have less than 5 visits. This is done since a minimum number of visits is necessary to have a sufficient statistics in order to describe the dynamics.

The blue thick line in figure 4.8 represents the value of the dihedral angles for the crystallographic structure. Looking at the crystallographic dihedral angles, we see the presence helices when their value is $\Psi \sim -0.8$. This happens in three different regions: From residue 2 to 8, from 13 to 16 and from 21 to 30. The presence of folded clusters$(5, 4, 2)$ and of unfolded clusters$(1, 3)$, already seen in figure 4.7 (panel e)), is confirmed in this representation. The folded region is characterized by structures very similar to each other since the dihedral variance in small. Cluster 5 corresponds to the crystallographic Villin, with the three helices formed. The other two folded clusters (2 and 4) are characterized by structures that are almost totally folded except for the final part of the C-terminal helix. This kind of structure has already been seen as a possible intermediate state between the folded and the unfolded one both experimentally [111], in a computer simulation of triplet triplet energy transfer(TTET) experiments [112] and in a MSM built on the same trajectory [44]. The unfolded clusters are characterized by a high value of the dihedral variance, but their core sets contain structures which are different from each other. Cluster 1, is characterized by structures in which the N-terminal helix ($1 < res < 8$) and the first part of the C-terminal helix ($21 < res < 24$) are formed, whereas the rest of the protein is basically unfolded. Cluster 3 mainly contains totally unfolded frames. As we will see, this distinction has an impact on the relaxation kinetics within the unfolded state.

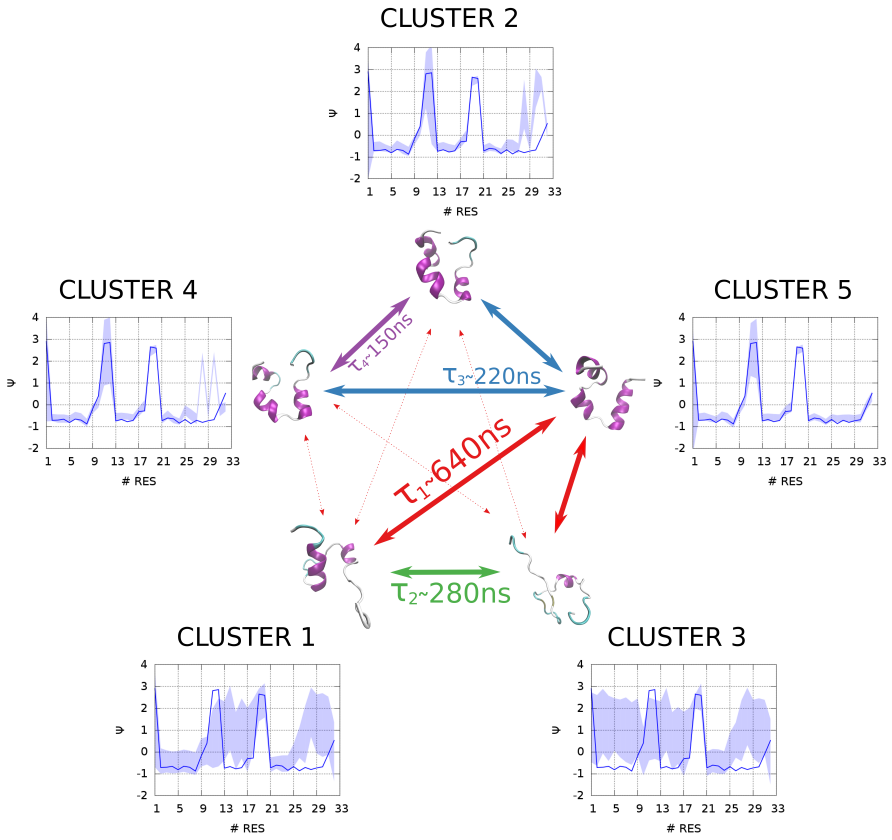


FIGURE 4.8: a) Diagrams representing the dihedral angles values and their variance for the core set structures of each cluster. Next to the diagrams the structures of the centers of the clusters are shown. The arrows link the clusters involved in the relevant transitions, for each transition a color code is assigned, the relaxation times$(\tau_1, \tau_2, \tau_3, \tau_4)$ are written over the arrows.

We also evaluated the heights of the free energy barriers between each couple of clusters. The free energy barrier between cluster A and cluster B is given by $\Delta F_{A-B} = F_{AB} - F_A$, where F_A is the free energy of the center of the cluster A and F_{AB} is the free energy of the saddle point between cluster A and cluster B. In agreement with the experimental picture of a downhill free energy landscape [113], all the barriers from unfolded to folded clusters are really low (around $1KT$). This is not surprising: indeed the folding process in this system is not a rare event due to the presence of a barrier, but rather due to the structure of the free energy landscape, which resembles the one of the toy model in figure 4.7. In this model there are no barriers, and yet finding the (only) free energy minimum is a rare event, since it requires diffusing through a large region where the free energy is approximately flat. On the other hand, the barriers between folded clusters and unfolded ones are large: the highest unfolding barrier, corresponding to the depth of the funnel, is of $\sim 15KT$,

between cluster 3 and cluster 5. Finally, the barriers between the folded state and the two defective folded states (clusters 4 and 5) are of the order of 4 KT.

4.5 Kinetics

Using the $\hat{k}$-Peaks clustering algorithm we have partitioned the entire conformation space into five clusters which, as we will see, allow describing satisfactorily also the dynamics.

In panel a) of figure 4.9, the temporal evolution of these five relevant clusters is shown, for a section of the simulation. Having applied the core set procedure described in section 4.4.2, the spurious transitions have been eliminated, thus the time spent in a cluster before moving to another one (ie the permanence time Δt) is reasonable. In panel b) the same temporal evolution is shown, grouping clusters with similar Q (fraction of native contacts). One group is composed by the clusters 1 and 3, one is composed by clusters 2 and 4, the last one only contains cluster 5. The temporal evolution of these groups (blue line) is compared to the evolution of Q (green line). It is evident the algorithm ability in detecting the transitions of the system between structures with different Q values, not only the transitions between the unfolded structure and the folded ones are seen but also the ones including the intermediate state described above.

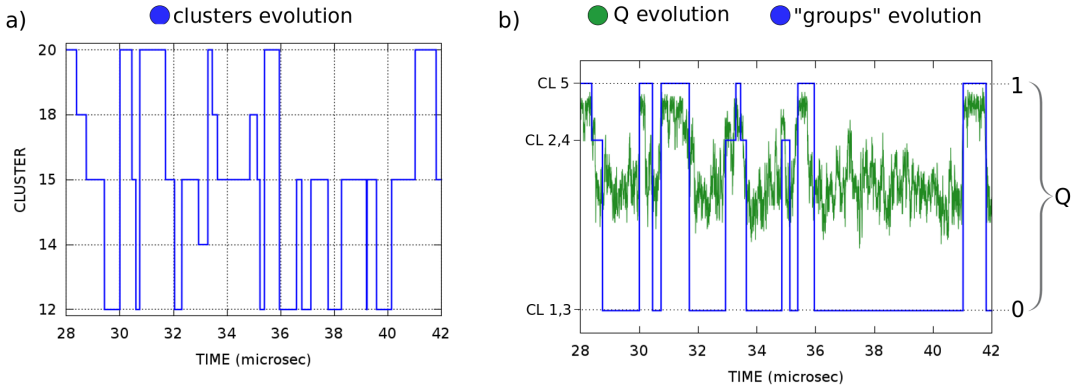


FIGURE 4.9: a)Temporal evolution of the five relevant clusters for a section of the trajectory. b) Temporal evolution, for the same section of the trajectory as a), of three groups of clusters(blue line), compared to the temporal evolution of the fraction of native contacts(Q-green line).

In figure 4.10, we show the negative cumulative distribution of the residence times(Δt) in each of the 5 clusters in semi-logarithmic scale. These curves are well fitted by straight lines. This means that the probability distribution $P(\Delta t)$ is approximately exponential and the process of moving from one cluster to another is a Poisson process.

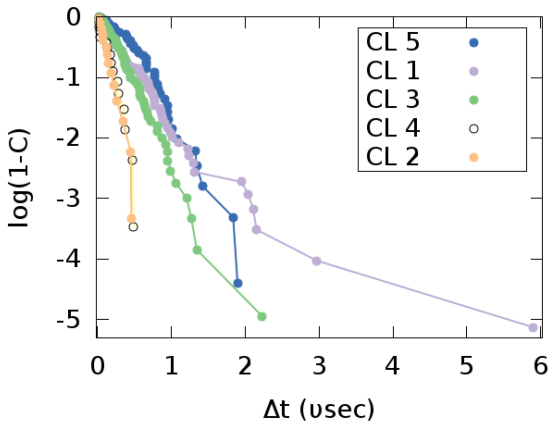


FIGURE 4.10: Logarithm of the negative cumulative distribution (ie $\log(1 - cumulative)$) of the permanence times(Δt) in each of the 5 clusters.

4.5.1 Markov State Model

We then built a Markov State Model (MSM) directly on the five states. The kinetics is assumed to be a memoryless jump process between the five clusters and it is summarized using a 5x5 transition probability matrix ($\hat{\Pi}_{dt}$). $\hat{\Pi}_{dt}$ is an approximation of the Markov operator (Π_{dt}) (see section 2.5). From the spectrum of $\hat{\Pi}_{dt}$ we get the relaxation times of the system τ_i (calculated from the eigenvalues through equation 2.29) and the connections between the clusters (from the eigenvectors) . In panel a) of figure 4.11, we show that there is a wide range of dt for which the relaxation times are almost constant. This proves that our model is approximately Markovian. Specifically, there are four relaxations times related to transitions between different clusters as shown from the eigenvectors in panels b),c),d),e) in figure 4.11 . None of these relaxation times is low enough compared to the others:

- $\tau_1 \sim 640ns$. This value represents the main relaxation time of the system. Indeed the corresponding transition is the general folding/unfolding transition, between clusters $(1, 3)$ and clusters $(5, 4, 2)$.

- $\tau_2 \sim 280ns$. The second largest relaxation time is internal in the unfolded state, between cluster 3 (containing totally unfolded structures) and cluster 1 (containing unfolded structures but with the N-terminal helix formed and C-terminal helix partially formed).

- $\tau_3 \sim 220ns$. The corresponding transition is another internal transition in the folded state, from cluster 5(crystallographic state) and clusters $2, 4$ (containing folded structures but with the C-terminal helix partially unformed).

- $\tau_4 \sim 150ns$. the corresponding transition is between clusters 2 and 4.

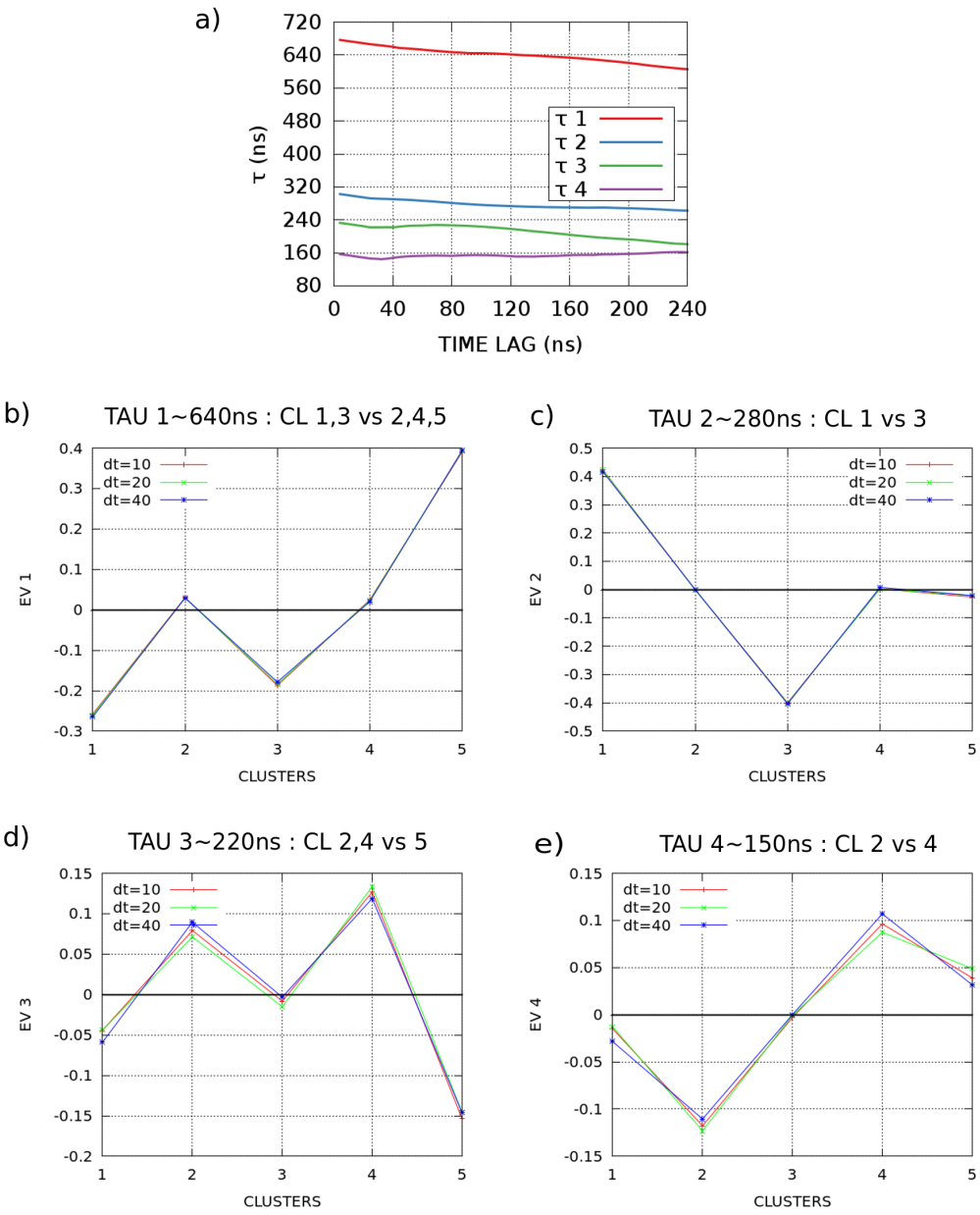


FIGURE 4.11: a)Relaxation times obtained from the transition matrix, as a function of the time lag b),c),d),e)Eigenvectors corresponding to the four relaxation times.

In panel a) of figure 4.8 the arrows represent the transitions. The relaxation times are indicated above the arrows. The longest relaxation time we found ($\tau_1 = 640ns$), is of the same order of magnitude of the one from Ref. [44].

In order to evaluate the folding time and compare it with the analysis from Ref. [97], we gathered the five clusters into two states: the folded one (clusters 2,4,5), and the unfolded one (clusters 1,3). We then evaluated the folding time(t_f) as the average time spent in the unfolded state, and the unfolding time (t_u) as the average

time spent in the folded state. We obtained $t_f = 2.26\mu s$ and $t_u = 0.915\mu s$, in good agreement with the ones obtained in Ref. [97]. This folding time is however bigger than the experimental one, estimated to be $\sim 1\mu s$ [109, 113].

4.5.2 Chapman-Kolmogorov Test

We finally perform an extra markovianity test on the five states model, which was introduced in section 2.5.1.

We compare the transition probabilities between states ($\hat{\Pi}_{ij}$), as a function of the time lag(dt), evaluated in two different ways:

1. directly counting the number of transitions from the trajectory (method 1)
2. scaling the transition matrix evaluated at fixed time lag ($\hat{\Pi}(dt = 120ns) = \hat{\Pi}(120)$, method 2)

Indeed, if the model is markovian, the Chapman-Kolmogorov equation 2.31 should hold. In this specific case we test the equation:

$$\hat{\Pi}(dt) = (\hat{\Pi}(1))^{dt} = \hat{\Pi}(120)^{dt/120} \tag{4.1}$$

The re-scaling of $\hat{\Pi}(120)$ is performed from its eigenvalues and eigenvectors, through equation 2.32. We choose to scale the matrix $\hat{\Pi}(120)$, since in this range the relaxation times of the system are independent of the time lag. In panel a) of figure 4.12, we compare the self transition probabilities for the five clusters, evaluated from method 1 (shown with dots), and from method 2 (shown with lines). For small time lags, there is a perfect correspondence for all clusters and the correspondence holds until $\simeq 200ns$ in the worst case (cluster 4), until $\simeq 300/400ns$ for the other clusters. This shows that the markovianity is respected for a wide range of time lags. The same test has been performed on the three states MSM from Ref. [44], where a similar high quality agreement is observed only at long timescales($> 100ns$).

In panel b) (of figure 4.12), we instead compare the self transition probabilities among different clusters, obtained with method 1) and 2). The correspondence is good for low lag time, it is however lost at a large time lag. This is due to the fact that there are fewer transitions among different clusters than self transitions, the

statistic is poorer, so the counts from the trajectory can deviate from the theoretical curve. Indeed we see that the better correspondence is for transition $1 \to 3$, which have good statistics since it is among clusters that are highly populated and similar to each other (both unfolded). In summary, this test is a strong proof of markovianity and of the precision of our five-state model.

