

University of Warsaw
Faculty of Physics

Maciej Dziubiński

**Development of selected mesoscopic
physical models with the aid of machine
learning methods and their applications
in studies of molecular systems**

DOCTORAL THESIS
in
BIOPHYSICS
in the
GRADUATE DIVISION
of the
FACULTY OF PHYSICS
at the
UNIVERSITY OF WARSAW

Promoter
Prof. Bogdan Lesyng
Department of Biophysics, Faculty of Physics,
University of Warsaw

Auxiliary Promoter
Dr. Paweł Daniluk
Department of Biophysics, Faculty of Physics,
University of Warsaw

Warsaw, October 2016

Acknowledgements

This work would not be possible without the patience, encouragement, and supervision of Bogdan Lesyng and Paweł Daniluk. I am grateful for their influence and the chance of working together.

Prof. Lesyng's research group provided a creative vibe that undoubtedly contributed to the following dissertation. I am thankful for the stimulating conversations, the friendly atmosphere, and the feeling of being a part of it all.

I would also like to thank my friends and fellow graduate students: Mateusz Iskrzyński for his eccentricity, Aleksandra Fijałkowska for her perspective, Michał Pecelerowicz for his scepticism, Małgorzata Prokopowicz for her determination, and Marcin Sobieraj for his optimism.

Last but not least, I would like to thank my wife, Marta, for her ceaseless support and love.

Abstract

This dissertation is concerned with the development and application of unsupervised machine learning methods in the field of theoretical biophysics and bioinformatics. The machine learning approach offers a powerful framework for extracting and purifying valuable information from large, multi-dimensional sets of data generated in simulations and experiments of biomolecular systems. It is not, however, the case that ready-made machine learning methods offer infallible means of dealing with all sorts of complex, and partially chaotic data encountered in computational biophysics and structural biology. Large portion of this work is devoted to the adaptation of unsupervised machine learning techniques to our particular purposes.

In this dissertation, we employed unsupervised machine learning strategies dealing with two problems arising in theoretical biophysics and bioinformatics. The first problem was the identification of quasi-rigid structural parts in proteins, whereas the second one was devoted to discovery of internal cooperation of molecular subsystems that propels a conformational transition. Both problems involved dynamical properties of molecular systems, and the analyses presented in this dissertation allowed for a simplified description of these phenomena.

We demonstrate how the unsupervised machine learning approach can help in explaining intricacies hidden within seemingly chaotic molecular dynamics simulation data. The methods developed in this thesis increase our ability to understand complex molecular phenomena. But we also point out potential problems associated with applying unsupervised machine learning algorithms in the field of molecular biophysics.

Contents

1	Introduction	5
1.1	Machine learning	5
1.1.1	Supervised machine learning	6
1.1.2	Unsupervised machine learning	7
1.1.3	Adjacency matrix representation and spectral clustering	9
1.2	Molecular dynamics	10
1.2.1	Sampling techniques and the potential of mean force	11
1.2.2	Unsupervised machine learning in molecular data analysis	12
1.3	The overall aim of this dissertation	13
2	ResiCon: a method for the identification of dynamic domains, hinges and interfacial regions in proteins	15
2.1	Introduction	16
2.2	Approach	17
2.3	Methods	17
2.4	Results and discussion	25
2.5	Summary of results	33
3	Towards the identification of molecular cogs	37
3.1	Introduction	37
3.2	Methodology	38
3.2.1	Decomposition of the energetic contribution	40
3.2.2	Clustering	42
3.2.3	Trapezoidal rule for integrating A	46
3.3	Results	47
3.3.1	The $[\text{NH}_3^+]\text{CC}(\text{I})\text{I}$ molecular model	48
3.3.2	Overview	49
3.3.3	Results for the $[\text{NH}_3^+]\text{CC}(\text{I})\text{I}$ molecular model	49
3.3.4	Results for related molecules	55
3.4	Discussion	56
3.5	Summary of results	57

4	Conclusions	61
4.1	Dynamic domains	61
4.2	Molecular cogs	62
4.3	Outcome	63
4.4	Final comments	64
	Appendices	65
A	Supplementary Materials for Chapter 2	67
A.1	Hierarchical clustering	67
A.2	Quality analysis for PiSQRD	69
A.3	Procedure for selecting representative configurations from a trajectory	71
A.4	Complete set of results	73
B	Supplementary Materials for Chapter 3	87
B.1	Constrained molecular dynamics	87
B.2	Error estimation using a bootstrapping procedure	88
B.3	Merging two clusterings from the AP method	89
B.4	Parameters of the genetic clustering algorithm	91
B.5	Complete results for the CCC(I)I, NCC(I)I, CC1CC(I)I molecules . .	91

Chapter 1

Introduction

The primary aim of this work was the development of methods for extracting relevant information from multi-dimensional data describing structural and dynamical properties of biomolecular systems. We adapted advanced approaches from the general-purpose field of *machine learning*, and in particular – from the *unsupervised* machine learning methodology. In Chapter 2, we present our recently published method of discovering quasi-rigid parts in proteins, that can be used to better interpret experimental as well as simulation data. Chapter 3 presents another problem of identifying parts of a molecular system that propel (or hinder) a given structural transition. In both these cases we applied *clustering* algorithms, commonly used for finding patterns in unstructured – and seemingly chaotic – data.

In this introductory chapter, we give an overview of the machine learning approach and the molecular dynamics simulation scheme. These comprehensive overviews were intended to sketch a map of the current state of knowledge, and mark the research frontier engaged in this work. Specific clustering algorithms and molecular dynamics techniques applied in this study are discussed in more details in Chapters 2 and 3.

1.1 Machine learning

Machine learning is a branch of computer science, concerned with algorithms that “learn” how to extract pertinent information from noisy, complex datasets. The three main branches of machine learning are: *supervised*, *unsupervised* and *reinforcement* learning. The first one is concerned with assigning a class (or: label) to a given observation, assuming that an independent and correctly labeled dataset is known beforehand. The unsupervised learning deals with a similar task, except that there is neither prior nor posterior knowledge about which observations should be assigned to which class, nor is it clear what is the “right” number of such classes. The reinforcement learning regime, often identified with artificial intelligence, deals with the problem of taking *actions* under certain *conditions* (or: in a given *environment*), so as to maximize the cumulative *reward*. Most notably, reinforcement learning algorithms are used in games (such as chess), but despite their tremendous

impact on present and – probably – future research¹, their discussion is beyond the scope of this dissertation, and we shall not present them in more detail.

1.1.1 Supervised machine learning

The data may have an underlying – but difficult to define – structure, so that each observation comprising the dataset belongs to a particular class². We say that the data are *labeled*, i.e. each observation has a label assigning it to one of m classes. In other words, the dataset is composed of N observations, $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where each observation is a pair with: a vector \mathbf{x}_i describing the i^{th} instance, and a corresponding label y_i . The vector \mathbf{x}_i contains features, each with its own domain (discrete, continuous, or other). The label y_i , on the other hand, takes on one of m values, corresponding to one of m pre-defined classes. Supervised machine learning algorithms try to capture correlation, mutual information, or any other types of relations between the features and the labels y , to produce a prediction for any new, previously unseen, observation.

Because the primary aim of supervised learning is the assignment of observations to classes, it is also known as the *classification problem*, and particular algorithms are referred to as *classifiers*. One of the first classifying algorithms was the *k nearest neighbors* (k -NN) method, proposed as early as 1951 [29]. In k -NN we assume the observations are points embedded in a metric space, so that their mutual distance can be readily calculated. For a given k , and a set of points with known labels, we assign a new observation to the class, which is most prevalent among its k nearest neighbors. In the case of a 2-class problem, it is advised to choose an odd number of neighbors to avoid ties. However, k -NN suffers from noisy data (uninformative coordinates in the vectors \mathbf{x} describing the observations), and from the “curse of dimensionality” (the problem of finding truly close neighbors in a high-dimensional space).