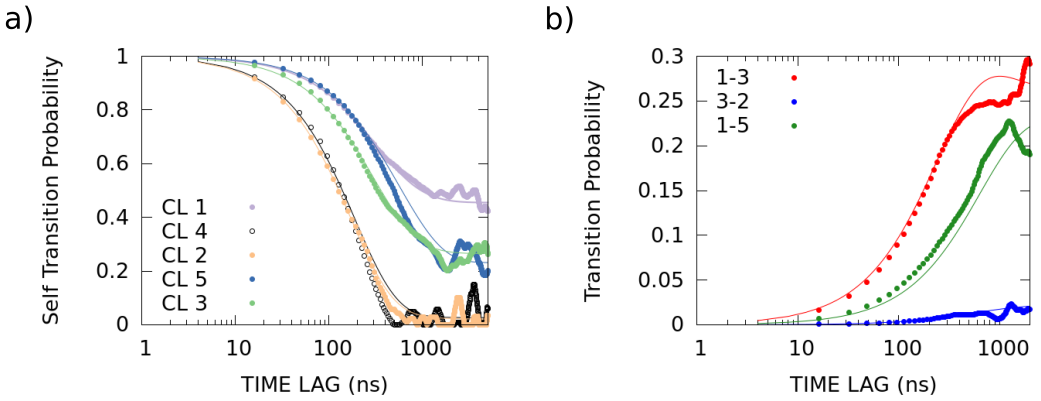


FIGURE 4.12: a)Self transition probabilities as a function of the time lag for the five clusters. b)Transition probabilities between clusters $1 \to 3, 3 \to 2, 2 \to 5, 2 \to 4$ as a function of the time lag. In both panels Dots represent probabilities obtained directly from the trajectory, lines probabilities obtained from the re-scaling of the transition matrix $\hat{P}i$ estimated at a time lag of 120ns.

4.6 RMSD as Distance

As mentioned before, we presented the results of our study using as coordinates the Ψ angles, but we also performed our analysis using as coordinates the X-Y-Z positions of the backbone atoms. The distance between two frames is then calculated as the RMSD distance between the two corresponding configurations. The intrinsic dimension of this dataset is $\simeq 14$. We evaluated the free energy of each frame using the PAK estimator. We then applied the $\hat{k}$-Peaks clustering (following the same procedure used for the dihedral metric). After the merging process (fixing Z=0.2) there are 9 remaining clusters. Applying the core set procedure only 5 clusters survive.

In panel a) of figure 4.13, the positions of the five relevant clusters are shown, in the Q vs $\hat{k}$ plane. In panel b) we present the average values of the dihedral angles (Ψ) and their variance for the core set structures of each cluster (transparency in purple). The blue thick line represents the dihedral angles of the native structure. Importantly there is a one to one correspondence with the clusters obtained using Ψ

coordinates (see panel a of figure 3 of the main text). Indeed, cluster 5 corresponds to the native structure, clusters 2 and 4 correspond to folded structures, but with partial unravelling of C-terminal helix. Cluster 3 and 1 cover the unfolded area: cluster 3 corresponds to totally unfolded structures and cluster 1 to unfolded structures but with the tendency of formation of N-terminal and C-terminal helices.

Let's underline that, analysing the same free energy landscape, but using two different metrics, we have a one-to-one correspondence of the main clusters. This is a strong indication of the robustness of our protocol.

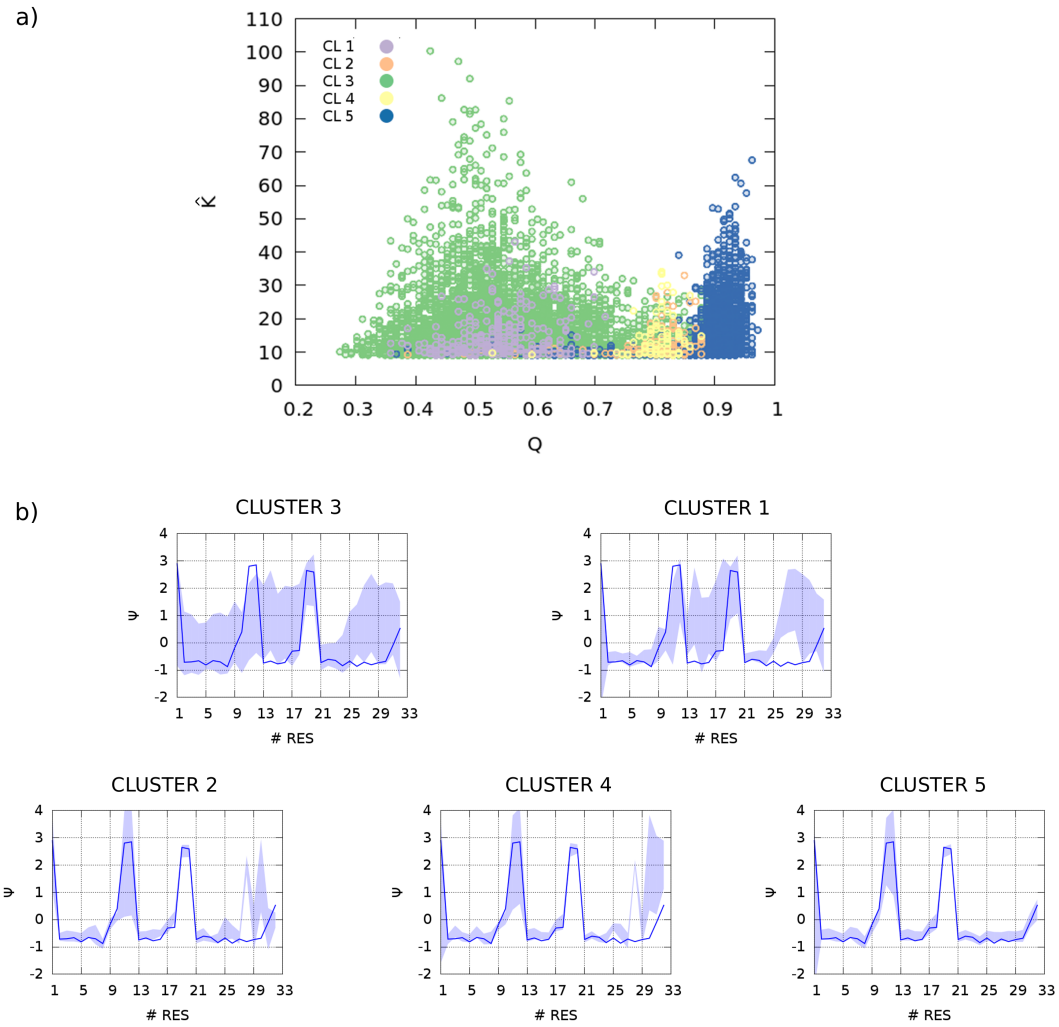


FIGURE 4.13: Panel a) position of the five clusters in the Q vs $\hat{k}$ plane. Panel b) variance of the values of the dihedral angles (Ψ)for the core set structures of each cluster (transparency in purple), the blue thick line represents the dihedral angles of the native structure.

4.7 Analysis of a MD Trajectory Generated with the Amber ff99SD*-ILDN Force Field

We also performed our analysis on a different trajectory of the double mutant of the Villin at 360K, from Ref. [112]. This simulation was also analyzed in Ref. [3, 39]. Also for this trajectory, we selected a frame every $4ns$, for a total length of $\simeq 150\mu s$. The difference between the two analyzed simulations is in the force field: the one in Ref. [97] is the CHARM22* [114] force field, while the one used in Ref. [112] it is Amber ff99SD*-ILDN [115] force field.

We can see from panel a) of figure 4.14, that the funnel structure of the free energy landscape is maintained : there is a strong anticorrelation among the free energy(F) and the fraction of native contacts(Q). The global minimum of the free energy corresponds to the native state. From panel b), however, we see that the percentage of folded frames is much higher than the one in the trajectory from Ref. [97].

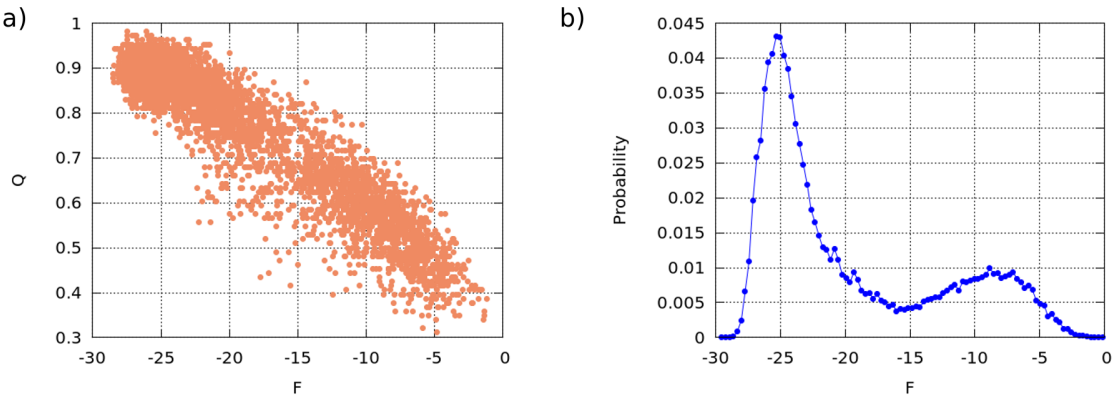


FIGURE 4.14: a) Free energy(F) vs fraction of the fraction of native contacts(Q) for each trajectory frame. b) The probability distribution of the free energy.

We applied the $\hat{k}$-Peaks Clustering procedure, selected the core set and finally evaluated the transition matrix Π, in order to estimate the relevant relaxation times of the system. As shown in panel b) of figure 4.15, with this force field there is a neat separation of time scales: the two first relaxation times are much longer than the others. Five clusters are involved in the transitions corresponding to the two relevant times, shown in the $\hat{k}$ vs Q projection in panel a) of figure 4.15. These five states do not exactly map to the ones obtained analyzing the trajectory from

Ref. [97]. However, also with this force field there is a state corresponding to the native structure, and the unfolded state is split in two main states.

In detail, the two relevant transitions are :

- $\tau 1 = 2400ns$: the corresponding transition is the folding-unfolding transition, between cluster 1,3,5 and clusters 2,4.
- $\tau 2 = 680ns$: the corresponding transition is internal in the unfolded state, between clusters 2 and 4.

Thus, there is correspondence between the two first relevant transitions from trajectory in Ref. [97], and the two relevant transitions from trajectory in Ref. [112]. However, the relevant relaxation times differ by a factor 4.

The comparison of the results of the analysis of the two MD simulations of the same protein, with different force field, underlines the strong influence that the force field has in shaping the free energy landscape. This is already evident from the different percentage of folded frames between the two simulations, which is an indication of the slope of the folding funnel.

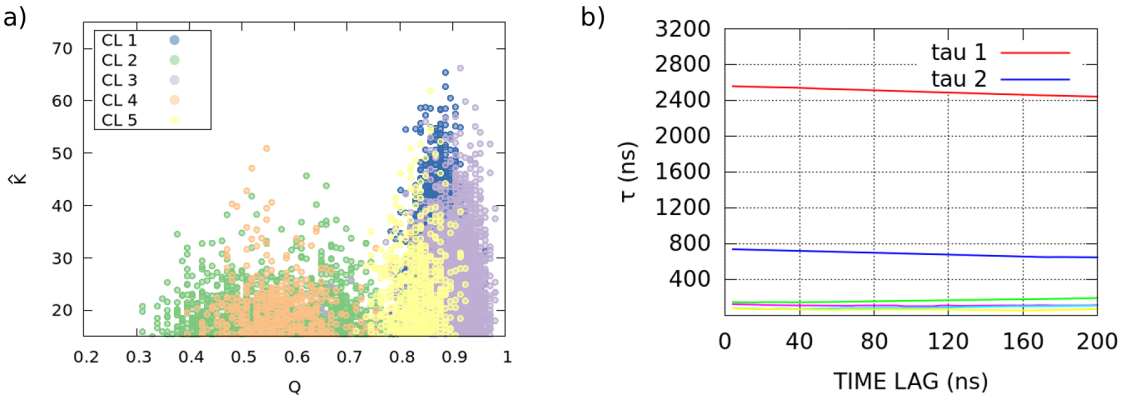


FIGURE 4.15: a) Optimal number of neighbours($\hat{k}$) vs fraction of native contacts(Q), for each frame. Different colors are used for the five clusters. b) Relaxation times of the MSM as a function of the time lag. The first two times(tau1,tau2), are the only relevant ones.

We also compared our 5 states model to the 12 states model from Ref. [3]. The three most populated clusters of their model resemble three of our states, namely the state containing the native structure, a state containing almost folded structures, but with tilted N-terminus and a state containing unfolded structures, but with partially

folded helices 1 and 3. However, we did not find any correspondence for other small clusters they detected as metastable.

4.8 Discussion

In chapter 2, we described a procedure which gives a detailed description of free energy landscapes and of the kinetics on these landscapes, avoiding the definition of any collective variable and the use of information from the dynamics for deriving the model. Our procedure consists of two main steps. The first one is the free energy calculation for each frame of the trajectory using the PAk estimator described in section 2.3.2, the second one is the analysis of the free energy landscape using the $\hat{k}$-Peaks algorithm, described in section 4.4.2 and applied here for the first time. The salient feature of our technique is the capability of identifying both flat regions of the free energy landscape, corresponding to the unfolded states, and minima of the free-energy corresponding to native or near-native states. The main difference with other procedures for building a Markov State Model is that the relevant states are here identified simply by analyzing the structure of the free energy landscape, *without using kinetic information to optimize the partition, or for choosing the number of states.*

Applying our algorithm to the MD-simulation of Villin from Ref. [97], we observe that the free energy landscape as a function of the 32 dihedral coordinates of the protein is actually funnel-shaped. This sheds a new light on the works from Wolynes and Onuchic [99, 101]: our method allows an explicit calculation of an efficacious energy function which describes the folding process. This function is defined as a function of the coordinates of the C_α carbons. We find that, as predicted in the above mentioned works, the free energy is a monotonic function of the fraction of native contacts Q. On the other hand, the number of states is not a monotonic function of Q: the scatter plot of the value of Q versus the value of the free energy F indicates that a bottleneck is present at intermediate values of Q and F ($F \sim -20$ and $Q \sim 0.75$, see figure 4.4). Moreover, our work allows characterizing explicitly the shape of the funnel: even if the feature space is 32-dimensional, the presence of correlations make the manifold on which the funnel lies 12-dimensional. In order to

investigate its structure, we performed an analysis of the trajectory using ISOMAP. Using this technique, a meaningful dimensional reduction on this system couldn't be performed, suggesting that the manifold on which the data are lying is not isomorphic to a hyperplane.

Applying the $\hat{k}$-Peaks clustering algorithm, we obtain five main states. Three of these states are folded: one of them corresponds to the native state, two of them to near-native states in which the C-terminal helix is partially unraveled. The remaining two states are unfolded: one contains totally unfolded conformations, the other contains unfolded conformations, but with the tendency of having parts of the N-terminal and C-terminal helices folded. The permanence times in these two states are long, meaning that these two states are separated by a well defined kinetic barrier. In the trajectory we analyzed we observe 117 direct transitions between the two states, without visiting the native state in between. This implies that the description that we present is not consistent with a kinetic hub scenario [42]. To the best of our knowledge, the existence of two well defined kinetic attractors in the unfolded state of Villin has never been reported before.

In figure 4.16, we present a summary of the five states detected by the $\hat{k}$-Peaks clustering algorithm. We show the positions of the five states in the plane having as y-axis the optimal number of neighbours($\hat{k}$) and on the x-axis the fraction of native contacts(Q). We also present a representative structure for each state (the structure corresponding to the cluster center).

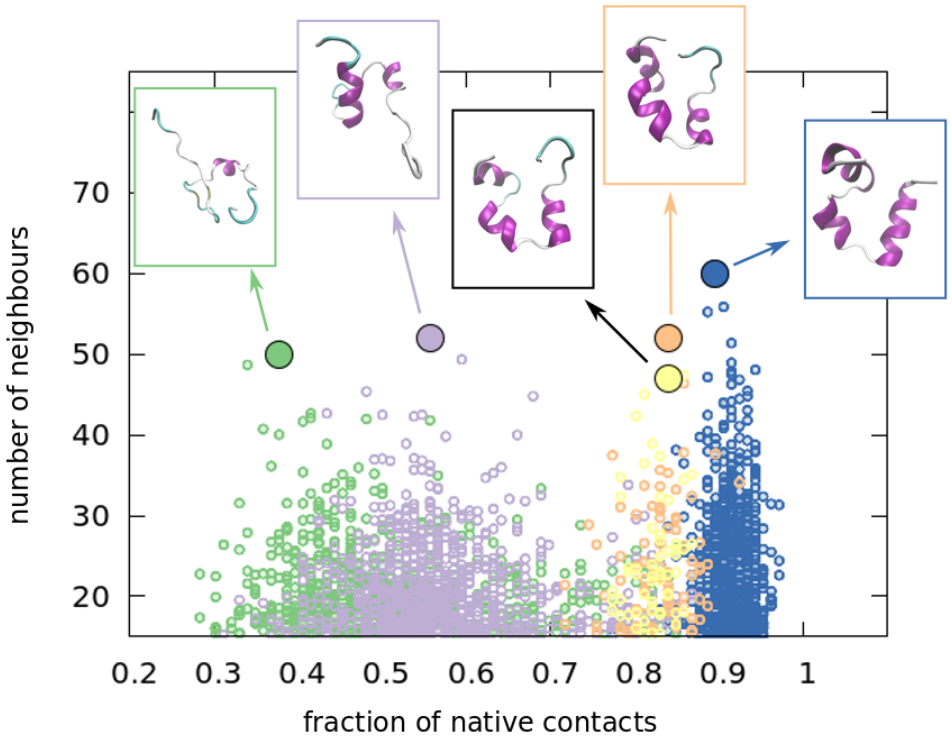


FIGURE 4.16: optimal number of neighbours($\hat{k}$) vs fraction of native contacts(Q) for each data points. Different colors are adopted for the five states. A representative structure is shown for each state.

Studying the kinetics, we predict a folding time which is similar to the one obtained in Ref. [97], but it is however longer than the experimental one (from Refs. [109, 113]). The folding barriers between unfolded states and folded ones are really low ($< KT$), in agreement with the experimental results of a downhill folding landscape. The markovianity of our model is assessed by various tests (see panel a of figure 4.11 and figure 4.12). We remark that in our approach markovianity is not imposed iteratively, but is only verified *a posteriori*.

In order to evaluate the reliability of our results, we applied our procedure to a second trajectory of the same protein, obtained with a different force field (from Ref. [112]). The whole analysis is presented in section 4.7. In this second simulation, performed at the same temperature, the relative time spent by the system in the folded state is $\simeq 70\%$, much more than in the simulation performed with the other force field. Despite this difference, there is an important consistency between the two analysis: in both cases the main relaxation time corresponds to the the folding-unfolding transition and the second one corresponds to a transition internal

in the unfolded state. The presence of two kinetics attractors in the unfolded state is observed with the two different force fields. The same trajectory was analyzed in Ref. [3]. They first performed a dimensional reduction with PCA, followed by a density based clustering and a final step of dynamic clustering using MPP. After this procedure, they obtained 12 metastable states, described according to the secondary structure propensity of each residue. Some of the states they find are similar to ours. A precise comparison on the description of the kinetics is not possible, since the relevant relaxation times of their model and the states involved in the main transitions are not indicated.

In conclusion, thanks to the high quality of the description and to the simplicity of the method, we believe our algorithm will become a popular tool for the studying the structure of (free) energy landscapes, in particular when these landscapes include metastable states stabilized by conformational disorder.

Bionformatic-Aware Rosetta Design

Differently from previous chapters, which focused on free energy landscapes, this chapter concerns a completely different subject: protein design. The "free energy" landscape of protein design is a complex function of a huge number of discrete variables: the identity of the amino-acids at each location. Similarly to previous chapters, the idea at the basis of our contribution is to improve the existing algorithms in order to minimize the need of human intervention.
We attempted to use a popular software of the field (RosettaDesign) to design a protein, and we soon realized that it cannot be used as a black box. Indeed, giving as input the backbone of a natural protein, the output sequences were not similar to the ones of the corresponding family. We thus tried to enhance the performance of the design algorithms using bionformatics: the idea is to drive the design towards the "correct" sequences through the exploitation of the great amount of information contained in a database of natural sequences.

5.1 Introduction

Protein design is the so called inverse folding problem: it aims at identifying sequences compatible with a given protein scaffold. Ultimately, one could use the new designed proteins in order to expand the toolbox available for biomedical and biotechnological application. This makes the development of design techniques a challenge of extraordinary practical importance.

Naturally occurring proteins cover only a tiny fraction of the sequence space. Indeed, if we consider a 100 residues protein and we let the 20 amino acids occupy each position, there would be 20^{100} possible different combinations. On the other hand the order of magnitude of the number of proteins expressed by an extant organism is

$\mathcal{O}(10^6)$ [116] (regardless of their length). Moreover, since more recent natural proteins were created from mutations of the more ancient ones, the sequence space coverage is not uniform, but it is clustered into protein families as depicted in the schematic figure 5.1. There is thus a huge number of sequences that have not yet been sampled by evolution, which can be the field of investigation of protein design.

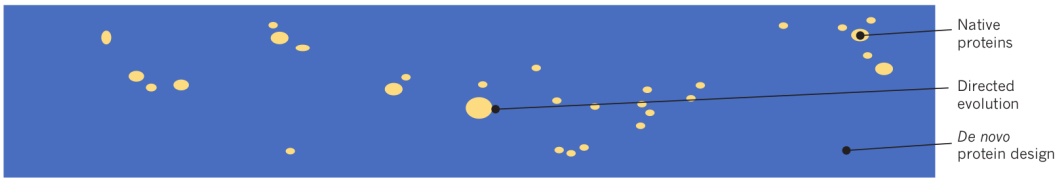


FIGURE 5.1: taken from [116]. Schematic representation of the sequence space. The blue rectangle represents the whole ensemble of sequences obtained with the 20 amino-acids, the beige spots the protein families sampled by evolution.

There are two main categories of design: the first one is the redesign of naturally occurring proteins, the second one is the design of novel protein structures. In the first case, the common aim is to modify existing proteins in order to achieve new functions. The pioneer work in this field was the redesign of a zinc finger domain by Mayo and coworkers [117]. More and more successes followed with designs achieving the stabilization of proteins [118–120], the creation of enzymes with high catalytic efficiency [121, 122], the creation of inhibitors of protein-protein interaction useful to avoid viral infection in animals [123]. In the second case, commonly referred to as de novo protein design, new proteins are generated on the basis of physical and chemical principles, whose sequences and structures are unrelated to those existing in nature. In this case, the protein backbone is chosen from scratch and the challenge is double: it is first of all necessary to build a protein backbone which is physically realizable and then to find a sequence that stabilizes this specific backbone [23, 124]. The construction of good starting backbones is not an easy task since many of the peculiar features of naturally occurring proteins have to be satisfied in order to make the design possible. The great potential advantages in the denovo protein design are the possibility to choose a specific structure for practical application [24, 26] and the possibility to create exceptionally stable proteins since the sequences are not restricted by evolutionary or functional constraints [125, 126].

In all kinds of design, once a model for protein backbone has been chosen (either created from scratch or taken from existing ones), the following step is to find a sequence of amminoacids that stabilizes the desired conformation. The key hypothesis on which the design procedures are based is the Anfisen dogma [127], according to which there is a unique lowest free energy state of a protein (the native state), which is determined only by its sequence. Thus, given

- an accurate method to evaluate the free energy of a amminoacid chain in a given structure, i.e a scoring function, capable of estimating approximately the free energy
- an efficient method to sample sequence and to sample the conformational space restricted to conformations which are consistent with respect to the target tertiary structure

it should be feasible to design sequences that have a given structure as their native state. The main difficulties in the design process are the huge dimension of the sequence space to be searched and the impossibility to perfectly reproduce in the scoring function all the chemical and physical interactions between the atoms.

Rosetta Design is among the most popular computational design software packages. Its development for protein design stemmed from the framework utilized for proteins structure prediction. At the basis of the design protocol there are two main components: a sampling algorithm, used to select candidate sequences from the huge sequences space [128] and the Rosetta Energy Function [129], used to evaluate the viability of a sequence according to the interactions among the amino-acids. An important feature of Rosetta Design is the possibility to couple sequence and conformational sampling, allowing the protein backbone to iteratively adapt to the accumulated mutations, while keeping the tertiary structure approximately fixed. Using Rosetta Design, many groups obtained remarkable successes [23–26]. In particular, Kuhlman *et al* were the first to design a novel globular protein, with a topology which has not been observed in nature [23]. Another example of a recent achievement is in the field of vaccine design [24]. In this work, the transplantation of a viral epitope from the respiratory syncytial virus onto a de novo designed scaffold is performed,

allowing full backbone flexibility. This designed immunogen induced the production of neutralizing antibodies in vivo.

These and many other impressive results were obtained exploiting Rosetta Design, but many of them included the experimental testing and optimization of large sets of candidate sequences [24,26,130,131]. Often the design process also included extensive human curation of the designed sequences by expert biochemists. This, for practical purposes, is a fully legitimate and valid protocol. But a question that remains open is *if it is possible to robustly design protein sequences with a fully automated protocol*, which exploits only an optimization algorithm, with no human curation of the results. This would make protein design as unsupervised as, say molecular dynamics. Here, we addressed this question by designing two small protein domains, which have already been widely studied in other computational design work [128, 132, 133]: one belonging to the SH3-1 family, the other to the Ubiquitin family. To do so, we use the representative backbone structures of the protein folds (from the PDB [134, 135]) and design sequences that are the most energetically favourable according to Rosetta Energy Function, while allowing backbone flexibility. To quantify the quality of the designed sequences, we used a different metric than those typically used in Rosetta Design benchmarks [132, 133]. We evaluate each designed sequence by estimating its probability to be evolutionary related with a natural protein belonging to the ground-truth family using BLAST [136] or, more directly, we estimate its probability to belong to the ground-truth family using Hmmer [137]. Both software evaluate the statistical match between a sequence and a sequence database (the sequences belonging to a protein family in our case), but BLAST is based on pairwise sequence alignment, whereas Hmmer uses the Hidden Markov Model(HMM) profile of a protein family.

We find that, if backbone flexibility is allowed, the re-designed sequences are not identified as being part of the original protein families. This finding is not necessarily at odds with the quality of Rosetta Design: Rosetta sequences are selected only to optimize the stability, while natural sequences are also selected to optimize function and other cellular requirements. In other words, the space of sequences which are compatible with a given structure can be in principle very large, much larger than the space spanned by natural sequences. However, we also find that Rosetta Design selects

a relatively small subset in sequence space: indeed, independent design simulations started from a poly-valine and from the natural sequence end up in a final set of design sequences which are, according to BLAST and Hmmer, the same protein family. Essentially, Rosetta does not seem to care about the initial sequence which is rejected as if it was not appropriate to stabilize the structure, but only about the initial backbone configuration.

This result on one hand demonstrates the remarkable robustness of Rosetta Design protocol, which is able to find bioinformatically consistent sequences (for a given backbone structure) following totally independent optimization pathways, but on the other hand poses a problem, since the solutions it often finds do not contain the features that are identifiable with other folds in nature. Indeed, we will see in section 5.5 that the proteins designed with this protocol do not fold to the correct structures. In order to address this problem, here we propose a Genetic Algorithm(GA) in which the design steps are combined with a progressive optimization of the agreement of the sequence with a database of natural sequences. Starting from a "parent" sequence, a number of sequences are generated through Rosetta Design, allowing a fixed number of mutations. Among these "progeny" sequences, we select those which are more compliant with the features of the sequences observed in nature. This compliance is quantified by scoring with BLAST each sequence of the progeny against the database of all the sequences in the pdb. This procedure is then iterated for a predetermined number of steps. To test our protocol we characterized experimentally several of the designed sequences and obtained folded proteins with the expected secondary structure signatures for one of the folds.

5.2 Rosetta Design

We here present the design techniques adopted in the software Rosetta Design, which is the one we used in our investigation.

There are many protocols implemented in Rosetta Design, however they are all based on the same optimization algorithm [128], used to search the sequence space, and on the same scoring function (the Rosetta Energy Function [129]) used to evaluate the different conformations. An important feature of Rosetta Design, is that

it allows the possibility of coupling of sequence and conformational sampling: often sequence optimization steps are alternated with steps of structure relaxation, in which restricted movements of the backbone are allowed. Examples of algorithms used to relax the backbone structure are in the references [138–141]. This possibility is especially relevant in the case of de novo protein design, in which the existence of a compatible sequence for a target structure is not warranted, and a certain degree of structural relaxation allows a broader sequence search and helps finding better solutions.

The differences among the protocols of Rosetta Design are in the techniques used to allow movements of the structure or in the way of alternating cycles of sequence design and structure relaxation. Design simulations are typically performed independently and in large numbers, in order to obtain many candidate sequences (typically thousands), which are then further selected according to structural and sequence quality metrics.

In the first two following subsections we describe the optimization algorithm and the Rosetta Energy Function, in the third one we present the protocol of Rosetta FastDesign [138].

5.2.1 Optimization Algorithm

The search in the sequence space is performed through a Monte Carlo optimization with simulated annealing. Each move consists of an exchange of an amminoacid in random position with another one. The conformation of the side chain is harvested at random from a rotamer library (the Dunbrack library [142]). The rotamers are around 150 for all aminoacids, these correspond to the conformations mostly occupied by residue sidechains and thus commonly observed in high-resolution structures from pdb. The described procedure is technically defined as packing of the side-chains . The new proposed sequence is accepted or not according to the Metropolis criterium, using as potential the all-atom Rosetta Energy Function.

5.2.2 Rosetta Energy Function

We used in our protocol the Rosetta all-atom Energy Function from 2017, a detailed description of this potential can be found in the dedicated paper [129].

In the Rosetta all-atom energy function, the energy of a conformation(E_{tot}) is given by the weighted linear combination of energy terms E_i which are function of the chemical type of atom (denoted aa), and of geometric degrees of freedom (denoted θ):

$$E_{tot} = \sum_i w_i E_i(\theta_i aa_i) \tag{5.1}$$

where the sum is over all the different energy terms and w_i is the weight of i-th term. There are two main categories of energy terms: the first is related to the interactions between atom pairs and the second is related to the torsion angles of backbone and side-chains. In the following paragraphs we will present the main terms belonging to these two categories.

Interactions between Atom Pairs

Typically in Rosetta, in the case of bonded interactions, bond lengths and angles are kept fixed and conformational space is sampled changing only torsions. The following terms of the total energy thus evaluate bond-length and bond-angle energetics in the case of non-bonded interactions:

- Van Der Waals interactions, described through the Lennard Jones potential:

$$E_{VDW} = 4\epsilon[(\sigma/r)^{12} - (\sigma/r)^6] \tag{5.2}$$

  where r is the distance between the couple of atoms, and the parameters ϵ (depth of the well) and σ (distance at which the potential is zero) are dependent on the specific couple of atoms.

- Electrostatic interactions between couples of atoms, with charges q_i and q_j, described through the Coulomb's law:

$$E_{coulomb}(ij) = \frac{q_i q_j}{\epsilon} \frac{1}{d_{i,j}^2} \tag{5.3}$$

  where $d_{i,j}$ is the distance between the atoms, and ϵ is the dieletric constant. The charges are taken from the CHARMM force field.

- Solvation. Since a potential with explicit solvent is computationally too expensive, Rosetta describe the interaction between the proteins atom and the solvent using the Lazaridis-Karplus implicit model [143]. This model has two components: the "isotropic solvation energy" to account for bulk water, which is uniformly distributed around the atoms; and the "anisotropic solvation energy" to account for specific water molecules nearby polar atoms.

- Hydrogen bonds, which form when a nucleophilic heavy atom donates electron density to a polar hydrogen. These bonds are difficult to take into account since they have both covalent and electrostatic contributions, moreover they require a precise geometry of the atom positions. Rosetta calculates the energy of hydrogen bonds using the electrostatic term (previously described) and a second term that evaluates energies based on statistics of the geometries of H-bonds present in a database containing high-resolution crystal structures [144] .

Terms for Protein Backbone and Side Chain Torsions

Rosetta performs the search in the conformational space varying the torsional angles, both of the backbone and of the side chains. The relative favourability of different "torsional conformations" is then established on the basis of statistical potentials which are a good strategy to reproduce the most common solutions adopted in nature. We here summarize these terms, without specifying the related equations and protocols since they are very technical. More details can be found in [129].

- Ramachandran term. Rosetta Energy Function include a term to evaluate the backbone ϕ and ψ angles which is based on the Ramachandran map of each amino acid. These maps are obtained using torsions from a high number of selected protein chains. The probabilities are then converted to energies via the Boltzman inversion [145].

- Backbone design term. This term takes into account the likelihood of placing a specific amino acid side chain given an existing ϕ, ψ backbone conformation.

- Side chain conformation term. As we previously explained, Rosetta performs its search only in the space of the rotamers belonging to the Dunbrack's library.

This library also supplies the values of the relative probability of each rotamer, which is then converted in an energy value.

The balance of different terms in equation 5.1 is a crucial step since some of the contributions may be overlapping in describing specific interactions. The followed strategy is to fix to one the weights of physics-based terms and to determine the weights of the terms based on statistics, by optimizing the agreement of Rosetta calculations with the thermodynamic data of small molecules and features of natural structures [146].

5.2.3 Rosetta FastDesign

FastDesign protocol is comprised of interlaced cycles of repacking of the side chains (using rotamers belonging to all amminoacids) and gradient-based minimization of the backbone and side chain degrees of freedom. In the two steps the same Rosetta energy function is used. The key feature of the FastDesign algorithm is that during the backbone relaxation the weight of the repulsive part of the VanDer Waals interactions is alternately increased and decreased. This softening of the repulsive forces is indeed able to enhance sampling in protein folding calculations.

5.3 Scoring a Sequence

In this section, we present two instruments which we used to evaluate the "quality" of our designed sequences: BLAST and Hmmer.

The Anfisen dogma from 1960 [127] created a bridge between the field of biological sequence analysis and protein structure prediction. Indeed, this dogma states that in physiological conditions the most stable conformation of a protein (i.e the native state) depends only on its sequence. A possible fast and cheap strategy to determine the native conformation of a protein, is thus to detect a significant similarity between its sequence and another protein of known structure. One of the main goal of bioinformatics is thus the development of techniques able to detect homologs, which are sequences related by evolution and thus having some similarities.