A great number of classification methods have been proposed in lieu of k -NN; see [46] for a more detailed discussion. However, it is noteworthy that alongside new supervised learning algorithms, auxiliary meta-methods have been developed, which combine several classifiers and greatly surpass their individual predictive power. Most notable are the *bagging* and *boosting* schemes. In fact, one of the most powerful classification methods to date is the *eXtreme Gradient Boosting* (XGBoost) algorithm [14], that combines boosting and *random forests* [45].

Practitioners of supervised machine learning can test their skills by participating in contests held at www.kaggle.com. Most of these contests offer prizes (most famous is the \$1 million Netflix competition), but more importantly, the discussion

¹See, for example, Google’s Gorila (General Reinforcement Learning Architecture) project [57], or Skynet from the *Terminator* franchise.

²Typically, *regression* is also considered a part of the supervised learning scheme [46]. In principle, the regression problem is broader than classification – instead of discrete classes, observations are ascribed real values. Thus, regression deals with the task of approximating the function f that for each observation \mathbf{x} assigns its corresponding value, $f(\mathbf{x})$. However, for the purposes of this short review we shall focus only on classification.

at www.kaggle.com forums is the most current source of information about the trends, novelties and successes of cutting-edge machine learning methods.

1.1.2 Unsupervised machine learning

Unsupervised machine learning is a class of algorithms for extracting information from unstructured datasets. Fundamentally, these methods address the following questions³:

- Is there a clear way of visualizing a complex set of data?
- Can we partition a dataset into cohesive subgroups, distinctive from each other?
- What statistically significant patterns (e.g. correlations, rules, causality relationships) are present in the dataset?

As we explain further, the data considered in this dissertation came from numerical simulations of molecular models. However, unsupervised machine learning is commonly used for analyses of all sorts of objects: websites, genomic data, points on a map, etc. But regardless of the source of these data, we are faced with the problem of purifying the information hidden within it. Unsupervised machine learning offers two main methods of simplifying the data: *dimensionality reduction* (predominantly: principal component analysis), and *clustering* (for identifying distinctive subgroups of objects).

Principal component analysis

The *principal component analysis* (PCA) is a transformation that, for a given set of observation points with correlated coordinates, yields a set of points with coordinates that are no longer correlated. Typically, those uncorrelated observations are represented using much fewer coordinates, therefore PCA is often regarded as a dimensionality reduction procedure. As such, PCA is often employed in the exploratory phase of an analysis.

However, exploratory data analysis is not the only application for PCA; it is commonly used in reducing noise in observations in supervised learning methods, thus enhancing their predictive power⁴. In Chapter 2, we facilitate PCA to select representative structures from a large set of configurations of the HIV-1 protease. These representatives are then used as input for our cluster analysis, leading to quasi-rigid structural parts of a protein.

³These are not the only questions raised by unsupervised machine learning methods. Most notably, one of the more interesting, and fast-changing studies is the *community detection*, which is a more general problem than clustering. But as machine learning becomes more and more popular, new problems and applications arise, and it is difficult to draw a clear line that would encompass the whole field of unsupervised machine learning.

⁴It should also be noted, that PCA is widely applied in the field of computational biophysics, and was used for analyzing molecular dynamics trajectories [3], as well as for estimating free energy differences (see for example [2]).

The PCA is not without its limitations. At its core, PCA relies on covariance as a measure of relatedness between coordinates of the vectors representing observations in the dataset. However, null covariance of two random variables X and Y does not imply that they are independent⁵. Thus, whenever we are unwilling to assume that covariance is a reliable measure of dependence, we should use more advanced techniques (e.g., one of many non-linear PCA variants [34], or the immensely popular t -SNE algorithm [53]).

Overview of popular clustering methods

One of the first clustering methods was the k -means algorithm [54] (k indicating the number of clusters), in which points are assigned to the cluster with the closest mean. Finding an exact solution to the k -means problem is computationally intractable, therefore most implementations carry out a greedy, iterative procedure that converges to a local minimum. Such an approximate solution is, in most cases, satisfactory, however the main disadvantage of the k -means algorithm reveals itself whenever clusters have a stretched out shape. In such cases, the k -means disregards an oblong shape of a grouping, because it does not account for gaps between clusters, neither for their “connectivity”.

Another popular clustering algorithm, DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) [27], alleviates k -means’ problems by identifying clusters as continuous, dense “clouds” of points. As a result, the clusters are allowed to have even complex shapes, as long as they retain their connected structure. However, although DBSCAN is capable of recognizing oblong clusters, it often fails to identify boundaries separating them. That is, DBSCAN has difficulties in discerning weak connections between clusters from strong, frequent ones inside them. As a result, DBSCAN is a poor choice when dealing with sets of points with many weak inter-cluster relations.

DBSCAN requires a metric or a similarity function, that expresses relatedness of two observations. This is a common scenario, in which we are more focused on the relations between objects, rather than how to describe them using a multi-dimensional vector. It makes for a convenient setup as it puts an emphasis on the meaning of clusters (assuming we chose a meaningful measure of relatedness), instead of individual objects.

Alternative to DBSCAN, and a fairly natural approach to clustering, is a method called *agglomerative hierarchical* clustering. Like in DBSCAN, we need to specify a metric (or a similarity measure) that quantifies relatedness of pairs of objects in a dataset. Agglomerative hierarchical clustering is a greedy, iterative procedure, that builds clusters by merging groups of points that are closest. This closeness may be expressed in terms of minimal distance between any two points from the two groups, in terms of their mean distance, or in a number of different ways. The procedure stops, when the minimal closeness of clusters exceeds a

⁵In particular, Y and $X = Y^2$ are uncorrelated if (for example) Y is normally distributed with a zero mean, yet they are clearly dependent.

pre-defined threshold, called the *height parameter*⁶. One of the drawbacks of the hierarchical clustering scheme is that it is difficult to choose an optimal value for the height parameter. Additional criteria may be utilized for this purpose, but even then the “best” value of the parameter may be volatile and even a slight change in its value yields a completely different result.

Many other effective clustering algorithms have been proposed to answer the needs arising in various fields of application. Currently, the most valued clustering algorithms are those that prove to be useful in a whole variety of problems. That is, methods that work well regardless of the shape of the clusters, the exact nature of the similarity measure, or small changes in the input parameters.

Number of clusters

Most clustering algorithms require as input a parameter k , indicating the number of clusters. Alternatively, some other parameter may be required that ultimately determines the number of clusters (for example, the height parameter in hierarchical clustering – see the discussion in Chapter 2).

The k parameter is crucial, as it pre-determines the result of any clustering algorithm. Assuming we chose a clustering algorithm, a typical strategy of finding the optimal number of clusters is to carry out the procedure for $k = 2, \dots, K$. Then, for each clustering we evaluate the result using an external quality measure (see [80] for a comprehensive review). However, such an approach only adds to the overall complexity of the clustering problem, because different external quality measures are suitable for different applications.

A more preferable solution would be an *internal* criterion, built into the clustering algorithm. This is one of the reasons why in Chapter 2 we used a particular spectral clustering method, that offers just that.

1.1.3 Adjacency matrix representation and spectral clustering

It is important to introduce at this point the matrix representation of a set of objects and their similarity. If we enumerate the objects in our set using natural numbers $1, 2, 3, \dots, N$, and denote the similarity between objects i and j by w_{ij} , we can represent the relations between the objects using a matrix $W \in \mathbb{R}^{N \times N}$. Matrix W is referred to as the *similarity* (or: *adjacency*) *matrix*, and we refer to it repeatedly throughout this dissertation.

The clustering method used in Chapter 2 finds an optimal subdivision of the dataset by examining the eigenvectors and eigenvalues of the similarity matrix. As we explain in Chapter 2, it seems plausible that the eigenvectors of the similarity matrix somehow summarize the structure of the data. But – perhaps more surprisingly – the eigenvalues contain information about the “distortion” of the clustering, and can thus help in choosing the optimal number of clusters. Therefore,

⁶Its name is associated with the fact that mergers carried out by the hierarchical procedure may be represented as an acyclic connected graph, also called a *tree*. The point at which we stop the mergers is related to the height of that tree, hence the name of the parameter.

the spectral clustering algorithm discussed in Chapter 2 not only provided us with high-quality partitions, but also suggested the *right* number of clusters.