If our only knowledge about a protein is its the sequence of amminoacids, then deciding that two sequences are similar is the same as deciding if two text strings are similar. Here the situation is a bit more complicated. Indeed, substitutions among amino-acids are more or less probable according to the residue type. Moreover, along the evolution also insertions and deletions are accumulated. The concept of alignment becomes thus essential in this framework: in order to establish the sequence similarity, it is first necessary to determine which positions should be paired. In this process, gaps in each sequence are generally allowed, provided that a penalty is paid in the score. A simple example of alignment between the two sequences *HEAGAWGHEE* and *PAWHEAE* looks like:

HEAGAWGHE- E

- - P- AW- HEAE

Here some residues are conserved (for instance A in the fifth position), some are deleted, other are inserted (for instance H in the first position).
The tools to perform sequence alignment are scoring schemes and alignment algorithms. The former associates each alignment with a score according to the sequences similarity; the latter, given the sequences and a scoring scheme, provides their best alignment. In the following sections we will review these two topics, focusing on the algorithms which we used in our analysis.

5.3.1 Scoring Schemes

Scoring schemes associate to each alignment a score according to the probability that the aligned sequences derive from a common ancestor. The simplest scheme we can imagine is '+1' for a match, '-1' for mismatch, however more complex solutions are usually adopted trying to exploit biological information. Indeed, mutations that radically change the chemical properties of a residue are rare since they can affect the protein structure and consequently its functionality. In order to quantify the evolutionary preferences for certain substitutions with respect to others, probabilistic matrices containing all possible pair-wise aminoacid scores were introduced. These 20x20 matrices are also defined scoring matrices and provide, for each pair of residues (i,j), a score s(i,j). s(i,j) is higher or lower according to the probability that i and j

are aligned because of a common ancestor rather than by chance:

$$s(i,j) = log(p_{ij}/q_i q_j) \tag{5.4}$$

where p_{ij} is the joint probability of having the residue i and j aligned, and q_i and q_j are the frequencies of residue i and j. The score of an alignment is given by:

$$S = \sum_i s(x_i, y_i) \tag{5.5}$$

where x_i and y_i are the residues at i-th position of the two aligned sequences .

The most used scoring matrices are PAM matrices [147] and BLOSUM matrices [148].

Aligning algorithms don't consider only the probabilities of substitutions, they also allow the insertion of gaps. The simplest way to take this possibility into account is to associate gaps to a penalty score; such penalty can be linearly increased with the number of consecutive gaps or be different for the opening of a gap and than for its extension. In this kind of approach insertions and deletions lose their evolutionary meaning, they are treated as a special kind of mismatch. However, observing alignments of many sequences coming from a coomon ancestor it is evident that gaps tend to line up with each other, leaving blocks where no insertions or deletions are present. More precise approaches have thus been developed in which gap-penalties are not uniform along the alignment. This is the case of Profile Hidden Markov Model [149]. In general, sequence analysis based on HMM is characterized by a stronger theoretical basis than the one of other scoring schemes. This topic will be discussed in section 5.3.3.

5.3.2 Algorithms for Sequence Alignment

Algorithms that perform sequence alignment are classified into different categories according to their specific aim. Indeed, according to the number of sequences to be treated, there is the distinction between pairwise sequence alignment and multiple sequence alignment(MSA) algorithms. Another distinction is between global algorithms that require every residue in every sequence to be aligned with something

(gap or residue) or local algorithms, where only part of the sequences may be present in the alignment. Note that local alignment is usually the most sensitive way to detect similarity in case of highly diverged sequences.

Pairwise Sequence Alignment

The algorithms for finding the optimal alignment are based on dynamic programming; which, given an additive alignment score, guarantees to find the optimal solution or a set of optimal solutions. In this method, complex problems are decomposed into a list of simpler sub-problems. Each sub-problem is then solved just once and its solution is stored so that, next time the same problem occurs again, its solutions can be retrieved without recomputing it. The most widespread algorithms in this field are the Needleman-Wunsch [150] for global alignments and the Smith-Waterman [151] for local ones. We briefly present the global version (following the explanation from Ref. [152]), being the local one its simple extension. The idea is to find the optimal alignment of the whole sequences using the previously calculated optimal alignment of smaller subsequences. In particular, the process is based on the calculation of a matrix in which columns correspond to letters of the first sequence and rows to letters of the second sequence. A matrix element $F(i, j)$ is the score of the best alignment between the initial segment of the first sequence $x_{1\ldots i}$ and the initial segment of the second sequence $y_{1\ldots j}$. $F(i, j)$ is built recursively from $F(0, 0) = 0$, moving to the right down corner. At each step $F(i, j)$ takes the maximum value between the three options:

1. $F(i-1, j-1) + s(x_i, y_i)$
2. $F(i-1, j) - d$
3. $F(i, j-1) - d$

where in the first case $F(i, j)$ comes from the diagonal, and corresponds to an amino-acids alignment with score $s(x_i, y_i)$, in the second case and third case $F(i, j)$ comes from a gap insertion and corresponds to the penalty d. This equation is repeated to fill in the matrix as in the following figure:

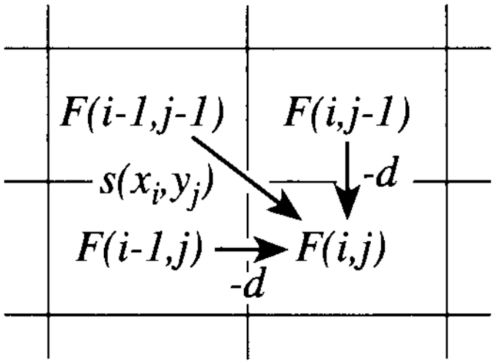


FIGURE 5.2: taken from [152]. The value $F(i,j)$ is calculated from one of the three top-left neighboring cells.

To complete the algorithm, we just miss the boundary conditions, which means the values of the first row and of the first column. Since $F(i,0)$ represent the alignment of the segment $x_{1..i}$ to all gaps, it takes the value $F(i,0) = -id$; for the same reasoning $F(j,0) = -jd$.

In figure 5.3, we show an example of the whole alignment procedure for two sequences, using the scores from BLOSUM50 matrix. Every time we evaluate $F(i,j)$, we need to keep a pointer to the cell from which its value was derived. In this way we store the path of choice to get the best score and we are thus able to get the alignment from the matrix with the so called "traceback" procedure.

		H	E	A	G	A	W	G	H	E	E
	0 ←	**-8** ←	**-16** ←	-24 ←	-32 ←	-40 ←	-48 ←	-56 ←	-64 ←	-72 ←	-80
P	-8	-2	-9	**-17** ←	**-25**	-33 ←	-42 ←	-49 ←	-57	-65	-73
A	-16	-10	-3	-4 ←	-12	**-20** ←	-28 ←	-36 ←	-44 ←	-52 ←	-60
W	-24	-18	-11	-6	-7	-15	**-5** ←	**-13** ←	-21 ←	-29 ←	-37
H	-32	-14	-18	-13	-8	-9	-13	-7	**-3** ←	-11 ←	-19
E	-40	-22	-8 ←	-16	-16	-9	-12	-15	-7	**3**	-5
A	-48	-30	-16	-3 ←	-11	-11	-12	-12	-15	**-5**	2
E	-56	-38	-24	-11	-6	-12	-14	-15	-12	-9	**1**

FIGURE 5.3: taken from [152]. Matrix for the global alignment of two example sequences. Values on the optimal alignment path are shown in bold, the arrows indicate the traceback pointers.

In the shown example the best alignment is thus:

HEAGAWGHE- E

- - P- AW- HEAE

The Smith-Waterman algorithm (for local pairwise sequence alignment) follows a slightly different procedure with respect to the one previously presented. Indeed, every time we calculate $F(i, j)$ there is the extra possibility to take the value 0 if all the top left neighbors display negative values: this corresponds to starting a new alignment. Moreover, alignments can end wherever across the matrix. To obtain the final alignment one needs to find the highest score in the matrix and reconstruct the trace back from there, the traceback procedure ends when meeting a cell with value 0.

BLAST Deterministic algorithms are guaranteed to find the optimal solution, however they are time consuming, and with the increasing of the number of sequences to be analysed, speed becomes an issue. For this reason, a lot of efforts have been employed looking for faster heuristic techniques. The most famous heuristic algorithm for pairwise alignment is BLAST [136], which we will use in our investigation.
BLAST is based on the idea that true match alignments have a high probability of containing short segments of identities, or very high scoring matches. First of all the algorithm looks for these short segment (called "seeds"), and then tries to extend the alignment looking for higher scores. Thanks to its speed, BLAST is currently used to perform alignment of sequences against sequence databases, for example a database containing all the sequences belonging to a specific protein family. Importantly, BLAST gives as output, along with each alignments with its score, statistical information on the significance of the match, based on the Karlin-Altschul theory [153]. The measure, which is usually taken into account, is the expectation value, or E-value, which is defined as the number of hits expected by chance during a sequence database search of this size. The E-value of an alignment is exponentially related to its score(S): $E_{value} \propto e^{-\lambda S}$ (λ is an empirically determined constant). In practice, the E-value decreases exponentially as the score of the match increases: the lower the E-value, the more "significant" the match is. For example, assigning to a hit an E value of 1 means that in a database of the this size one might expect to see one match with a similar score simply by chance. Instead, if the E-value of the hit is

e^{-10}, you expect to see that alignment by chance e^{-10} times: the alignment is very unlike to be random. A such low E-value is a sign of a probable biological relation between the sequences. In figure 5.4, we show the BLAST output for the alignment of two sequences of 46 residues, both belonging to the SH3-1 protein family. Indeed, there are a lot of conserved residues(34/46) and of substitutions among chemically similar residues(39/46), marked with a + sign in the alignment. The E-value for this match is very low(10^{-23}), sign of a biological affinity.

```
 Score = 76.3 bits (186),  Expect = 1e-23, Method: Compositional matrix adjust.
 Identities = 34/46 (74%), Positives = 39/46 (85%), Gaps = 0/46 (0%)

Query  1   MALYDFQARSPREVTMKKGDVLTLLSSINKDWWKVEAADHQGIVPA  46
           +ALYD+  +SPREV+MKKGDVLTLL+S NKDWWKVE  D QG VPA
Sbjct  1   VALYDYTEKSPREVSMKKGDVLTLLNSNNKDWWKVEVNDRQGFVPA  46
```

FIGURE 5.4: BLAST output

Multiple Sequence Alignment

MSAs are computationally difficult to manage: even if, it is in principle possible to extend dynamic programming to many sequences, it gets extremely slow already for small numbers and it is then rarely used for more than three or four sequences. It was thus necessary for the modelers to look for approximate methods. One example is given by the so called "progressive method" (such as ClustalW [154]), which produce a MSA by first aligning the most similar sequences and successively adding less related sequences. MSA of sequences belonging to a protein family can be used to detect new members of the same family. Indeed, from the MSA, it is possible to build position-specific scoring matrices as in PSI-BLAST [155] or hidden Markov models as in HMMer [137]. We made an extensive use of this last tool in our analysis, we thus present its underlying theory in the next sections, following the explanation from Ref. [152] .

5.3.3 HMM Profiles for Sequence Families

Functional sequences typically come in families. Protein families consist of proteins having the same or related function, whose primary sequences derived from a common ancestor and have diverged along evolution. Identifying the relationship between an individual sequence and a sequence family is at the base of many sequence analysis

methods. Once the set of sequences forming a family is known, it is possible to perform a database search, looking for other members, using pairwise alignment with one of the family members as query sequence. However, this technique can be considerably improved by considering features which are conserved in the whole family. Let's consider, for example, the MSA of ten sequences belonging to the SH3-1 family (from Pfam database [95]). It is clear that some position in the SH3-1 alignment are more conserved than others, for example at the 24^{th} position we find Isoleucine in nine over ten cases. Another general feature that comes out, is that gaps tend to line up with each other, leaving solid blocks in which no insertions or deletions are present.

```
VAV_HUMAN/788-834          KARYDFCARD--RSELSLKEGDIIKILNKKGQ--QGWWRGEIY-----GRVGWFPA
HSE1_YEAST/223-268         RALYDLTTNE--PDELSFRKGDVITVLEQVYR---DWWKGALR-----GNMGIFPL
MYOC_DICDI/1129-1176       IALYEYDAMQ--PDELTFKENDVINLIKKVDA---DWWQGELVRT---KQIGMLPS
HCLS1_HUMAN/434-479        VAVYDYQGEG--SDELSFDPDDVITDIEMVDE---GWWRGRCH-----GHFGLFPA
Q6FWR1_CANGA/526-572       -AEYDYEAAE--DNELTFEENDKIINIEFVDD---DWWLGELEKT---GEKGLFPS
YKA7_CAEEL/197-244         IAKFDYAPTQ--SDEMGLRIGDTVLISKKVDA---EWFYGENQNQ---RTFGIVPS
NCF2_HUMAN/463-508         EALFSYEATQ--PEDLEFQEGDIILVLSKVNE---EWLEGECK-----GKVGIFPK
YKA7_CAEEL/277-322         TAIYDYNSNE--AGDLNFAVGSQIMVTARVNE---EWLEGECF-----GRSGIFPS
DRK_DROME/158-203          QALYDFVPQE--SGELDFRRGDVITVTDRSDE---NWWNGEIG-----NRKGIFPA
SEM5_CAEEL/160-205         QALFDFNPQE--SGELAFKRGDVITLINKDDP---NWWEGQLN-----NRRGIFPS
GRB2_CHICK/162-207         QALFDFDPQE--EGELGFRRGDFIQVLDNSDP---NWWKGACH-----GQTGMFPR
```

FIGURE 5.5: MSA of ten sequences belonging to the SH3 family.

Trying to capture the properties of a MSA of sequences, modelers resorted to a very specific type of probabilistic model, called the Hidden Markov Model or HMM ([156])). The application of this theory to the MSA of the sequences belonging to a family, creates the so called profile HMMs. In the following section, we will briefly recall the mathematical theory at base of HMMs and present the software Hmmer [137], which is based on them.

Markov Chains and Hidden Markov Models

Markov chains are models in which the probability of a symbol(a residue of the sequence in our case) depends only on the previous symbol. The probability of a sequence, of lenght L, can thus be written as

$$\begin{aligned} P(x) &= P(x_L, x_{L-1}, .., x_1) \\ &= P(x_L \mid x_{L-1}, .., x_1)P(x_{L-1} \mid x_{L-2}, .., x_1)..P(x_1) \\ &= P(x_L \mid x_{L-1})P(x_{L-1} \mid x_{L-2})P(x_2 \mid x_1)P(x_1) \end{aligned} \tag{5.6}$$

Important parameters of a Markov model are the transition probabilities (we will call them t_{AB}), which determine the probability of a certain residue to follow another one: $t_{kl} = P(x_i = l \mid x_{i-1} = k)$.

HMMs are a subset of Markov Models in which there is a distinction between the sequence of states(which is called the path π), which is unknown, and the sequence of symbols, which is observed. The path itself follows a simple Markov chain, with t_{kl} transition probabilities. We now need to introduce new parameters, which are called emission probabilities: $e_k(b) = P(x_i = b \mid \pi_i = k)$, literally the probability to observe symbol b when in state k. The reason for the name "emission probabilities" is that we can think HMMs as generative models, which emit sequences.

Given a sequence x, we are thus interested in calculating the probability that x was generated by a specific model. Since many different state paths can generate the same sequence, we need to sum the probabilities over them:

$$p(x) = \sum_{\pi} P(x, \pi) \tag{5.7}$$

where $P(x, \pi)$ is the joint probability of an observed sequence x and a state sequence π and be can calculated from the transition and emission probabilities:

$$p(x, \pi) = t_{0\pi_1} \sum_{i} e_{\pi_i}(x_i) t_{\pi_i \pi_{i+1}} \tag{5.8}$$

Note that the transition probability for the first state($t_{0\pi_1}$), which represents the probability of beginning the sequence with a specific state, is treated independently. Going back to equation 5.7, the sum over all paths is not practical, since their number increases exponentially with the lenght of the sequence. A dynamic programming procedure, called forward algorithm [157], is however able to calculate $P(x)$ in a recursive way.

One of the main difficulties to be overcome when using HMMs is establishing the model to be used. This task has two different steps, the first is to design the structure of the model which means to choose the possible states and the way in which they are connected; the second is to assign the value of the parameters (emission and transition probabilities). Generally, parameters are learnt from a training set of

example sequences. On the other hand, the choice of the topology of the model is based on the deep knowledge of the system under consideration.

HMMs Profiles

As we previously stated, our aim is to find a method able to calculate the probability that a query sequence belong to a specific protein family. In order to be efficient this method has to take into account the general properties of the MSA of sequences of the family. A possible solution comes from the HMMs theory. A specific HMM for MSA of sequences, called HMM profile, was first introduced from Krogh et al in 1994 [158]. HMM profiles capture position-specific information about how conserved each column of the alignment is, and which residues are likely in each position. The transition structure of an HMM profile is shown in figure 5.6. An advantage of this approach is the possibility to take into account the presence of insertions and deletions directly in the topology of the model through the presence of the so called insert states and delete states .

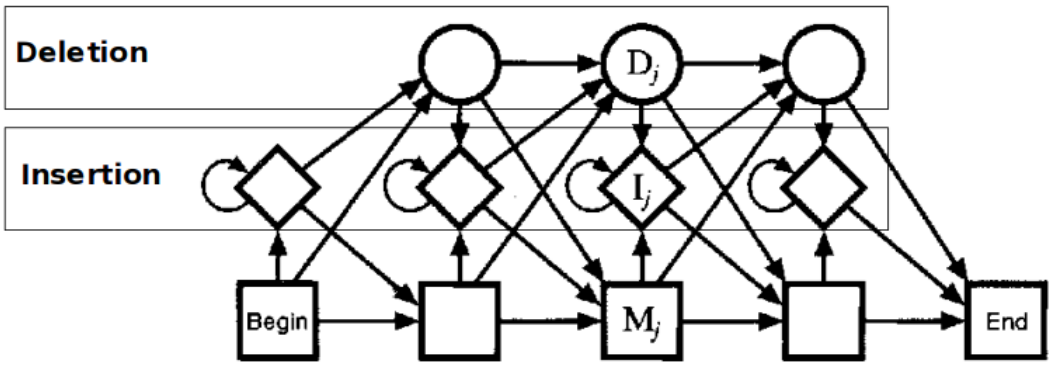


FIGURE 5.6: Schematic representation of topology of HMM profiles, taken from [152]. All possible transitions are represented, diamonds indicate the insert states, circles the delete states

Having decided the topology of the model, the parameters are trained on the MSA of the specific family of interest. The probability of having gaps becomes thus dependent on the position along the sequence, as it happens in nature. HMM profiles can be used to detect other members of the same family. This is done through the calculation of the probability that a query sequence x has been "generated" by the Hmm profile of the corresponding family ($P(x \mid M)$). We will consider the so called

expectation-value(E-value) of a specific match, which is defined as the log-odd ratio of $P(x \mid M)$ over the probability that the sequence x was generated by a random model(R) given by the equation $P(x \mid R) = \Pi_i q_{x_i}$ where q_{x_i} are the standard amino acid frequencies.

Hmmer and Pfam

Hmmer [137] is a software used to search sequence databases for homologs of protein (or DNA sequences) which is strongly based on HMM profiles. Given a MSA of sequences belonging to a family, Hmmer builds the corresponding HMM profile which can then be used to evaluate the probability that a sequence has been generated from this HMM rather then from a random model (the so called E-value).

The capability of building HMM profiles from MSAs, has allowed the creation of libraries of hundreds of profile HMMs which were then applied on a very large scale to whole genome analysis. In particular, Hmmer is strictly related to the Pfam database [95], in which proteins domains are classified into families. To do so, a combination of manual and automatic approaches is adopted. First, from a set of sequences known to be member of the family, the so called seed alignment is built. This set of sequences is manually checked, so that it is not redundant and that the representatives truly belong to the family. From the seed alignment, the HMM of the corresponding family is built and then compared with all the sequences in the protein databases Swissprot and TrEMBL [159]. Trough this procedure, all the sequences belonging to the family are selected and their MSA is the so called full alignment of the family.

5.4 Results of the Design Protocols

In this section we present the core of our investigation on protein design. First we specify which are the target proteins and the metrics we use to quantify the quality of each design. Then, we describe the behaviour of the designs obtained through Rosetta FastDesign and through the Genetic Algorithm. Finally we present a comparison of the designs obtained by the two procedures.

5.4.1 Target Proteins

To benchmark the quality of Rosetta Design and of the Genetic Algorithm proposed in this work, we performed the design of two target proteins: the first belonging to the Ubiquitin family (1ubq.pdb from Protein Data Bank [135], 76 residues), the second belonging to the SH3-1 family (SH3 domain clan- 1shg.pdb from Protein Data Bank [134], 57 residues). We chose these proteins since they belong to families which are well represented in the Pfam database, allowing an accurate bioinformatics analysis of the designed sequences. For the Ubiquitin family the Pfam-seed is composed of 61 sequences, the total number of sequences related to this family is 38111. For the SH3-1 family the Pfam-seed is composed of 55 sequences, the total number of sequences related to this family is 55784.

5.4.2 Scoring the Sequences

To evaluate the capability of the protocols of recovering the natural sequence of a target protein we estimate

- the probability that the designed sequence is evolutionary related with a natural protein belonging to the ground-truth family using BLAST
- the probability that the designed sequence belongs to its ground-truth family using Hmmer

These quantities are quantified by the two parameters called expectation values (E-values), described respectively in sections 5.3.2 for BLAST and 5.3.3 for Hmmer. Summarizing: Hmmer calculates the probability of a sequence of belonging to a protein family (measured trough the E-value) by comparing it to the HMM-profile of the family (which is built from its Pfam-seed). Sequences that score significantly better to the HMM-profile compared to a null model are considered to be homologous to the sequences that were used to construct the profile. This means higher probability of belonging to the family. Instead, BLAST performs pairwise sequence alignment by finding regions of local similarity between sequences. The pairwise alignment can be done between a target sequence against all the sequences belonging to a database, giving a measure of the sequence E-value against the database itself.

To score a sequence, we thus need a sequence database for the Ubiquitin family and for the SH3-1 family. We choose to create these databases from the sequences belonging to the Pfam-seed of the corresponding families. The database for the Ubiquitin family thus contains 61 sequences, the database for the SH3-1 family contains 55 sequences.

5.4.3 Rosetta FastDesign

We first perform the design of 1ubq and 1shg by Rosetta FastDesign. These tests are carried out to understand if Rosetta Design alone is able to recover sequences which are bioinformatically compliant with the natural sequence.
We use the framework of RosettaScripts [160], a scripting language interface which allows the specification of different modeling task in Rosetta (called Rosetta movers). First of all, we get from the pdb file the conformation of the folded state and its natural sequence. We then create two different starting conditions for the design:

- One is the natural protein, which retains the natural sequence and its folded structure. The only change we do to the original pdb is a little relaxation of the structure minimizing the Rosetta energy function (using the FastRelax mover [138]). This starting condition is then used as a reference.

- The other is a poly-valine version of the folded structure. This is obtained trough Rosetta MutateResidue mover: the backbone of the protein is fixed, but all the side chains are mutated to valine's side chain. To avoid steric clashes between atoms is then necessary a relaxation of the new protein trough FastRelax mover. After the relaxation step the structures obtained differ by approximately by $1.27\mathring{A}$ from the 1ubq structure and $1.23\mathring{A}$ for 1shg (C_α rmsd).

From these two starting conditions, we perform our flexible-backbone designs using Rosetta FastDesign mover [138], which was described in section 5.2.3. The number of complete design cycles can be controlled varying a parameter of this mover. We are interested in understanding if the sequences converge to similar solutions, and in analyzing the dependence of the solutions on the starting condition. We thus

perform, for both 1ubq and 1shg, 2500 designs for each different design situation, where the varying conditions are

- the starting sequence of the design (the natural sequence or poly-valine)
- the number of cycles of Rosetta FastDesign

For each design situation, we then calculate the average of the Rosetta Score and of the Hmmer Score (i.e $\log(Evalue)$) over the 2500 sequences. The E-values of the sequences are calculated using Hmmer, against the HMM-profile of the corresponding family (Ubiquitin family or SH3-1 family) . Moreover, for the set of sequences which are designed using the natural sequence as initial condition, the E-value is calculated also using the HMM-profile of an artificial family consisting of the 2500 sequences obtained by Rosetta FastDesign using the poly-valine sequence as initial condition.

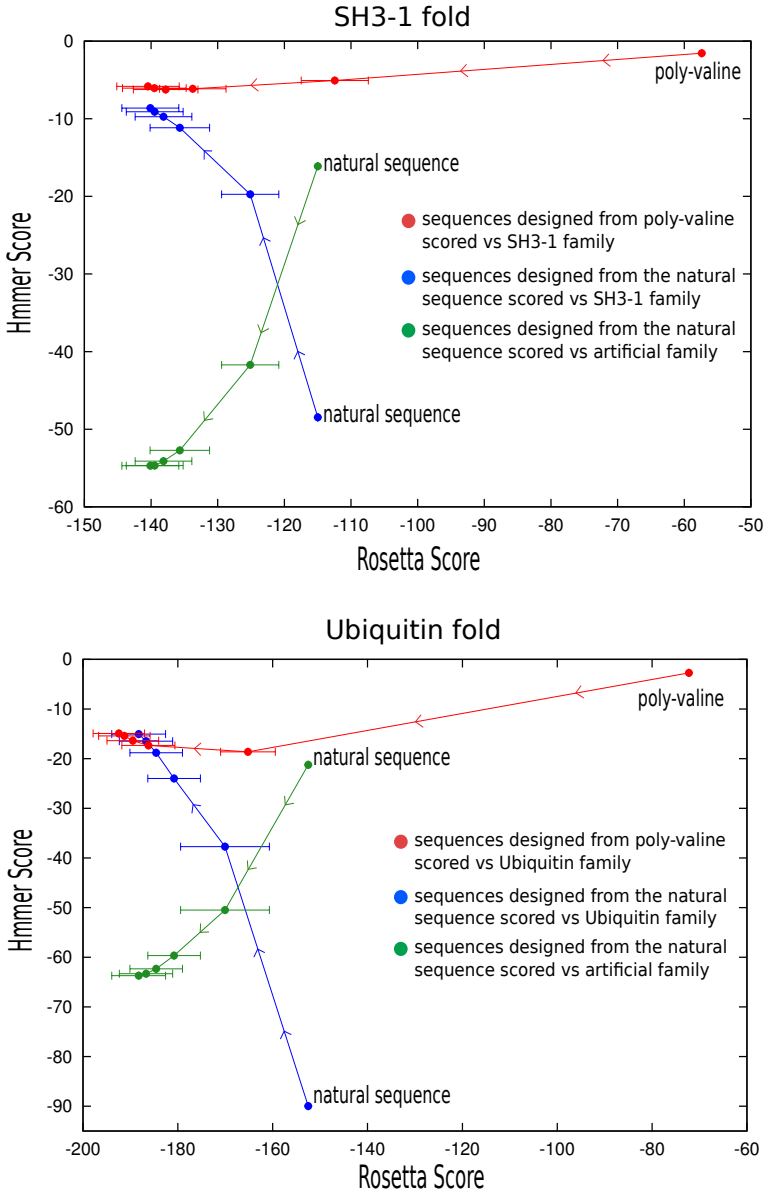


FIGURE 5.7: Top panel: design of 1shg . Evolution of the average Rosetta Score and of the average Hmmer Score increasing the number of cycles of Rosetta FastDesign (from 1 to 32 cycles). Different colors correspond to different starting sequences for the design or to different protein family against which the Hmmer Score is evaluated (see the legend). Bottom panel: respectively for the design of the 1ubq.

In each of the two panels of figure 5.7, we represent the evolution of Rosetta Score and Hmmer Score starting from the two initial conditions and increasing the number of the design cycles. The top panel is for the SH3-1 fold (1shg), the bottom panel for the ubiquitin fold (1ubq). The red line represents the evolution of a design started from poly-valine. The abscissa of every point is the Rosetta score at different design cycles. The ordinate is the Hmmer Score against the SH3-1 (top panel) and Ubiquitin

family (bottom panel). The blue line corresponds to a design started from the native sequence. The ordinate of every point is the Hmmer Score against the SH3-1 (top panel) and Ubiquitin family (bottom panel). Finally, the green line corresponds to the same design of the blue line (namely a design performed starting from the native sequence) but with the ordinate corresponding to the Hmmer score estimated against the artificial family generated by the Rosetta Design of the red line, as explained before. The arrows indicate the direction of increasing number of FastDesign cycles, from 1 to 32.

In all cases the Rosetta Score improves during the design, reaching values of approximately -140 for the design of 1shg and of -190 the design of 1ubq. However, in the designs started from poly-valine (blue lines) the optimization of the Rosetta Score is not associated with a monotonic improvement of the Hmmer Score. Indeed, the Hmmer Score improves in the first design cycles, but then becomes worse in the last cycles. In the designs started from the native sequence (red lines), the Hmmer score against the natural family becomes worse and worse, reaching at the end scores similar to those observed in poly-valine design. Rosetta design does not seem to be able to recognize the natural sequence as good sequences, and many mutations are accepted through the rounds of design. During the design started from the native sequence, the Hmmer Score against the "artificial family" improves significantly as a function of the number of cycles, reaching values of ~ -55 and ~ -65 in the 1shg and in the 1ubq design. This implies that according to Hmmer, the sequences generated starting from poly-valine and from the native sequence belong to the same family, indicating that Rosetta Design is capable of finding very similar solutions starting from totally different initial sequences. This result is remarkable, showing that the FastDesign protocol converges in sequences belonging to the same "family" regardless of the starting point, indicating an impressive robustness of the algorithm.

5.4.4 The Genetic Algorithm

The Genetic Algorithm(GA) aims at driving Rosetta flexible-backbone design towards natural sequences using hints from bioinformatics. Rosetta Design has already been combined with evolutionary conservation and covariation analyses to redesign existing

proteins [161, 162]. The scope of these works was however different from our: they aimed to enhance the stability and the activity of some target proteins.

The idea, at the base of our GA, is to drive the design procedure from the "parent" sequence (usually a poly-valine), not only trough the minimization of the Rosetta potential, but also taking into account the presence of signatures which make a sequence more *natural* than another one. Importantly, we do not drive the design towards the sequences which take the *correct* fold, but generically towards sequences observed in nature which can fold. The working principle of the algorithm is illustrated in the flow chart 5.8 .

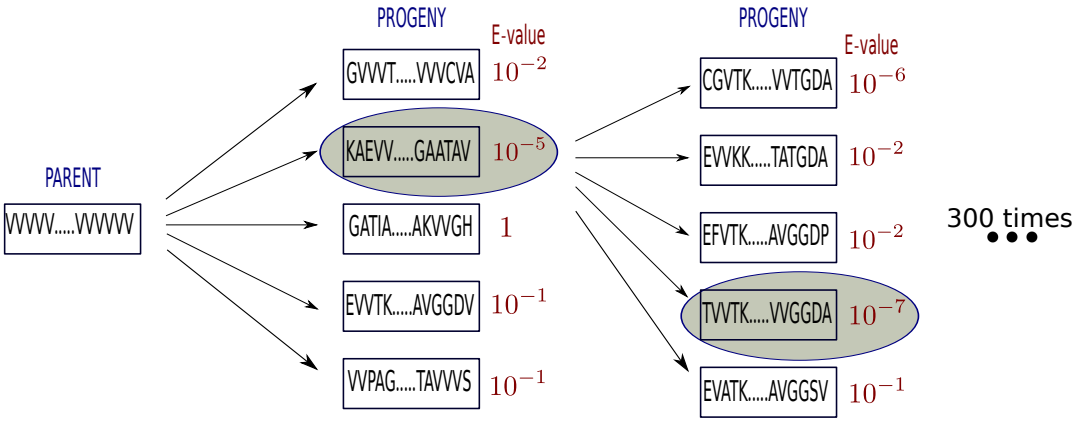


FIGURE 5.8: Schematic representation of the Genetic Algorithm procedure.

At each step of the procedure, from the "parent" sequence, 5 sequences are generated using FastDesign mover of Rosetta, allowing a maximum of 20 mutations. These 5 "progeny" sequences are then scored with BLAST using as database the ensemble of all sequences belonging to the Protein Database Bank. Among them the one with the highest E-value is chosen to become the new parent sequence.
The entire procedure is iterated for 300 steps. It's important to notice that, during the optimization, the sequences are not scored against the family databases of the target protein (ie Ubiquitin family database or SH3-1 family database), but against the database containing all the sequences of the pdb. This means that we are driving the design towards more natural sequences, regardless of the target protein.

We test the Genetic Algorithm on the same design tasks as simple Rosetta FastDesign. The top panel of the figure 5.9 shows the evolution, along the 300 steps of the procedure, of the BLAST E-value calculated against the proteome database, of

the sequences designed from poly-valine mounted on 1shg structure. The red line represents the case in which simple Rosetta FastDesign is used to generate the new sequences, without using the selection criterium of the Genetic Algorithm. The gray lines show the evolution of the E-value for 15 illustrative runs over the 100 runs of the Genetic Algorithm. We thus see that the optimization performs well in its job: the E-values reached trough the procedure are on average better than the ones obtained by simple Rosetta FastDesign. This means that the Genetic Algorithm is able to drive the design towards more natural sequences, which, we recall, are not necessarily the sequences of the target family.

The bottom panel shows the evolution of the BLAST E-value, calculated against the database of the SH3-1 family, for the same 15 illustrative runs. The lines are colored when BLAST identifies as the best possible corresponding protein, a protein belonging to SH3-1 family. In 5 out of 15 runs the final sequence has SH3-1 as best BLAST corresponding family (5 lines are colored at the step 300). Moreover, the optimization of the E-value against the database of the whole proteome is associated with a decreasing of the E-value against the database of the SH3-1 family. The sequences are not only more "BLAST-compliant", but more similar to the ones belonging to the SH3-1 family. This is obtained automatically: using the BLAST score, the algorithm is able to drive the sequences towards "correct" ones.