Unsupervised machine learning in this dissertation

In Chapter 2, we propose and utilize a similarity function expressing geometrical variability of pairs of amino acids. As a result, the clustering indicates structural parts of a protein that are internally rigid, i.e. composed of amino acids sharing low geometrical variability. In Chapter 3, the similarity is defined for pairs of atoms, and expresses their contribution to the free energy change associated with a given structural transition. As a result, the clustering yields two parts of a molecule that cooperate in pushing the transformation forward, or pulling it backwards. We call these parts *molecular cogs*, to draw an analogy to the popular view that large biomolecules can often be thought of as “molecular machines”.

1.2 Molecular dynamics

Molecular dynamics (MD) comprise a wide range of numerical methods designed for the study of motions and properties of molecular systems. Typically, the MD approach assumes a classical potential energy function, approximated by an analytical function U , with force field parameters, which treats atoms as electrostatically charged points, and chemical bonds as springs and hinges. The arguments of the U function are atomic configurations \mathbf{q} (we use boldface to denote vectors) of the molecular system, with the potential energy values, $U(\mathbf{q})$. If we denote the set of all configurations of a given system by Ω , then $U: \Omega \rightarrow \mathbb{R}$. The force acting on a particular atom i is then derived from U by taking the gradient with respect to position \mathbf{q}_i of that atom:

$$\mathbf{F}_i(\mathbf{q}) = -\nabla_i U(\mathbf{q}),$$

which requires of U to be differentiable.

More accurate algorithms take into account quantum effects, making the simulation more reliable, but also immensely expensive in terms of computational time. However, although modeling intricate properties of molecular systems using a classical potential may seem an over-simplification, MD has achieved considerable success over the recent years, and is constantly improving. Classical MD simulations are popular mainly because of their significantly lower computational cost in comparison with their quantum-mechanical counterparts. This, in particular, allows for estimation of free energy profiles of structural transitions, even for large, biologically-relevant systems (such as enzymes, DNA, membrane systems, and others). MD can be therefore thought of as a testing ground for advanced sampling techniques, but also pattern-recognition techniques such as the aforementioned machine learning.

The free energy translates into experimentally observable properties. For example, the affinity of a ligand and an enzyme, or the chance of transferring an ion through a membrane channel. Free energy profiles offer insight into the mechanics

of such processes, but also allow for utilizing feedback from experimental data to refine the modeling potentials.

To run MD simulations, one needs to choose the right force field, but also – and perhaps more importantly – the appropriate *sampling technique*. The underlying premise is that molecular systems in thermal equilibrium experience random impulses, which can be modeled *via* collisions with molecules of the surrounding world. The total energy of a system in thermal equilibrium is, therefore, *not* constant, but rather fluctuates around a certain mean value.

Consequently, configurations of a system are characterized by a probability density, which in the case of thermal equilibrium ($T = \text{const.}$) is the well-known *Boltzmann distribution*:

$$\rho_B(\mathbf{q}) = Z^{-1} e^{-U(\mathbf{q})/k_B T}, \quad (1.1)$$

where k_B is the Boltzmann constant, and Z is a normalizing factor of great importance, often referred to as the *partition function*, given by:

$$Z = \int_{\Omega} e^{-U(\mathbf{q})/k_B T} d\mathbf{q}.$$

From this probabilistic perspective we can view an MD simulation as a sampling procedure, during which we *should* acquire configurations according to the Boltzmann distribution (assuming $T = \text{const.}$). Macroscopic properties of the system (affinity of a ligand to an enzyme, for example) are defined as *expected values* of appropriate microscopic quantities, and are estimated by averages over configurations sampled in an MD simulation. In the constant temperature regime, MD (but also Monte Carlo) simulation data can be used to estimate internal energy, E , entropy, S , and the free energy, A , of a system using basic formulae:

$$E = \langle U \rangle,$$

$$S = -k_B \langle \log \rho_B \rangle,$$

and

$$A = E - TS,$$

respectively. The $\langle \cdot \rangle$ denotes either the time-average (MD case) or ensemble-average (Monte Carlo case). It is of paramount importance that the sample is generated correctly, i.e. so that the *whole* configurational space is adequately scanned, and that the estimates do not suffer from unexpectably high statistical errors.

1.2.1 Sampling techniques and the potential of mean force

Underlying the simplistic, classical assumptions embedded in MD simulations are complex sampling techniques used for estimating macroscopic properties of microscopic systems. MD simulations can, for example, mimic time evolution of a system with the use of integration schemes such as the *Langevin dynamics*, which numerically integrates the following stochastic differential equation:

$$\mathbb{M}\ddot{\mathbf{q}} = -\nabla U(\mathbf{q}) - \gamma \mathbb{M}\dot{\mathbf{q}} + \sqrt{2\gamma k_B T \mathbb{M}} \mathbf{R}(t)$$

where \mathbb{M} is a diagonal matrix of atom masses, and $\mathbf{R}(t)$ is a stationary Gaussian process with zero mean, and with white noise auto-correlation expressed by the Dirac delta function: $\langle \mathbf{R}(t)\mathbf{R}(t') \rangle = \delta(t - t')$. The output of a Langevin dynamics simulation is a trajectory, that approximates the time-evolution of a system.

Potential of mean force

Vast samples of multidimensional points produced in the course of an MD simulation are impossible to interpret, unless we utilize some simplifying technique. Probably the most popular representation of a change or transformation occurring in a system is the so-called *collective variable*, and the accompanying concept of the *potential of mean force* (PMF). A collective variable ξ is a function of the system's configuration \mathbf{q} , defined so that its change can be readily translated into the progress of a particular transformation. The PMF for a particular value ξ^* of that collective variable, is defined as follows:

$$A(\xi^*) := -k_B T \log \rho(\xi^*),$$

where ρ is a probability density of the configuration \mathbf{q} , such that $\xi(\mathbf{q}) = \xi^*$.

While the collective variable ξ serves as a method of reducing the dimensionality of a complex structural transition, the PMF provides a probabilistic interpretation of such process. Through the PMF we gain insight into the transformation's bottlenecks and into the ranges of values of the collective variable corresponding to the system's meta-stable states. In Chapter 3, we describe our method of discovering the so-called *molecular cogs*, which was possible by facilitating a particular sampling technique aimed at extracting the PMF.

1.2.2 Unsupervised machine learning in molecular data analysis

Data produced by MD simulations of biomolecules are multi-dimensional trajectories, points $\{\mathbf{q}_t\}_{t=1}^M$, where $\mathbf{q}_t \in \mathbb{R}^{3n}$ (M is the number of steps of the simulation, and n – the number of atoms of the system). Depending on the sampling technique adapted in the MD simulation, the t parameter may or may not be associated with time, as, for example, in the case of Monte Carlo sampling. Analogously to MD, configurations acquired from NMR experiments can be also thought of as a set of multi-dimensional points, $\mathbf{q}_t \in \mathbb{R}^{3n}$, in which the order imposed by the t parameter has very little to do with time.

The methods presented in Chapters 2 and 3 of this dissertation exploit unsupervised machine learning to infer properties and mechanics of molecular systems from noisy, multi-dimensional data. As we try to explain throughout this dissertation, this type of analysis allows for better understanding of molecular systems, and gives an incentive for posing new hypotheses regarding the mechanism underlying their function.

Dynamic domains as clusters

Small, globular proteins are often thought of as fairly rigid biomolecules. However, as the number of atoms in a system increases, the set of low-energy, accessible configurations grows rapidly. Consequently, large molecular systems can undergo significant structural changes, involving collective, multi-step transitions. Due to immense complexity of these systems, such phenomena are difficult to describe, let alone to interpret.

In Chapter 2, we introduce a method of simplifying these incomprehensible transitions using a clustering scheme. The main idea of our method is to find parts of the protein which – although moving with respect to each other – remain internally rigid. Or, more accurately, *quasi-rigid*, because small variations in distances between amino acids are omnipresent. In the literature, these quasi-rigid parts of the protein are referred to as *dynamic domains* [5, 42, 44].