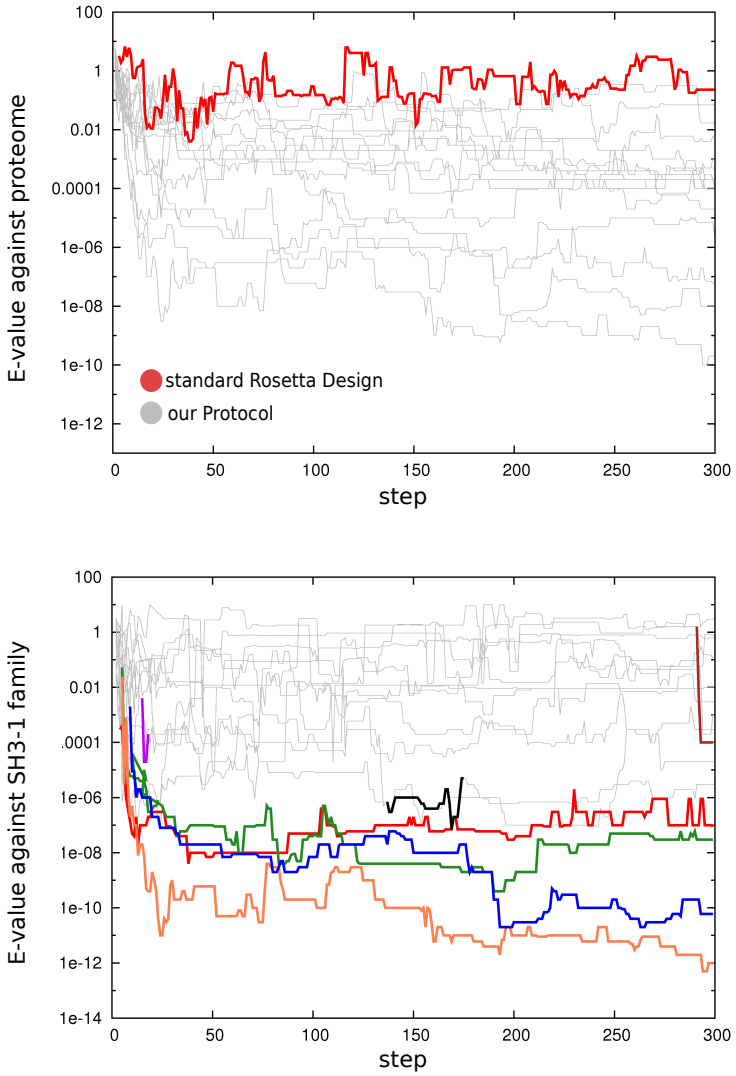


FIGURE 5.9: Top panel: The grey lines show the evolution of the BLAST E-value against the proteome database, along the 300 steps of the Genetic Algorithm procedure (15 illustrative runs). The red line shows the evolution of the BLAST E-value against the proteome database in a run in which only Rosetta FastDesign is used, without any optimization of the BLAST E-value. Bottom panel: Evolution of the BLAST E-value against the database of SH3-1 family, for the same 15 illustrative runs of the top panel. The lines are colored when BLAST identifies SH3-1 as best corresponding family.

5.4.5 Comparison of FastDesign vs Genetic Algorithm

For both 1ubq and 1shg we carry out a detailed comparison of the E-values, calculated with respect to their original families databases, of the following groups of sequences:

- the 2500 sequences generated through Rosetta FastDesign (8 repeats), starting from poly-valine

- the 100 sequences obtained at the end of the optimization procedure of the Genetic Algorithm
- the sequences belonging to the Pfam seed of the corresponding family: 61 sequences for Ubiquitin family (referred as ubiquitin from seed), 55 sequences for SH3-1 family (referred as SH3-1 from seed).

The behaviour of these different groups of sequences is presented in the four panels of figure 5.10, which show the cumulative probability of the E-values. The E-values are calculated both with Hmmer (left panels) and BLAST (right panels). The two upper panels are for 1ubq, the two lower ones for 1shg.

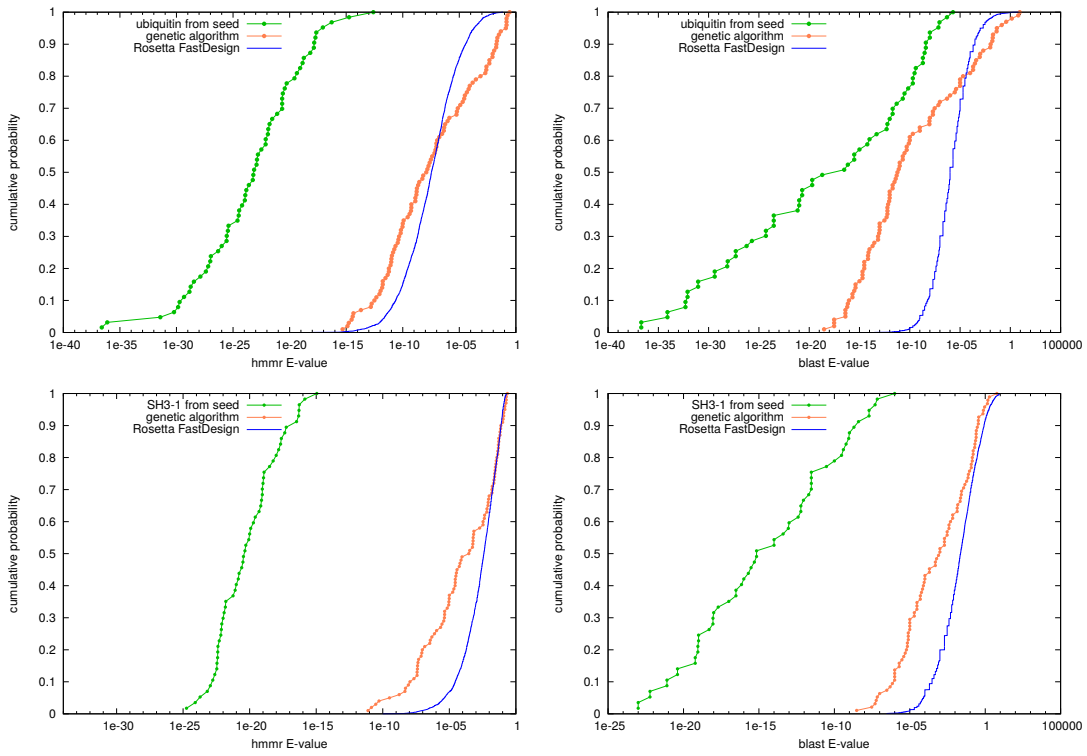


FIGURE 5.10: Cumulative probabilities of the E-values. Panels a and b refer to 1ubq design, the scores are calculated against the Ubiquitin family database (hmmr/BLAST). Panels c and d refer to 1sh3 design, the scores are calculated against the SH3-1 family database (hmmr/BLAST). Different colors refer to different groups of families as indicated in the legend.

Generally, we see that the E-values of the sequences obtained from the Genetic Algorithm (orange cumulative curves) are moving towards the E-values of the sequences belonging to the original families (green cumulative curves), improving the results of Rosetta FastDesign (blue cumulative curves). This doesn't happen for all the 100 sequences from the GA: sometimes the algorithm optimizes the E-value against the

proteome database "reproducing" the sequence of a protein belonging to a different family. If this happens for the majority of the 300 steps of the procedure, the designed sequence drifts away from the original one. However, the results are encouraging: at the end of the 300 steps of the optimization, 77 out of 100 sequences have Ubiquitin as best corresponding family and 40 out of 100 sequences have SH3-1 as best corresponding family. The improving of the E-values after the optimization procedure is more evident if we score the sequences using BLAST: in this case the "optimised" sequences are almost reaching the E-values of the sequences from the original family both for SH3-1 and for Ubiquitin.

5.5 Experimental Characterization of Designed Proteins

To test whether the predicted sequences could yield folded proteins, we selected some sequences for experimental characterization, which was performed in the group of Prof. Correia at EPFL (Lausanne). We focused on the SH3-1 designed series, and tested experimentally a set of sequences generated using both the GA approach and simple Rosetta FastDesign. Among the 100 sequences obtained trough the GA we selected the ones which are more similar to the natural SH3-1, which means the six sequences with best Hmmer E-value. On the other hand, among the 2500 sequences obtained after 24 cycles of Rosetta FastDesign, we selected ten according to the following procedure:

- we select the sequences with a good packing of the aminoacids (RosettaHoles [163] score below 0, packstat [164] score above 0.65)
- among those, we select the 100 sequences with the best Rosetta Score
- we finally select the 10 sequences with the best correspondence between the secondary structure predicted from sequence only (using psi-pred [165]) and the secondary assignment obtained from the structure by DSSP [166, 167].

The designs were expressed in bacteria and those that were soluble and purifiable were further characterized according to their oligomerization state in solution using size exclusion chromatography coupled to a multi angle light scattering (SEC-MALS).

Folding and thermal stability were characterized using circular dichroism (CD) spectroscopy. We here present the experimental protocol and the obtained results.

5.5.1 Protocols

Protein Expression and Purification

DNA encoding the sequences of the tested proteins was purchased from Twist Bioscience as DNA fragments, which were cloned into pET11b or pET21b expression vectors using Gibson cloning. A 6x His tag was added at the C terminus of the sequences to facilitate the purification. Plasmids were transformed into E.coli BL21 (DE3) (Merck), and grown overnight in LB media supplemented with 100 μg/ml ampicillin. Overnight cultures were diluted 1:50 in TB medium and grown at 37°C until the OD600 reached 0.6-0.8. To induce expression, 1 mM of isopropyl β-d-1-thiogalactopyranoside (IPTG) was added and cells were grown for 12-16 hours at 22°C. Cultures were harvested and resuspended in lysis buffer (50 mM Tris, pH 7.5, 500 mM NaCl, 5 % glycerol, 1 mg/ml lysozyme, 1 mM PMSF, 1 μg/ml DNase), and lysed by sonication. The cell lysate was pelleted by centrifugation (20,000 rpm, 20 mins) and supernatant was filtered with a 0.22 μm filter before loading onto a 1 ml HisTrap HP column (GE Healthcare). Proteins bound to the column were washed with 10 column volumes of washing buffer (50 mM Tris, pH 7.5, 500 mM NaCl, 10 mM imidazole) and eluted in 5 column volumes of elution buffer (50 mM Tris, pH 7.5, 500 mM NaCl, 300 mM imidazole). Eluted proteins were further purified by size exclusion chromatography on a Hiload 16/600 Superdex 75 pg column (GE Healthcare) in PBS buffer.

Size-Exclusion Chromatography Coupled with Multi-Angle Light Scattering (SEC-MALS)

SEC-MALS was performed on a HPLC system (Thermo Fisher) connected to a light scattering detector (miniDAWN TREOS, Wyatt). 100 μl of freshly purified protein (concentration 1-2 mg/ml) was injected on a Superdex 75 300/10 GL column (GE Healthcare) at a flow rate of 0.5 ml/min. UV absorption and light scattering were recorded and processed using the ASTRA software (version 6.1, Wyatt).

Circular Dichroism

All circular dichroism data were collected on a Chirascan CD spectrometer (Applied Photophysics) using a quartz cuvette with path length of 1 mm. Purified proteins were diluted in 10 mM sodium phosphate buffer pH 7.4 to a final concentration of 30 μM. Far UV spectra were recorded between a wavelength of 190 nm and 250 nm with a scanning speed of 20 nm/min. The spectra were averaged from two repeated measurements and corrected for buffer absorption. To determine the thermostability of the designed proteins, temperature was ramped stepwise from 25°C to 95°C in increments of 2°C in the presence of 2.5 mM TCEP reducing agent. Thermal denaturation curves were plotted by the change of ellipticity at the global curve minimum and fitted with the sigmoidal two-state model to determine the melting temperature (Tm) using Prism 8 (GraphPad).

5.5.2 Results of the Experimental Validation

From the six GA designs, 2 were expressed, soluble and monomeric in solution (panel a of figure 5.11). For the Rosetta FastDesign designs, 3 were soluble and purified, however their solution behavior was not optimal according to the SEC-MALS elution profiles. The secondary structure analysis by CD revealed that from all the designs tested, only $SGD44$ showed a very similar secondary structure signature to that of the native protein, specifically the two well defined minima at approximately 202 and 228 nm (panel b of figure 5.11). Interestingly, $SGD44$ showed the same melting temperature than the native sequence.

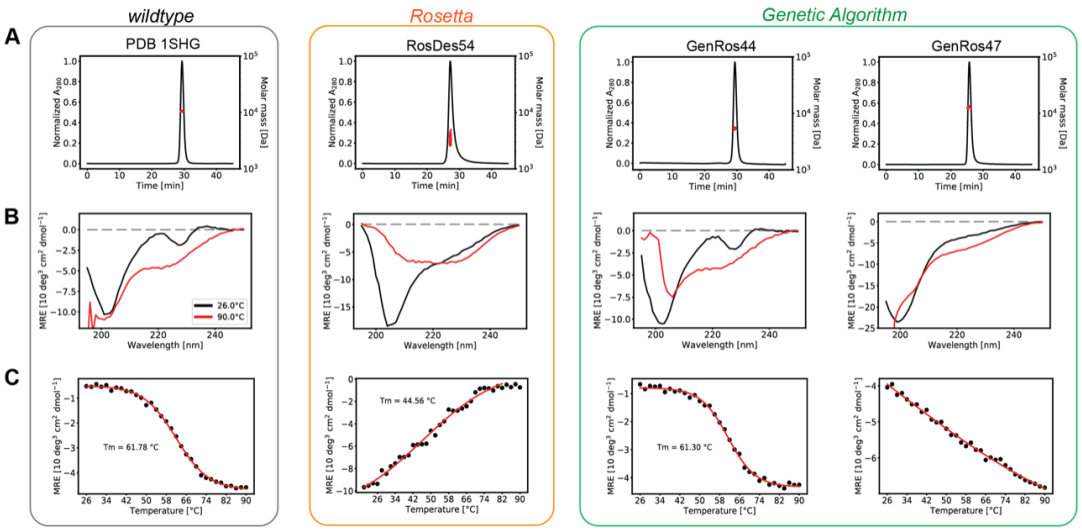


FIGURE 5.11: Biochemical analysis of the wild-type (WT) 1shg and designs. (a) SEC-MALS for WT 1shg and designs. Only the WT and the designs generate by the SGD (GenRos44 and GenRos47) show a profile indicating a clear monomeric form. (b) CD spectra for WT 1shg and designs. Two peaked signals at around 205 nm and 230 nm for the WT, GenRos44 indicate similar secondary structure content. (c) Thermal melting CD spectra (at 200 nm) are shown

5.6 Discussion and Conclusions

In our work, we first tested the capability of Rosetta Design of recovering the natural sequence of a protein, using as benchmark the Ubiquitin and Sh3-1 folds. The probability that a designed sequence belongs to the original families, is evaluated trough the E-value parameter, which is chosen as a measure of the quality of the designed sequences. Importantly, in the spirit of this thesis, we choose to avoid any expert curation, in order to analyze the quality of the algorithm as a "black box", well aware of the difficulty of this challenge. We find that, if backbone flexibility is allowed, the sequences generated through Rosetta Design are not recognised as belonging to the original families. This result appears to be in contrast with the findings of a previous work, stating that Rosetta Design allowing backbone flexibility, reproduces the sequence variability observed in nature better than the approach with fixed backbone [133]. However, in this work the manner of quantifying the consistency between a designed sequence and the natural one is very different form ours. In particular, in this work the consistency is measured from the cross entropy between the designed and the natural sequence. This quantity can be small if an aminoacid is with high

probability the correct one, but also if the amino acid in the design sequence is almost always the same, but never the correct one.

A second important finding is that, for a given backbone configuration, the designed sequence converge to similar solutions, independently from the sequence from which the design is started. This means that Rosetta protocol is robust and almost deterministically brings to a specific tiny region in sequence space, which however, in the two cases we considered, does not correspond to the sequence of natural proteins with the same structure. Of course it is possible (and even likely) that in other folds Rosetta Design will be able to find sequences more compliant with the natural sequences. We hope that our work will encourage extensive and systematic benchmark tests of Rosetta Design on a large number of folds, estimating the quality of the results with the metrics proposed in this work.

In order to improve the capability of Rosetta Design to find natural sequences, we propose a Genetic Algorithm in which, along the steps of the design procedure, the E-value against a database of natural sequences is iteratively improved. The aim is to select sequences that at the same time satisfy the requirement of having a low Rosetta score and of being similar to the natural ones. The design of the proteins 1ubq and 1shg by the GA gives encouraging results. Indeed, a high percentage of the designed sequences is bionformatically recognized as belonging to the corresponding family, and in general the sequences E-values against the corresponding natural families become lower than the ones obtained with Rosetta Design. Moreover, among the sequences that were experimentally tested only designs from the GA behave as monomers in solution. For one of them the CD spectrum and the unfolding curve resemble closely the one of the 1shg wild type, meaning that the two proteins are likely to have a similar secondary structure.

A weak point of the GA is that, the sequences of the database used for the E-value optimization includes the sequences of the family of the target protein. We verified that if one attempts designing the 1shg fold using a database of sequences in which if all the sequences of SH3-1 family are removed, the GA loses its ability of driving the design towards the correct solution. This is a problem in the case of a fully *de novo* design, in which the desired conformation does not exist in nature. One possible solution to overcome this issue, could be to optimize the alignments of multiple pieces

of the designed sequence with pieces of existing proteins, instead of optimizing the alignment of the whole sequence. The idea is to find the sequence that adopts a specific backbone, linking strings of existing proteins, with each string corresponding to a specific secondary structure element of the target backbone. We are aware that a lot of work is necessary to test this idea and make it work in in practice.

In conclusion, in our work, we pinpoint a possible pitfall of Rosetta flexible backbone design procedure and suggest a possible direction to find a solution. In general, we believe that taking into account the information contained in the huge quantity of natural sequences can lead to important improvements in the field of protein design.

Concluding Remarks

This thesis discusses the results obtained during my phd on two topics which are very different: the study of free energy landscapes and the development of algorithms for protein design. The unifying element of these two topics is the attempt to reduce the amount of human curation in solving problems, in the spirit of unsupervised techniques.

The first field we investigate is the study of the free energy landscapes explored in MD simulations of biomolecules. In chapter 2 we presented a procedure devoted to this aim. In chapters 3 and 4 we presented the results of its application for the study of the free energy landscapes of two different proteins.

The first key step of our procedure, is the estimate of the free energy and of its uncertainty, for each data point, using the PAk estimator. Importantly, this estimate is performed in the manifold on which the data actually lie, the so called embedding manifold. This is a great advantage, with respect to previous techniques (such as the ones presented in [5, 12, 13]), in the case in which the embedding manifold is topologically complex. Indeed, in such a situation, any representation obtained through the projection of the data necessarily introduces errors. The PAk estimator only requires the knowledge of the Intrinsic Dimension of the embedding space, and does not require the explicit definition of the variables defining it. In our approach, the Intrinsic Dimension is assumed to be constant in all the dataset. However, techniques for relaxing this hypothesis have already been developed [168]. In perspective, it would be interesting to extend the approach described in this thesis to situations in which the Intrinsic Dimension is not constant.

The second key step is the detection of the relevant states for describing a biomolecule from the direct analysis of the free energy landscape, through a clustering technique.

Our procedure includes two of them, both strictly connected to the PAk estimator. In the first one, which is directly akin to DP clustering [58], the metastable states correspond to the free energy basins. This algorithm explicitly depends on a single parameter Z, defined in equation 2.22. However, since the PAk estimator calculates the value of the uncertainty of the free energy through a precise mathematical model, the parameter Z has a well-defined meaning: it is the statistical confidence at which a basin is considered meaningful. The second clustering algorithm is the $\hat{k}$-Peaks clustering, which represents the main algorithmic novelty of this thesis. In this algorithm the metastable states are both the free energy basins and the large flat regions of the free energy corresponding to entropic traps. Also this algorithm explicitly depends on a single parameter Z defined as equation 2.23. However, the mathematical model at the basis of PAk estimator does not provide an estimate of the uncertainty of the number of neighbours for which the density around each point can be considered constant ($\hat{k}_i$ in equation 2.17). We thus choose to estimate the uncertainty of this variable for each data point, as its standard deviation among the points which are inside the constant density neighborhood of the selected point. This procedure is not statistically grounded. Therefore, the parameter Z doesn't have a rigorous statistical interpretation. A possible improvement of the $\hat{k}$-Peaks clustering is thus finding a strategy to calculate the statistical significance of the detected states.
The presented procedure is very general: it can be exploited for the study of any system simulated through MD. Many process have already been characterized using the "DP version" of the whole approach such as the behaviour of water networks around biomolecules [169], RNA base fraying [170], the forming of dendritic voids in liquid water [171]. Moreover, the "$\hat{k}$-Peaks version" of the whole approach can be a useful tool to analyse systems that undergoes a phase transition from a disordered state to an ordered one. For example, this algorithm could bring new insights in the study of intrinsically disordered proteins.

The second field we investigate is protein design. We aimed at developing an unsupervised version of the Rosetta Design algorithms which, given a natural protein backbone as input, is able to find sequences similar to the ones of the corresponding protein family. To do so, we devised a Genetic Algorithm in which the design steps are

combined with a progressive improvement of the similarity of the designed sequences with the sequences belonging to a database of natural sequences. Importantly, along the optimization we don't give any information about the family membership of the input structure. Applying the Genetic Algorithm, we obtained sequences which are more similar to the natural ones, if compared with the sequences obtained simply using Rosetta Design. A weak point of this approach is that, the sequences of the database used for the optimization include the sequences of the family of the target protein. Indeed, we verified that if one attempts to design the SH3-1 fold using a database of sequences in which if all the sequences of SH3-1 family are removed, the Genetic Algorithm loses its ability of driving the design towards the correct solution. This is a problem in the case of a fully *de novo* design, in which the desired conformation does not exist in nature. One possible solution to overcome this problem, could be to optimize the alignments of multiple fragments of the designed sequence with regions of existing proteins, instead of optimizing the alignment of the whole sequence. The idea is to find the sequence that adopts a specific backbone, linking pieces of sequences taken from existing proteins, with each piece corresponding to a specific subdomain or secondary structure element of the target structure. A more radical solution could be improving directly the RosettaEnergy function. The value of these parameters should be set in such a way that the constraint of finding natural sequence upon design is satisfied for a set of protein families.

In conclusion, in the last chapter of this thesis, we pinpoint a possible pitfall of Rosetta flexible backbone design procedure and suggest a possible direction to find a solution, even if we are fully aware that the proposed solution is far from optimal. In general, we believe that taking into account the information contained in the huge quantity of natural sequences can lead to important improvements in the field of protein design.

Bibliography

[1] Xiaofeng, M. & Xiang, C. Big data management: concepts, techniques and challenges. *Journal of computer research and development* **50**, 146 (2013).

[2] Chen, M., Mao, S. & Liu, Y. Big data: A survey. *Mobile networks and applications* **19**, 171–209 (2014).

[3] Sittel, F. & Stock, G. Robust density-based clustering to identify metastable conformational states of proteins. *Journal of chemical theory and computation* **12**, 2426–2435 (2016).

[4] Villa, P. & Roebroeks, W. Neandertal demise: an archaeological analysis of the modern human superiority complex. *PLoS one* **9**, e96424 (2014).

[5] Amadei, A., Linssen, A. B. & Berendsen, H. J. Essential dynamics of proteins. *Proteins: Structure, Function, and Bioinformatics* **17**, 412–425 (1993).

[6] Karpen, M. E., Tobias, D. J. & Brooks III, C. L. Statistical clustering techniques for the analysis of long molecular dynamics trajectories: analysis of 2.2-ns trajectories of ypgdv. *Biochemistry* **32**, 412–420 (1993).

[7] Van Aalten, D. *et al.* Protein dynamics derived from clusters of crystal structures. *Biophysical journal* **73**, 2891–2896 (1997).

[8] de Groot, B. L., Daura, X., Mark, A. E. & Grubmüller, H. Essential dynamics of reversible peptide folding: memory-free conformational dynamics governed by internal hydrogen bonds. *Journal of molecular biology* **309**, 299–313 (2001).

[9] Tournier, A. L. & Smith, J. C. Principal components of the protein dynamical transition. *Physical review letters* **91**, 208106 (2003).

[10] Lange, O. F. *et al.* Recognition dynamics up to microseconds revealed from an rdc-derived ubiquitin ensemble in solution. *science* **320**, 1471–1475 (2008).

[11] Das, P., Moll, M., Stamati, H., Kavraki, L. E. & Clementi, C. Low-dimensional, free-energy landscapes of protein-folding reactions by nonlinear dimensionality reduction. *Proceedings of the National Academy of Sciences* **103**, 9885–9890 (2006).

[12] Tenenbaum, J. B., De Silva, V. & Langford, J. C. A global geometric framework for nonlinear dimensionality reduction. *science* **290**, 2319–2323 (2000).

[13] Schölkopf, B., Smola, A. & Müller, K.-R. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation* **10**, 1299–1319 (1998).

[14] Dempster, A. P., Laird, N. M. & Rubin, D. B. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)* **39**, 1–22 (1977).

[15] Epanechnikov, V. A. Non-parametric estimation of a multivariate probability density. *Theory of Probability & Its Applications* **14**, 153–158 (1969).

[16] Lepskiui, O. V. A problem of adaptive estimation in gaussian white noise. *Teoriya Veroyatnostei i ee Primeneniya* **35**, 459–470 (1990).

[17] Mack, Y. & Rosenblatt, M. Multivariate k-nearest neighbor density estimates. *Journal of Multivariate Analysis* **9**, 1–15 (1979).

[18] Ester, M., Kriegel, H.-P., Sander, J., Xu, X. *et al.* A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, vol. 96, 226–231 (1996).

[19] Ankerst, M., Breunig, M. M., Kriegel, H.-P. & Sander, J. Optics: ordering points to identify the clustering structure. *ACM Sigmod record* **28**, 49–60 (1999).

[20] Rodriguez, A. & Laio, A. Clustering by fast search and find of density peaks. *Science* **344**, 1492–1496 (2014).

[21] Campello, R. J., Moulavi, D. & Sander, J. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, 160–172 (Springer, 2013).

[22] Cossio, P. *et al.* Exploring the universe of protein structures beyond the protein data bank. *PLoS Comput Biol* **6**, e1000957 (2010).

[23] Kuhlman, B. *et al.* Design of a novel globular protein fold with atomic-level accuracy. *science* **302**, 1364–1368 (2003).

[24] Correia, B. E. *et al.* Proof of principle for epitope-focused vaccine design. *Nature* **507**, 201 (2014).

[25] Jiang, L. *et al.* De novo computational design of retro-aldol enzymes. *science* **319**, 1387–1391 (2008).

[26] Azoitei, M. L. *et al.* Computation-guided backbone grafting of a discontinuous motif onto a protein scaffold. *Science* **334**, 373–376 (2011).

[27] Vendruscolo, M. & Dobson, C. M. Towards complete descriptions of the free–energy landscapes of proteins. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **363**, 433–452 (2004).

[28] Tuckerman, M. *Statistical mechanics: theory and molecular simulation* (Oxford university press, 2010).

[29] Beveridge, D. L. & DiCapua, F. Free energy via molecular simulation: applications to chemical and biomolecular systems. *Annual review of biophysics and biophysical chemistry* **18**, 431–492 (1989).

[30] Christ, C. D., Mark, A. E. & Van Gunsteren, W. F. Basic ingredients of free energy calculations: a review. *Journal of computational chemistry* **31**, 1569–1582 (2010).

[31] Ensing, B., Laio, A., Parrinello, M. & Klein, M. L. A recipe for the computation of the free energy barrier and the lowest free energy path of concerted reactions. *The journal of physical chemistry B* **109**, 6676–6687 (2005).

[32] Bussi, G. & Laio, A. Using metadynamics to explore complex free-energy landscapes. *Nature Reviews Physics* 1–13 (2020).

[33] Noé, F. & Fischer, S. Transition networks for modeling the kinetics of conformational change in macromolecules. *Current opinion in structural biology* **18**, 154–162 (2008).

[34] Prinz, J.-H. *et al.* Markov models of molecular kinetics: Generation and validation. *The Journal of chemical physics* **134**, 174105 (2011).

[35] Chodera, J. D. & Noé, F. Markov state models of biomolecular conformational dynamics. *Current opinion in structural biology* **25**, 135–144 (2014).

[36] Bowman, G. R., Pande, V. S. & Noé, F. *An introduction to Markov state models and their application to long timescale molecular simulation*, vol. 797 (Springer Science & Business Media, 2013).

[37] Hartigan, J. A. & Wong, M. A. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* **28**, 100–108 (1979).

[38] Ward Jr, J. H. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association* **58**, 236–244 (1963).

[39] Jain, A. & Stock, G. Identifying metastable states of folding proteins. *Journal of chemical theory and computation* **8**, 3810–3819 (2012).

[40] Schütte, C., Fischer, A., Huisinga, W. & Deuflhard, P. A direct approach to conformational dynamics based on hybrid monte carlo. *Journal of Computational Physics* **151**, 146–168 (1999).

[41] Deuflhard, P. & Weber, M. Robust perron cluster analysis in conformation dynamics. *Linear algebra and its applications* **398**, 161–184 (2005).

[42] Bowman, G. R. & Pande, V. S. Protein folded states are kinetic hubs. *Proceedings of the National Academy of Sciences* **107**, 10890–10895 (2010).

[43] Lane, T. J., Bowman, G. R., Beauchamp, K., Voelz, V. A. & Pande, V. S. Markov state model reveals folding and functional dynamics in ultra-long md trajectories. *Journal of the American Chemical Society* **133**, 18413–18419 (2011).

[44] Beauchamp, K. A., McGibbon, R., Lin, Y.-S. & Pande, V. S. Simple few-state models reveal hidden complexity in protein folding. *Proceedings of the National Academy of Sciences* **109**, 17807–17813 (2012).

[45] Pietrucci, F., Marinelli, F., Carloni, P. & Laio, A. Substrate binding mechanism of hiv-1 protease from explicit-solvent atomistic simulations. *Journal of the American Chemical Society* **131**, 11811–11818 (2009).

[46] Kohlhoff, K. J. *et al.* Cloud-based simulations on google exacycle reveal ligand modulation of gpcr activation pathways. *Nature chemistry* **6**, 15 (2014).

[47] Farimani, A. B., Feinberg, E. & Pande, V. Binding pathway of opiates to μ-opioid receptors revealed by machine learning. *Biophysical Journal* **114**, 62a–63a (2018).

[48] Cossio, P., Laio, A. & Pietrucci, F. Which similarity measure is better for analyzing protein structures in a molecular dynamics trajectory? *Physical Chemistry Chemical Physics* **13**, 10421–10425 (2011).

[49] Bonomi, M. *et al.* Plumed: A portable plugin for free-energy calculations with molecular dynamics. *Computer Physics Communications* **180**, 1961–1972 (2009).

[50] Piana, S. & Laio, A. Advillin folding takes place on a hypersurface of small dimensionality. *Physical review letters* **101**, 208101 (2008).

[51] Ceruti, C. *et al.* Danco: An intrinsic dimensionality estimator exploiting angle and norm concentration. *Pattern recognition* **47**, 2569–2581 (2014).

[52] Kégl, B. Intrinsic dimension estimation using packing numbers. In *Advances in neural information processing systems*, 697–704 (2003).

[53] Levina, E. & Bickel, P. J. Maximum likelihood estimation of intrinsic dimension. In *Advances in neural information processing systems*, 777–784 (2005).

[54] Fan, M., Qiao, H. & Zhang, B. Intrinsic dimension estimation of manifolds by incising balls. *Pattern Recognition* **42**, 780–787 (2009).

[55] Facco, E., d'Errico, M., Rodriguez, A. & Laio, A. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Scientific reports* **7**, 12140 (2017).

[56] Rodriguez, A., d'Errico, M., Facco, E. & Laio, A. Computing the free energy without collective variables. *Journal of chemical theory and computation* **14**, 1206–1215 (2018).

[57] Neyman, J. & Pearson, E. S. Ix. on the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* **231**, 289–337 (1933).

[58] d'Errico, M., Facco, E., Laio, A. & Rodriguez, A. Automatic topography of high-dimensional data sets by non-parametric density peak clustering. *arXiv* arXiv–1802 (2018).

[59] Pillai, S. U., Suel, T. & Cha, S. The perron-frobenius theorem: some of its applications. *IEEE Signal Processing Magazine* **22**, 62–75 (2005).

[60] Schütte, C. & Sarich, M. *Metastability and Markov State Models in Molecular Dynamics*, vol. 24 (American Mathematical Soc., 2013).

[61] Zhou, P. *et al.* A pneumonia outbreak associated with a new coronavirus of probable bat origin. *Nature* **579**, 270–273 (2020). URL http://dx.doi.org/10.1038/s41586-020-2012-7.

[62] Wu, F. *et al.* A new coronavirus associated with human respiratory disease in china. *Nature* **579**, 265–269 (2020).

[63] Jin, Z. *et al.* Structure of Mpro from SARS-CoV-2 and discovery of its inhibitors. *Nature* **582**, 289–293 (2020). URL http://dx.doi.org/10.1038/s41586-020-2223-y.

[64] Zhang, L. *et al.* Crystal structure of SARS-CoV-2 main protease provides a basis for design of improved a-ketoamide inhibitors. *Science* **368**, 409–412 (2020).

[65] Pillaiyar, T., Manickam, M., Namasivayam, V., Hayashi, Y. & Jung, S. H. An overview of severe acute respiratory syndrome-coronavirus (SARS-CoV) 3CL protease inhibitors: Peptidomimetics and small molecule chemotherapy. *Journal of Medicinal Chemistry* **59**, 6595–6628 (2016).

[66] Anand, K. *et al.* Structure of coronavirus main proteinase reveals combination of a chymotrypsin fold with an extra α-helical domain. *EMBO Journal* **21**, 3213–3224 (2002).

[67] Yang, H. *et al.* The crystal structures of severe acute respiratory syndrome virus main protease and its complex with an inhibitor. *Proceedings of the National Academy of Sciences* **100**, 13190–13195 (2003).

[68] Pant, S., Singh, M., Ravichandiran, V., Murty, U. & Srivastava, H. K. Peptide-like and small-molecule inhibitors against covid-19. *Journal of Biomolecular Structure and Dynamics* 1–10 (2020).

[69] Mittal, L., Kumari, A., Srivastava, M., Singh, M. & Asthana, S. Identification of potential molecules against covid-19 main protease through structure-guided virtual screening approach. *Journal of Biomolecular Structure and Dynamics* 1–26 (2020).

[70] Jiménez-Alberto, A., Ribas-Aparicio, R. M., Aparicio-Ozores, G. & Castelán-Vega, J. A. Virtual screening of approved drugs as potential sars-cov-2 main protease inhibitors. *Computational biology and chemistry* **88**, 107325 (2020).

[71] Chen, Y. W., Yiu, C.-P. B. & Wong, K.-Y. Prediction of the sars-cov-2 (2019-ncov) 3c-like protease (3cl pro) structure: virtual screening reveals velpatasvir, ledipasvir, and other drug repurposing candidates. *F1000Research* **9** (2020).

[72] Gentile, D. *et al.* Putative inhibitors of sars-cov-2 main protease from a library of marine natural products: A virtual screening and molecular modeling study. *Marine drugs* **18**, 225 (2020).

[73] Kandeel, M. & Al-Nazawi, M. Virtual screening and repurposing of fda approved drugs against covid-19 main protease. *Life sciences* 117627 (2020).

[74] Cooper, A. & Dryden, D. T. Allostery without conformational change - A plausible model. *European Biophysics Journal* **11**, 103–109 (1984).

[75] Ma, B., Tsai, C. J., Haliloğlu, T. & Nussinov, R. Dynamic allostery: Linkers are not merely flexible. *Structure* **19**, 907–917 (2011).