In the dynamic domains identification we adapted a clustering framework. We proposed a measure of strength of a contact between amino acids, and applied an appropriate clustering algorithm to identify quasi-rigid parts of a protein. The strength of a contact depends on structural variability of a given pair of residues. Having a set of objects (amino acids) and a similarity measure (contact strength between residues), we constructed an adjacency matrix. This matrix was used as input for a suitable clustering procedure, one of the class of *spectral algorithms*, which is focused on finding eigenvectors of a matrix, closely related to the adjacency matrix for a given protein.

Molecular cogs as clusters

In Chapter 3, we present our newly-developed method of identifying *molecular cogs* – subsystems of a molecule which, for a given structural transformation, drive the transition forwards or backwards. This is another application of the cluster analysis, although very different from the one discussed in Chapter 2.

For the purpose of identifying molecular cogs we proposed a similarity function between pairs of atoms, which quantifies the contributions of that pair the free energy change along a collective variable. With that, we again construct adjacency matrices, and discover clusters which correspond to groups of atoms sharing a common contribution to the free energy (positive or negative). It should be noted, however, that the method of dynamic domains and molecular cogs identification differ not only at the stage of adjacency matrix construction, but also in the choice of the clustering algorithm.

1.3 The overall aim of this dissertation

Both, simulation and experimentally acquired data become increasingly complex, in particular because:

- the systems under study are progressively larger,

- and their structural transitions comprise seemingly chaotic paths in multi-dimensional configurational spaces.

For both these reasons, we need to facilitate methods of extracting the underlying structure of the data, to gain perspective, and to augment our 3-dimensional perception in order to pose new hypotheses about structural and functional properties of the biomolecular world.

The overall aim of this dissertation was the simplification of complex, dynamical properties of molecular systems. We applied unsupervised machine learning techniques to two such problems. One problem arising in computational biophysics is the identification of dynamic domains, where the input consist of configurations produced in the course of an MD simulation, or an NMR experiment. Although highly mobile and flexible, many proteins have structurally static sub-regions. By expressing protein's conformations in terms of quasi-rigid parts that move with respect to one another, but remain internally rigid, we were able to simplify the description of its dynamic nature.

The second problem tackled in this dissertation is the identification of molecular subsystems that play active roles in structural transformations of the whole systems. The focus of this study was more technical, as the molecules analyzed therein were small. However, we were also aiming at discovering conceptual and methodological caveats that might hinder analyses of larger molecules such as proteins.

In the following chapters we describe how we construct the adjacency matrices used for clustering, how we choose an optimal number of clusters, and how we validated information obtained from our methodology.

Chapter 2

ResiCon: a method for the identification of dynamic domains, hinges and interfacial regions in proteins

The text of Chapter 2 is a verbatim citation of the work published in the Oxford Journal *BIONFORMATICS* [24]. The co-authors, Paweł Daniluk and Bogdan Lesyng, came up with the idea of using contacts as a means of assessing structural variability and provided expertise regarding analysis of protein structures and clustering strategies.

Motivation and set-up

In the introductory chapter, we mentioned that the interpretation of a clustering procedure depends on the measure used to express pairwise similarity between objects in a dataset. Likewise, the *quality* of the clustering depends on how cleverly this measure is chosen. The similarity measure can be conjured up from experience, intuition or any other premise, but it stands to reason that only those measures that successfully capture pertinent information are likely to yield valuable insight.

The method of the dynamic domains detection described in the following chapter makes use of a structure-based similarity measure between two amino-acid residues. On the one hand, such a measure is fairly simple, which makes the interpretation of the clustering much easier. On the other hand, the measure was expected to capture essential interactions between residues, as pairs of amino-acids that remain close together are more likely to attract, rather than repel one another.

The results of our method (ResiCon) were compared to those of two other methods in the field. The first one (GeoStaS) is purely structure-based, however the formulation of the similarity measure used therein lacks any physical motivation. The second (PiSQRD) uses a measure that was aimed at describing physical interactions between amino-acid residues. Results presented in this chapter show

that ResiCon found a middle ground between these two methods, yielding more compact and rigid dynamic domains than either GeoStaS or PiSQRD.

2.1 Introduction

Proteins are not static. Nuclear magnetic resonance (NMR) spectroscopy [55] and the spin-echo spectroscopy [11] experiments show that. In several cases it was proven that flexibility may be crucial to protein functionality [28, 40]. Although experimental methods provide only general clues about intramolecular motions, molecular dynamics (MD) simulations extend their reach by giving a higher resolution picture – both in space and time – of protein mobility. By studying an NMR ensemble or MD trajectory one may notice that it is composed of relatively rigid structural parts, often referred to as *dynamic domains* [43].

Domains in traditional sense are regarded as parts of the protein which are: conserved (in terms of evolution), autonomous (in terms of folding), and/or compact (in terms of tertiary structure). Such “static” domains are identified through sequence homology, structural analysis of a single configuration, or both. (For conventional methods of identifying protein domains based on multiple sequences or a single structure see e.g. [63, 9].) Conversely, dynamic domains depend on structural transitions performed by the protein.

A number of methods for identification of dynamic domains have been developed. The simplest procedures are based on normal mode analysis, which assumes a harmonic approximation of the potential energy function [44]. More advanced approaches use the Gaussian Network Model and analyze correlations in motions between the residues of the protein [82]. Many other approaches have also been developed [5, 79, 7, 32, 61], but all of them anticipate dynamic domains by analyzing a single structure of a protein. Another class of methods for dynamic domains assignment requires exactly two configurations (see [42, 52, 81]). However, the assumption that two “representative” structures encompass all relevant motions is rather speculative.

Experimental and *in silico* methods reach beyond single-structure representation, and are capable of producing numerous configurations of a given protein. Rather than inferring dynamic domains from one or two structures, a more natural approach would be to determine them based on an ensemble of configurations. GeoStaS is the only method known to us that analyzes a whole ensemble of configurations and assigns each residue to a dynamic domain [64]. Although GeoStaS can analyze not only proteins but also nucleic acids, it fails to discover dynamic domains whenever they rotate with respect to each other. Alternative methods of analyzing ensembles of configurations assign residues to a static “core” or unstructured bundle (see [70, 48]).

The purpose of this study was to develop a novel methodology named ResiCon, capable of extracting dynamic domains from an ensemble of protein’s configurations. ResiCon analyzes strengths of contacts between residues based exclusively on geometrical changes occurring in the provided set of structures. The set may

be an NMR ensemble of configurations, or snapshots produced in the course of an MD simulation. ResiCon’s main functionality is to identify dynamic domains, but it can also find hinges and interfacial (interdomain) regions.

2.2 Approach

ResiCon starts with identifying pairs of residues which are *in contact*. There are several definitions of contacts between amino-acid residues in the literature. We used the definition presented in [20] and adapted it to the case when more than one structure is given (see also [21]).

Next, ResiCon constructs a virtual scaffold, connecting residues which are in contact with bars. Stiffness of a given bar reflects the estimated strength of the corresponding contact.

Finally, to identify dynamic domains, ResiCon carries out a partitioning (by computing minimal cuts) of the scaffold, cutting weaker and preserving stiffer bars. This partitioning is carried out by applying a spectral clustering algorithm presented in the following section.

The fundamental underlying assumption is that stability of rigid parts results from stable interactions between its residues. However, in our approach we do not analyze physical interactions between residues – they may be hydrophobic, electrostatic or significant in some other way. We simply assume that the measure of strength of a contact between residues is reflected by their geometrical variability across a given sample of structures.

2.3 Methods

Throughout this paper we use terms: *model*, *configuration*, *structure* and *conformation* interchangeably. We will refer to a set of structures acquired from an MD trajectory or NMR experiment as the *ensemble of configurations* or simply: *an ensemble*. Let us denote the number of structures in an ensemble by S .

Residue contact

For each pair of residues in every model we compute distances between C_α atoms (d_α) and between geometrical centers of side-chains R_C (d_C) (for glycine $R_C = C_\alpha$, and for alanine $R_C = C_\beta$). We say that two residues are *in contact*, if at least one configuration in the ensemble satisfies the condition:

$$(d_\alpha \leq 6.5\text{\AA}) \quad \text{or} \quad (d_C \leq 8\text{\AA} \quad \text{and} \quad d_\alpha - d_C \geq 0.75\text{\AA}) \quad (2.1)$$

Threshold values are the same as in definition of contact presented in [20] and relate to the range of distances in which physical interactions between residues occur. The second sub-condition favours residues whose side-chains point towards each other (see Figure 2.1). Residues that are sequential neighbors are not taken into account.

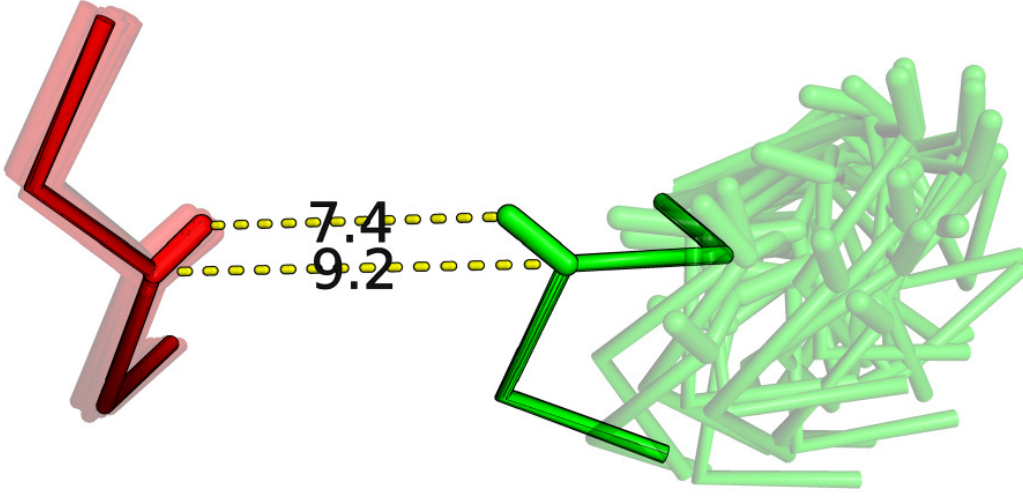


Figure 2.1: For each configuration in an ensemble, pairs of elements are constructed. In this picture residues are in contact because there exists a configuration for which the condition (2.1) is satisfied (the distance $d_C = 7.4\text{\AA}$, and $d_\alpha - d_C = 1.8\text{\AA}$).

We assign a quantitative value to the strength of a contact in terms of geometrical variability of the structural part associated with that contact. Such measure is required to capture not only changes in d_α , but also rotational shifts and alterations in the backbone in the vicinity of both residues that are in contact. To do so, we constructed structural parts, comprising sequential neighbors of the two residues in contact. ResiCon assigns a numerical value to the geometric variability by using the least root mean square deviation (RMSD) ([47]). Before elaborating on the details, we proceed with the following definitions.

Elements.

An *element* is a structural part of a protein centered around a given residue. It comprises five points, corresponding to the positions of the C_α atoms of the central residue, and its four sequential neighbors (two preceding and two following). For each model s in an ensemble, and each residue i , an element – denoted by E_i^s – is constructed. Residues for which an element cannot be built (e.g. N- and C-termini) are omitted.

Geometrical variability.

We consider pairs of elements, $E_{ij}^s := (E_i^s, E_j^s)$, and express structural deviation of a contact between two configurations r and s by RMSD of E_{ij}^r and E_{ij}^s . We use the following function to express the strength of a contact in terms of the whole ensemble:

$$G(i, j) := \max_{\text{pairs of states } (r, s)} \text{RMSD}(E_{ij}^r, E_{ij}^s).$$

The smaller the geometric variability, the stronger the contact.

We tested several statistics based on RMSDs of pairs of elements. This particular definition of G assumes that a strong contact is not “broken” in any pair of models. Conversely, a contact whose structural stability is breached at least once is assumed to be weak and not contributing to the stability of a given dynamic domain.

Note that conformational transitions may be rapid or insufficiently sampled. Therefore, defining geometrical variability in terms of some averaging statistic (e.g. mean, median) might lead to omitting significant structural changes occurring in a protein.

Contact matrix

We now describe a matrix representation of an edge-weighted graph, in which nodes correspond to residues. Because we used a spectral clustering algorithm which required that weights in the graph were in the interval $[0, 1]$, we needed to renormalize the geometrical variability. We calculated the weight between node i and j using the following *contact function*:

$$D(i, j) := \begin{cases} 1 & |i - j| \leq 1 \\ L_{\alpha, \beta}(G(i, j)) & \text{residues } i \text{ and } j \text{ are in contact} \\ 0 & \text{otherwise} \end{cases}$$

where $L_{\alpha, \beta}$:

$$L_{\alpha, \beta}(x) := \left(1 + e^{\frac{x - \alpha}{\beta}}\right)^{-1}$$

is a logistic function. We refer to matrix $[D(i, j)]_{ij}$ as the *contact matrix*.

Parameters $\beta > 0$ and $\alpha > 0$ allow for the customization of ResiCon. The logistic function, $L_{\alpha, \beta}$, can be thought of as a rescaling transformation for the measure of geometrical variability, G . It has a simple interpretation – values of G exceeding α are smoothly cut-off, where the degree of “smoothness” is determined by β . All results presented in this paper were acquired with default values of α and β :

$$\beta_{\text{default}} := \frac{1}{\sigma(G)} \quad \alpha_{\text{default}} := \mu(G) \quad (2.2)$$

where the σ and μ stand for standard deviation and mean taken over values of G for all pairs of elements.

Another feature of the logistic function becomes apparent if we consider an ensemble composed of (nearly) identical structures. This has two possible interpretations: the protein is very stable and no conformational changes exist, or that the provided ensemble does not represent such changes. Thus, ResiCon will assume that the contacts are strong – nearly as strong as the peptide bonds ($G \approx 0 \Rightarrow L_{\alpha, \beta} \approx 1$). Consequently, the contact matrix becomes a so-called *contact*

map, assigning binary values to pairs of residues (i.e. 1 if a contact occurred at least once, and 0 otherwise).

Clustering

The contact matrix may be treated as a *similarity matrix*, denoted W , and be used as input in a clustering procedure. Thus, we consider residues to be similar if they are likely to belong to the same dynamic domain. The identified clusters would then correspond to quasi-rigid structural parts.

The choice of a clustering algorithm is not a trivial task and for the identification of dynamic domains two crucial requirements need to be met. Firstly, contact matrices for various proteins vary in dimension and density and the clustering algorithm needs to perform well regardless of these variabilities. Secondly, the algorithm should facilitate an automated method of choosing the optimal number of clusters.

Agglomerative hierarchical clustering algorithms are one of the most popular approaches to clustering [41]. They follow a greedy scheme to construct a dendrogram encoding distances between clusters. This dendrogram can be cut at a certain height, which determines the number of clusters. If the height parameter could be set so that for all similarity matrices we would obtain high-quality clusters, the hierarchical clustering would have been a good candidate for a clustering procedure. However, as we explain in the *Supplementary Materials*, estimation of this parameter is difficult, and to determine the number of clusters we would need to extend the conventional hierarchical clustering with a measure of cluster quality.

This was one of the reasons we have chosen a spectral clustering algorithm, which has an inherent indicator of a partitioning's quality.

Spectral clustering.

Clustering algorithms based on finding the eigensystem of the similarity matrix (or more often a matrix derived from it) are termed *spectral algorithms*. They perform a clustering by minimizing the cost of cutting a graph into subgraphs, which agrees with our intuition about finding quasi-rigid parts based on a contact matrix. Optimal clustering is achieved by discarding contacts with the lowest total weight (as few and as weak as possible) to achieve a partitioning into unconnected regions.