[76] Nussinov, R. & Tsai, C.-J. Allostery in disease and in drug discovery. *Cell* **153**, 293–305 (2013).

[77] Hauser, A. S., Attwood, M. M., Rask-Andersen, M., Schiöth, H. B. & Gloriam, D. E. Trends in gpcr drug discovery: new agents, targets and indications. *Nature reviews Drug discovery* **16**, 829–842 (2017).

[78] Conn, P. J., Christopoulos, A. & Lindsley, C. W. Allosteric modulators of gpcrs: a novel approach for the treatment of cns disorders. *Nature reviews Drug discovery* **8**, 41–54 (2009).

[79] Pargellis, C. *et al.* Inhibition of p38 map kinase by utilizing a novel allosteric binding site. *Nature structural biology* **9**, 268–272 (2002).

[80] Wu, P., Clausen, M. H. & Nielsen, T. E. Allosteric small-molecule kinase inhibitors. *Pharmacology & therapeutics* **156**, 59–68 (2015).

[81] De Smet, F., Christopoulos, A. & Carmeliet, P. Allosteric targeting of receptor tyrosine kinases. *Nature biotechnology* **32**, 1113–1120 (2014).

[82] D. E. Shaw Research. Molecular Dynamics Simulations Related to SARS-CoV-2 (2020). URL http://www.deshawresearch.com/resources_sarscov2.html.

[83] Cocina, F., Vitalis, A. & Caflisch, A. Sapphire-Based Clustering. *Journal of Chemical Theory and Computation* **16**, 6383–6396 (2020).

[84] Bacha, U., Barrila, J., Velazquez-Campoy, A., Leavitt, S. A. & Freire, E. Identification of Novel Inhibitors of the SARS Coronavirus Main Protease 3CLpro. *Biochemistry* **43**, 4906–4912 (2004).

[85] Bzówka, M. *et al.* Structural and evolutionary analysis indicate that the sars-COV-2 mpro is a challenging target for small-molecule inhibitor design. *International Journal of Molecular Sciences* **21**, 3099 (2020).

[86] Grottesi, A. *et al.* Computational Studies of SARS-CoV-2 3CLpro: Insights from MD Simulations. *International Journal of Molecular Sciences* **21**, 5346 (2020). URL https://www.mdpi.com/1422-0067/21/15/5346.

[87] Humphrey, W., Dalke, A. & Schulten, K. VMD: Visual molecular dynamics. *Journal of Molecular Graphics* **14**, 33–38 (1996). URL https://linkinghub.elsevier.com/retrieve/pii/0263785596000185.

[88] Hussein, H. A. *et al.* Pockdrug-server: a new web server for predicting pocket druggability on holo and apo proteins. *Nucleic acids research* **43**, W436–W442 (2015).

[89] Sztain, T., Amaro, R. & McCammon, J. A. Elucidation of cryptic and allosteric pockets within the SARS-CoV-2 protease. *bioRxiv* (2020). URL https://www.biorxiv.org/content/early/2020/07/24/2020.07.23.218784.

[90] Diamond Light Source, U. n. s. Main protease structure and xchem fragment screen (2020). URL https://www.diamond.ac.uk/covid-19/for-scientists/Main-protease-structure-and-XChem.html.

[91] Dubanevics, I. & McLeish, T. C. Computational analysis of dynamic allostery and control in the sars-cov-2 main protease. *bioRxiv* (2020). URL https://www.biorxiv.org/content/early/2020/07/20/2020.05.21.105965. https://www.biorxiv.org/content/early/2020/07/20/2020.05.21.105965.full.pdf.

[92] Zimmerman, M. I. *et al.* Sars-cov-2 simulations go exascale to capture spike opening and reveal cryptic pockets across the proteome. *bioRxiv* (2020).

[93] Owen, C. *et al.* Covid-19 main protease with unliganded active site. URL http://www.rcsb.org/structure/6YB7.

[94] Owen, C. *et al.* Rcsb pdb - 6y84: Sars-cov-2 main protease with unliganded active site (2019-ncov, coronavirus disease 2019, covid-19). URL https://www.rcsb.org/structure/6Y84.

[95] El-Gebali, S. *et al.* The pfam protein families database in 2019. *Nucleic acids research* **47**, D427–D432 (2019).

[96] ul Qamar, M. T., Alqahtani, S. M., Alamri, M. A. & Chen, L.-L. Structural basis of sars-cov-2 3clpro and anti-covid-19 drug discovery from medicinal plants. *Journal of pharmaceutical analysis* **10**, 313–319 (2020).

[97] Lindorff-Larsen, K., Piana, S., Dror, R. O. & Shaw, D. E. How fast-folding proteins fold. *Science* **334**, 517–520 (2011).

[98] Kuhlman, B. & Bradley, P. Advances in protein structure prediction and design. *Nature Reviews Molecular Cell Biology* **20**, 681–697 (2019).

[99] Onuchic, J. N. & Wolynes, P. G. Theory of protein folding. *Current opinion in structural biology* **14**, 70–75 (2004).

[100] Dill, K. A. Dominant forces in protein folding. *Biochemistry* **29**, 7133–7155 (1990).

[101] Leopold, P. E., Montal, M. & Onuchic, J. N. Protein folding funnels: a kinetic approach to the sequence-structure relationship. *Proceedings of the National Academy of Sciences* **89**, 8721–8725 (1992).

[102] Wolynes, P. G. Recent successes of the energy landscape theory of protein folding and function. *Quarterly reviews of biophysics* **38**, 405–410 (2005).

[103] Bryngelson, J. D. & Wolynes, P. G. Spin glasses and the statistical mechanics of protein folding. *Proceedings of the National Academy of Sciences* **84**, 7524–7528 (1987).

[104] Udgaonkar, J. B. Multiple routes and structural heterogeneity in protein folding. *Annu. Rev. Biophys.* **37**, 489–510 (2008).

[105] Chavez, L. L., Gosavi, S., Jennings, P. A. & Onuchic, J. N. Multiple routes lead to the native state in the energy landscape of the β-trefoil family. *Proceedings of the National Academy of Sciences* **103**, 10254–10258 (2006).

[106] Bonomi, M., Barducci, A., Gervasio, F. L. & Parrinello, M. Multiple routes and milestones in the folding of hiv–1 protease monomer. *PloS one* **5**, e13208 (2010).

[107] Onuchic, J. N., Socci, N. D., Luthey-Schulten, Z. & Wolynes, P. G. Protein folding funnels: the nature of the transition state ensemble. *Folding and Design* **1**, 441–450 (1996).

[108] Onuchic, J. N., Wolynes, P. G., Luthey-Schulten, Z. & Socci, N. D. Toward an outline of the topography of a realistic protein-folding funnel. *Proceedings of the National Academy of Sciences* **92**, 3626–3630 (1995).

[109] Kubelka, J., Chiu, T. K., Davies, D. R., Eaton, W. A. & Hofrichter, J. Sub-microsecond protein folding. *Journal of molecular biology* **359**, 546–553 (2006).

[110] Pedregosa, F. *et al.* Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* **12**, 2825–2830 (2011).

[111] Reiner, A., Henklein, P. & Kiefhaber, T. An unlocking/relocking barrier in conformational fluctuations of villin headpiece subdomain. *Proceedings of the National Academy of Sciences* **107**, 4955–4960 (2010).

[112] Piana, S., Lindorff-Larsen, K. & Shaw, D. E. Protein folding kinetics and thermodynamics from atomistic simulation. *Proceedings of the National Academy of Sciences* **109**, 17845–17850 (2012).

[113] Beauchamp, K. A., Ensign, D. L., Das, R. & Pande, V. S. Quantitative comparison of villin headpiece subdomain simulations and triplet–triplet energy transfer experiments. *Proceedings of the National Academy of Sciences* **108**, 12734–12739 (2011).

[114] Piana, S., Lindorff-Larsen, K. & Shaw, D. E. How robust are protein folding simulations with respect to force field parameterization? *Biophysical journal* **100**, L47–L49 (2011).

[115] Lindorff-Larsen, K. *et al.* Improved side-chain torsion potentials for the amber ff99sb protein force field. *Proteins: Structure, Function, and Bioinformatics* **78**, 1950–1958 (2010).

[116] Huang, P.-S., Boyken, S. E. & Baker, D. The coming of age of de novo protein design. *Nature* **537**, 320–327 (2016).

[117] Dahiyat, B. I. & Mayo, S. L. De novo protein design: fully automated sequence selection. *Science* **278**, 82–87 (1997).

[118] Shimaoka, M. *et al.* Computational design of an integrin i domain stabilized in the open high affinity conformation. *Nature structural biology* **7**, 674–678 (2000).

[119] Malakauskas, S. M. & Mayo, S. L. Design, structure and stability of a hyperthermophilic protein variant. *Nature structural biology* **5**, 470–475 (1998).

[120] Magliery, T. J. Protein stability: computation, sequence statistics, and new experimental methods. *Current opinion in structural biology* **33**, 161–168 (2015).

[121] Kries, H., Blomberg, R. & Hilvert, D. De novo enzymes by computational design. *Current opinion in chemical biology* **17**, 221–228 (2013).

[122] Garrabou, X., Wicky, B. I. & Hilvert, D. Fast knoevenagel condensations catalyzed by an artificial schiff-base-forming enzyme. *Journal of the American Chemical Society* **138**, 6972–6974 (2016).

[123] Koday, M. T. *et al.* A computationally designed hemagglutinin stem-binding protein provides in vivo protection from influenza independent of a host immune response. *PLoS pathogens* **12**, e1005409 (2016).

[124] Lin, Y.-R. *et al.* Control over overall shape and size in de novo designed proteins. *Proceedings of the National Academy of Sciences* **112**, E5478–E5485 (2015).

[125] Hsia, Y. *et al.* Design of a hyperstable 60-subunit protein icosahedron. *Nature* **535**, 136–139 (2016).

[126] Bhardwaj, G. *et al.* Accurate de novo design of hyperstable constrained peptides. *Nature* **538**, 329–335 (2016).

[127] Anfinsen, C. B., Haber, E., Sela, M. & White Jr, F. The kinetics of formation of native ribonuclease during oxidation of the reduced polypeptide chain. *Proceedings of the National Academy of Sciences of the United States of America* **47**, 1309 (1961).

[128] Kuhlman, B. & Baker, D. Native protein sequences are close to optimal for their structures. *Proceedings of the National Academy of Sciences* **97**, 10383–10388 (2000).

[129] Alford, R. F. *et al.* The rosetta all-atom energy function for macromolecular modeling and design. *Journal of chemical theory and computation* **13**, 3031–3048 (2017).

[130] Fleishman, S. J. *et al.* Computational design of proteins targeting the conserved stem region of influenza hemagglutinin. *Science* **332**, 816–821 (2011).

[131] Glasgow, A. A. *et al.* Computational design of a modular protein sense-response system. *Science* **366**, 1024–1028 (2019).

[132] Dantas, G., Kuhlman, B., Callender, D., Wong, M. & Baker, D. A large scale test of computational protein design: folding and stability of nine completely redesigned globular proteins. *Journal of molecular biology* **332**, 449–460 (2003).

[133] Saunders, C. T. & Baker, D. Recapitulation of protein family divergence using flexible backbone protein design. *Journal of molecular biology* **346**, 631–644 (2005).

[134] Musacchio, A., Noble, M., Pauptit, R., Wierenga, R. & Saraste, M. Crystal structure of a src-homology 3 (sh3) domain. *Nature* **359**, 851–855 (1992).

[135] Vijay-Kumar, S., Bugg, C. E. & Cook, W. J. Structure of ubiquitin refined at 1.8 åresolution. *Journal of molecular biology* **194**, 531–544 (1987).

[136] Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. Basic local alignment search tool. *Journal of molecular biology* **215**, 403–410 (1990).

[137] Eddy, S. R. *et al.* Multiple alignment using hidden markov models. In *Ismb*, vol. 3, 114–120 (1995).

[138] Khatib, F. *et al.* Algorithm discovery by protein folding game players. *Proceedings of the National Academy of Sciences* **108**, 18949–18953 (2011).

[139] Smith, C. A. & Kortemme, T. Backrub-like backbone simulation recapitulates natural protein conformational variability and improves mutant side-chain prediction. *Journal of molecular biology* **380**, 742–756 (2008).

[140] Bonet, J. *et al.* Rosetta funfoldes–a general framework for the computational design of functional proteins. *PLoS computational biology* **14**, e1006623 (2018).

[141] Mandelhttps://www.overleaf.com/project/5eda14e0fabcda00016e0ba9l, D. J. & Kortemme, T. Backbone flexibility in computational protein design. *Current opinion in biotechnology* **20**, 420–428 (2009).

[142] Dunbrack Jr, R. L. & Cohen, F. E. Bayesian statistical analysis of protein side-chain rotamer preferences. *Protein Science* **6**, 1661–1681 (1997).

[143] Lazaridis, T. & Karplus, M. Effective energy function for proteins in solution. *Proteins: Structure, Function, and Bioinformatics* **35**, 133–152 (1999).

[144] O'Meara, M. J. *et al.* Combined covalent-electrostatic model of hydrogen bonding improves structure prediction with rosetta. *Journal of chemical theory and computation* **11**, 609–622 (2015).

[145] Finkelstein, A. V., Badretdinov, A. Y. & Gutin, A. M. Why do protein architectures have boltzmann-like statistics? *Proteins: Structure, Function, and Bioinformatics* **23**, 142–150 (1995).

[146] Park, H. *et al.* Simultaneous optimization of biomolecular energy functions on features from small molecules and macromolecules. *Journal of chemical theory and computation* **12**, 6201–6212 (2016).

[147] Dayhoff, M. O. *Atlas of protein sequence and structure* (National Biomedical Research Foundation., 1972).

[148] Henikoff, S. & Henikoff, J. G. Automated assembly of protein blocks for database searching. *Nucleic acids research* **19**, 6565–6572 (1991).

[149] Eddy, S. R. Profile hidden markov models. *Bioinformatics (Oxford, England)* **14**, 755–763 (1998).

[150] Needleman, S. B. & Wunsch, C. D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology* **48**, 443–453 (1970).

[151] Pearson, W. R. Searching protein sequence libraries: comparison of the sensitivity and selectivity of the smith-waterman and fasta algorithms. *Genomics* **11**, 635–650 (1991).

[152] Durbin, R., Eddy, S. R., Krogh, A. & Mitchison, G. *Biological sequence analysis: probabilistic models of proteins and nucleic acids* (Cambridge university press, 1998).

[153] Karlin, S. & Altschul, S. F. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proceedings of the National Academy of Sciences* **87**, 2264–2268 (1990).

[154] Thompson, J. D., Gibson, T. J. & Higgins, D. G. Multiple sequence alignment using clustalw and clustalx. *Current protocols in bioinformatics* 2–3 (2003).

[155] Altschul, S. F. *et al.* Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic acids research* **25**, 3389–3402 (1997).

[156] Rabiner, L. R. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE* **77**, 257–286 (1989).

[157] Blunsom, P. Hidden markov models. *Lecture notes, August* **15**, 48 (2004).

[158] Krogh, A., Brown, M., Mian, I. S., Sjolander, K. & Haussler, D. Hidden markov models in computational biology. applications to protein modeling. *Journal of molecular biology* **235**, 1501–1531 (1994).

[159] Boeckmann, B. *et al.* The swiss-prot protein knowledgebase and its supplement trembl in 2003. *Nucleic acids research* **31**, 365–370 (2003).

[160] Fleishman, S. J. *et al.* Rosettascripts: a scripting language interface to the rosetta macromolecular modeling suite. *PloS one* **6**, e20161 (2011).

[161] Goldenzweig, A. *et al.* Automated structure-and sequence-based design of proteins for high bacterial expression and stability. *Molecular cell* **63**, 337–346 (2016).

[162] Khersonsky, O. *et al.* Automated design of efficient and functionally diverse enzyme repertoires. *Molecular cell* **72**, 178–186 (2018).

[163] Sheffler, W. & Baker, D. Rosettaholes: rapid assessment of protein core packing for structure prediction, refinement, design, and validation. *Protein Science* **18**, 229–239 (2009).

[164] Rosetta packstat filter (2020). URL https://www.rosettacommons.org/docs/latest/scripting_documentation/RosettaScripts/Filters/filter_pages/PackStatFilter.

[165] Buchan, D. W. & Jones, D. T. The psipred protein analysis workbench: 20 years on. *Nucleic acids research* **47**, W402–W407 (2019).

[166] Kabsch, W. & Sander, C. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers: Original Research on Biomolecules* **22**, 2577–2637 (1983).

[167] Touw, W. G. *et al.* A series of pdb-related databanks for everyday needs. *Nucleic acids research* **43**, D364–D368 (2015).

[168] Allegra, M., Facco, E., Laio, A. & Mira, A. Clustering by the local intrinsic dimension: the hidden structure of real-world data. *arXiv preprint arXiv:1902.10459* (2019).

[169] Jong, K. & Hassanali, A. A. A data science approach to understanding water networks around biomolecules: the case of tri-alanine in liquid water. *The Journal of Physical Chemistry B* **122**, 7895–7906 (2018).

[170] Pinamonti, G., Paul, F., Noé, F., Rodriguez, A. & Bussi, G. The mechanism of rna base fraying: Molecular dynamics simulations analyzed with core-set markov state models. *The Journal of chemical physics* **150**, 154123 (2019).

[171] Ansari, N., Laio, A. & Hassanali, A. Spontaneously forming dendritic voids in liquid water can host small polymers. *The journal of physical chemistry letters* **10**, 5585–5591 (2019).