In the case of the clustering algorithm used in ResiCon optimal partitioning is decoded from eigenvectors corresponding to the largest eigenvalues of a stochastic matrix $D^{-1}W$, where $D := \text{diag}(d(1), \dots, d(n))$ and $d(i) := \sum_{j=1, j \neq i}^n w(i, j)$ (see [77]). This transformation of the similarity matrix ensures that the identified clusters tend to have similar sizes, which prevents from identifying singular nodes as clusters. Spectral algorithms make no assumptions on the shape of clusters, and, in contrast to the greedy procedures, are insensitive to the ordering of vertices.

In Figure 2.2 we present a 150×150 similarity matrix with rows and columns ordered in two different ways. The ordering on the right is dictated by the spectral

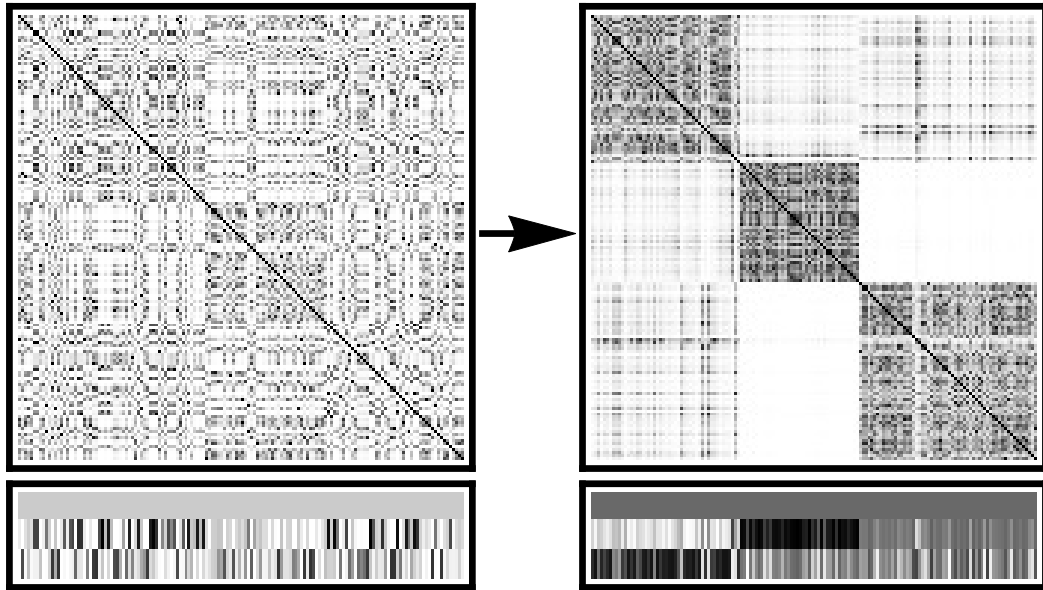


Figure 2.2: An example of a similarity matrix W and three eigenvectors of the stochastic matrix $D^{-1}W$ with two different orderings.

clustering – elements 1–50 were assigned to the first cluster, 51–90 to the second, and 91–150 to the third. To find these three clusters we use the first three eigenvectors (corresponding to three largest eigenvalues) of the stochastic matrix $D^{-1}W$. The first eigenvector always has a constant value in all positions and corresponds to a trivial clustering into a single group. The second eigenvector encodes a partitioning into two groups: nodes 51–90 in the first, and the remaining nodes in the second group. The third eigenvector allows to discern the third cluster, composed of nodes 1–50.

Clustering algorithm.

We used a spectral clustering algorithm in which partitioning of a graph is expressed in terms of a membership matrix χ . A short description of the procedure is presented below, for details refer to [77].

Let Y denote the matrix containing k eigenvectors of the $D^{-1}W$ stochastic matrix, corresponding to k largest eigenvalues. The procedure computes a linear mapping \mathcal{A} from the eigenvectors Y to the membership matrix:

$$\mathcal{A}Y = \chi$$

The element χ_{ij} of this matrix represents the membership of the i^{th} node in the j^{th} cluster. Therefore, if n is the number of nodes in the graph and k is the number of clusters, then $\chi \in \mathbb{R}^{n \times k}$.

This algorithm has two important features:

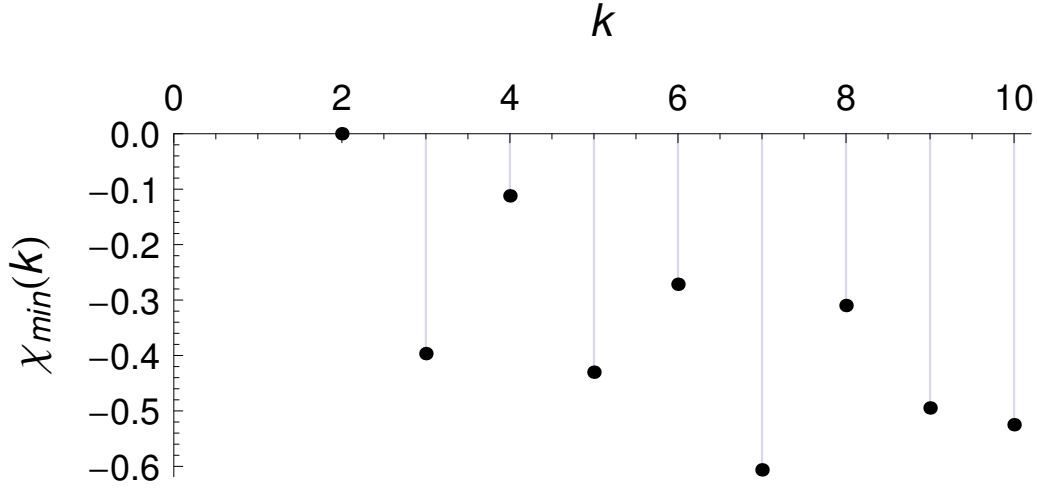


Figure 2.3: Values of the χ_{\min} indicator for $k = 2, \dots, 10$, computed for HIV-1 protease. Because $\chi_{\min}(4)$ is closer to 0 than $\chi_{\min}(6)$, the optimal number of clusters is 4.

- it computes the membership matrix χ which allows for overlapping clusters,
- it offers an indicator, called χ_{\min} , used to determine the optimal number of clusters.

Number of clusters.

ResiCon first checks if $k = 1$. To do so, a partitioning into two clusters is carried out. The spectral algorithm presented above finds an optimal cut [77] leading to clusters A and B (two sets of indices, that correspond to nodes). We express the *cost* of such cut by:

$$f := \frac{\sum_{i \in A} \sum_{j \in B} w_{ij}}{\left(\sum_{k, l \in A} w_{kl} \right) \left(\sum_{k, l \in B} w_{kl} \right)}$$

where w_{ij} is the weight of the edge between nodes i and j . The validity of a clustering into clusters A and B was checked by asserting that its cost is less than a given threshold. If the criterion was met, we assumed that a clustering into two or more clusters existed. Default threshold for f used to produce all results presented in this study was 0.1. Above this value we observed that compact static proteins were partitioned into short (less than four residues long) segments, which we regarded as improper dynamic domains. On the other hand, lower values resulted in a single dynamic domain assignment in several cases where two domains were apparent.

If $k > 1$, the optimal number of clusters is determined with the use of the indicator presented in [77]. Here, we give a short overview of the properties of χ_{\min} and propose a simple procedure for computing the optimal number of clusters. The

indicator $\chi_{\min}(k)$ is defined as the minimal element of the membership matrix χ found by partitioning it into k clusters.

In the case of $k = 2$ the indicator is always zero, $\chi_{\min}(2) = 0$. For $k > 2$ the indicator is less than zero, however if $\chi_{\min}(k) \approx 0$, the clustering into k clusters is the optimal one (Figure 2.3). Let us recall the notion of visualizing a clustering by a block-like similarity matrix. Roughly speaking, the value of $\chi_{\min}(k)$ resembles the deviation of the similarity matrix from the “pure” block-like form. However, it is difficult to decide which values of χ_{\min} are sufficiently close to zero to indicate the optimal number of clusters. Therefore, the problem at hand is: does the optimal number of clusters equal two, or more?

In our first approach, the optimal number of clusters was chosen based on a threshold – the optimal k was the one for which χ_{\min} was above a certain value. However, it was difficult to find the right threshold because values of χ_{\min} strongly depend on the number of nodes in the graph. Therefore, the following procedure was adapted in ResiCon:

1. Determine the cost of the optimal cut. If it exceeds the 0.1 threshold, the optimal number of clusters is $k = 1$. Otherwise, assume that $k > 1$ and continue the procedure.
2. Compute the values of χ_{\min} for partitionings into $3, 4, \dots, M$ clusters.
3. Find $k_1 > 2$ for which χ_{\min} is closest to 0, and $k_2 > 2$ for which χ_{\min} is the second closest to 0.
4. If $\chi_{\min}(k_1) > 0.5 \chi_{\min}(k_2)$, then the number of clusters is $k = k_1$. Otherwise, $k = 2$.

The 0.5 constant in the fourth point means that we choose k_1 as the number of clusters, if $\chi_{\min}(k_1)$ is closer to 0 than to the next best χ_{\min} (i.e. $\chi_{\min}(k_2)$). The maximal number of clusters M is set to 10 by default, but the user can specify a different number. All results presented in this paper were computed with $M = 10$.

In other words, ResiCon chooses $k > 2$ for which the indicator χ_{\min} is “relatively close” to 0. When no such value exists, a partitioning into two clusters is assumed to be optimal.

Hinges and interfacial regions

Hinges

We define hinges as parts of the structure satisfying both of the following conditions:

1. they do not belong conclusively (in terms of membership as explained below) to any dynamic domain ,
2. they are sequentially located between dynamic domains.

The first condition is tested in terms of membership: if a residue membership in any cluster does not exceed certain threshold χ_{hinge} it may belong to a hinge. The default value of the parameter is 0.65, but the user can specify a different value.

Interfacial regions

A residue is assumed to compose an interfacial region if two conditions are met:

1. it does not belong to any hinge,
2. it was in contact with a residue that does not belong to the same dynamic domain at least once.

Results comparison

According to our knowledge, no expert curated database of dynamic domains exists. Also, we are not aware of any quality measure for the dynamic domains assignment. Therefore, we compared different methods by analyzing agreement between their results.

We used the measure presented in [56], called *Variation of Information* (\mathcal{VI}) to analyze the results compatibility. It has the advantage of being a metric in the space of all partitionings of a given dataset. The downside of \mathcal{VI} is that its values do not lie in a fixed interval (e.g. $[0, 1]$), but instead have an upper bound that depends on the size of data. In our case data size equals the number of residues in a given protein. This means that values of \mathcal{VI} for partitionings of one protein are not directly comparable to the values acquired for partitionings of another protein, with a different number of residues. Nonetheless, when considering a particular protein, the \mathcal{VI} metric quantifies the agreement between different assignments of dynamic domains. Here we give an outline of the method, for details see [56].

Let us denote a clustering by \mathcal{C} . It is composed of clusters – mutually disjoint subsets C_1, \dots, C_k . That is, $\mathcal{C} = \{C_1, \dots, C_k\}$ such that $C_i \cap C_j = \emptyset$ for all pairs i, j . Assume that the numbers of points in consecutive clusters are n_1, \dots, n_k . Then, the probability that a random point from the dataset belongs to the i^{th} cluster is

$$P(i) := \frac{n_i}{n},$$

where n is the number of all points in the set. Note that $\sum_{i=1}^k n_i = n$. Analogously, let another clustering of the same set of points $\mathcal{C}' = \{C'_1, \dots, C'_{k'}\}$ be composed of clusters with $n'_1, \dots, n'_{k'}$ points. By $n_{ij'}$ we will denote the number of points assigned to cluster i in clustering \mathcal{C} and cluster j' in \mathcal{C}' . Then,

$$P(i, j') := \frac{n_{ij'}}{n}$$

is the probability of randomly choosing a point that belongs to both clusters.

The \mathcal{VI} measure is defined in terms of entropy and joint entropy of the probability distributions defined above. That is, if the entropy of clustering \mathcal{C} is expressed by:

$$H(\mathcal{C}) := - \sum_{i=1}^k P(i) \log_2 P(i),$$

and the joint entropy of two clusterings is given by:

$$H(\mathcal{C}, \mathcal{C}') := - \sum_{i=1}^k \sum_{j'=1}^{k'} P(i, j') \log_2 P(i, j'),$$

then the variation of information of the two clusterings is defined as:

$$\mathcal{VI}(\mathcal{C}, \mathcal{C}') := 2H(\mathcal{C}, \mathcal{C}') - H(\mathcal{C}) - H(\mathcal{C}')$$

Quality of dynamic domains

We consider dynamic domains to be structural parts of the protein, which move with respect to each other, but remain internally rigid. In order to assess which method for dynamic domains identification is better, a measure was required that would quantify the quality of a given assignment. We did not find such a scoring function in the literature, and propose the following geometrical measure called *total geometrical variability*:

$$Q := \sum_{i=1}^k \max_{\text{pairs of states } (r,s)} \text{RMSD}(D_i^r, D_i^s),$$

where D_i^s is the set of C_α atoms comprising the domain D_i in the state s , and k is the number of domains. Smaller values of Q indicate higher quality.

Note that typically, if a domain is structurally rigid, the maximal RMSD for that domain is smaller than the sum of maximal RMSDs of its two subsets. Therefore the proposed measure favors large, compact domains. It is also worth noting that for a trivial dynamic domain (single residue) the RMSD is undefined. We set its value to 0, although this artificially reduces Q (see *Quality analysis*).

2.4 Results and discussion

We compared ResiCon with two recent methods: GeoStaS [64] and PiSQRD [61]. The latter method represents the class of methods which identify dynamic domains by analyzing a single structure. We used a test set comprising 30 NMR-resolved protein structures exhibiting significant mobility, which was previously used in [70, 48], and in [64]. The set was initially proposed to examine the efficacy of a method of identifying structurally-stable cores in flexible proteins. These structures often contain a single rigid core with significant geometrical distortions present in peripheral regions (as indicated in [70] and also observable in ResiCon’s results).

We have also used ResiCon to analyze a canonical test case – the HIV-1 protease molecule – using an MD trajectory computed with a coarse-grained force field RedMD [35, 36] as input data. This example shows that ResiCon is capable of finding dynamic domains of a protein whose functionality depends on flexibility and mobility of its rigid parts.

We also explain the so-called *zebra effect* – a peculiar result produced by ResiCon, observed in several cases. This effect is especially strong when a suboptimal number of clusters is chosen.





		R	G	P1	P2
R		0	1.47	3.29	0.21
G		1.47	0	3.76	1.51
P1		3.29	3.76	0	3.29
P2		0.21	1.51	3.29	0

Figure 2.4: Values of \mathcal{VI} for partitionings of the 1d1d protein molecule. A pair of results produced by PiSQRD which had the highest \mathcal{VI} are denoted by P1 and P2.

Comparative analysis

ResiCon and GeoStaS are both designed to work on an ensemble of structures, and produce a single partitioning into dynamic domains. Both methods impose no assumptions about sampling and order of provided conformations. ResiCon uses maximal local distortions computed over all pairs of frames, indifferent to over- or undersampling of configurations, as long as they are present in the ensemble.

However, the PiSQRD server by default analyzes a single structure and estimates the so-called *low-energy modes*, which are the eigenvectors of the structural covariance matrix (under the canonical ensemble, i.e. assuming Boltzmann distribution of configurations). These low-energy modes are assumed to carry the information relevant to dynamic domains identification. We observed that the choice of an input structure influences the results significantly, but there is no definite criterion for choosing the *right* structure for the analysis. The PiSQRD server provided with a PDB file containing NMR models by default finds dynamic domains for the first model.

This might introduce a bias unfavorable for PiSQRD’s performance. We, therefore, decided to examine results produced by PiSQRD for every structure in the ensemble in order to compare ResiCon against its full capacity. The best dynamic domains were chosen and presented together with the results produced by ResiCon and GeoStaS in Table 2.1.

PiSQRD also gives a possibility of providing a set of low-energy modes extracted from a structural covariance matrix estimated from a set of structures, but this procedure is not straightforward and requires additional assumptions (see *Supplementary Materials*). We scrutinize the quality of dynamic domains found by PiSQRD from user-provided low-energy modes in section *Quality analysis*.

Because of a large number of models, it was not possible to provide a graphical representation for each partitioning. We therefore focused on values of the agreement measure \mathcal{VI} between results produced by the three methods.

To familiarize the Reader with values of the measure \mathcal{VI} , we take a look at two most distant partitionings produced by PiSQRD (denoted by P1 and P2) for an exemplary 1d1d protein, and how they relate to the results given by ResiCon (R) and GeoStaS (G) (Figure 2.4). The clustering P2 seems to be very similar to the one produced by ResiCon, while P1 gives a sliced-and-diced picture of the

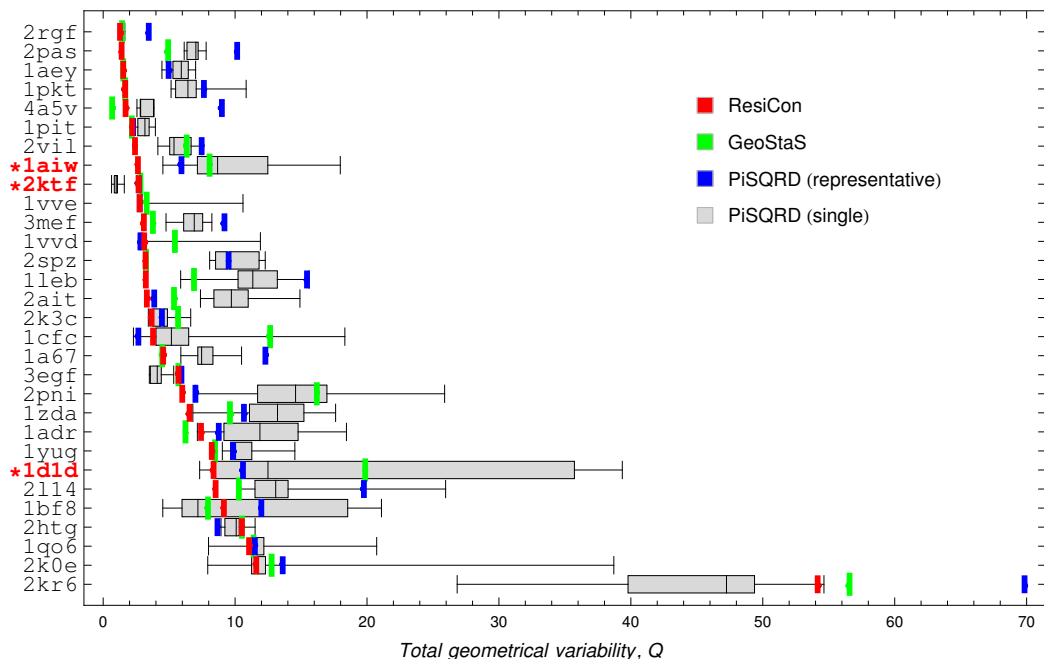


Figure 2.5: Box and whiskers plot of the dynamic domains quality score Q for ResiCon, GeoStaS, PiSQRD. In blue are Q values for PiSQRD’s dynamic domains determined from structural covariance matrices (see *Supplementary Materials*).

protein’s mobility. It seems that a partitioning produced by PiSQRD depends on physical properties embedded in a particular configuration. Although NMR models carrying information about dynamic domains may exist, others lead to chaotic partitionings. In order to produce a single clustering, PiSQRD would need a procedure for interpreting physical properties based on an ensemble of configurations.

Nearly 50% of the assignments produced by PiSQRD for 1d1d are coherent with the clustering given by ResiCon (see *Supplementary Materials*). On the other hand, the result given by GeoStaS differs from all results produced by PiSQRD. In fact, ResiCon and PiSQRD are more coherent than GeoStaS and PiSQRD, which can be expressed by the mean of \mathcal{VI} . However, it would be naive to use the mean value as an indicator of self-consistency of PiSQRD. Among partitionings produced by PiSQRD, $N(N - 1)/2$ agreements were calculated (where N is the number of models of 1d1d), whereas the comparison of ResiCon or GeoStaS with PiSQRD gave N values of \mathcal{VI} . Consequently, the average value of \mathcal{VI} between PiSQRD clusterings is not directly comparable with the average value of \mathcal{VI} , e.g. for ResiCon vs. PiSQRD.

Therefore, to give a better picture of the divergence of results we exploit the fact that \mathcal{VI} is a metric in the space of all clusterings. For a given partitioning we computed the radius of a ball centered at that partitioning, encompassing a certain fraction of the results. In Table 2.2 we provide values of radii of balls which encompass 25, 50 and 75% of the results. In the case of PiSQRD, we constructed

balls (for a given percentage of results) centered at each partitioning, and computed the mean values of their radii. The mean of radii does not carry the bias mentioned earlier. Additionally, histograms of \mathcal{VZ} for each protein can be found in *Supplementary Materials*.

Quality analysis

The box-and-whisker plot in Figure 2.5 provides a concise picture of quality scores Q (see *Methods*) of the dynamic domains assignments. It clearly shows that beyond a few exceptions ResiCon gives the best results. There are 7 notable exceptions: **4a5v**, **2ktf**, **3egf**, **1adr**, **1bf8**, **2htg** and **2kr6**.

The single dynamic domains found by PiSQRD (blue in Figure 2.5) were produced using low-energy modes, computed as eigenvectors of a structural covariance matrix estimated by superimposing all models in an ensemble on a representative structure. In the *Supporting Materials* we explain how this representative structure is chosen, and show qualities of dynamic domains found by PiSQRD using different methods of estimating the structural covariance matrix. It should be emphasized, however, that these dynamic domains strongly depend on the method of estimating the structural covariance matrix, which is not part of PiSQRD's functionality. Therefore, these results should only be treated as an additional insight into what the user can expect from a more complex analysis of NMR structures.

In the case of **1adr** GeoStaS gave partitionings with lower Q , by introducing a trivial domain (cutting off C- and N-termini – see *Supplementary Materials*). On the other hand, for **4a5v** GeoStaS identified a single domain, which gave a lower value of Q than any other partitioning.

For proteins **2ktf**, **3egf** and **2htg**, PiSQRD produced results with lower Q than ResiCon, by finding numerous small quasi-rigid fragments. However, for **2ktf** more than 25% of assignments found by PiSQRD contained a trivial, single-residue domain (indicated in red, with an asterisk). Although our measure does not penalize for this, we consider such behavior undesirable. Also, note that proteins **3egf** and **2htg** comprise 53 and 27 residues accordingly. It seems that in case of small proteins ResiCon often identifies a single domain, which does not necessarily result in the lowest Q .

The **2kr6** protein is an interesting example. It contains a flexible linker composed of more than 30 residues. As a consequence, partitionings with high values of Q are observed. Only PiSQRD was able to produce lower values of Q , by cutting the linker into many shorter parts. The example of the **2kr6** protein shows, that ResiCon does not consider unstructured regions to be separate dynamic domains. Instead, residues constituting linkers and lacking long-distance contacts, are assigned to dynamic domains which are closest in sequence.

Comments

Although PiSQRD's capability of analyzing a single structure may be considered an advantage, results presented in Table 2.2 and Figure 2.5 show that there is a large

discrepancy for different configurations of the same protein. Nevertheless, based on the histograms of \mathcal{VI} presented in *Supplementary Materials* and the values of radii in Table 2.2, we conclude that in most cases results given by ResiCon and PiSQRD are mutually more coherent, than GeoStaS and PiSQRD (e.g. 1cfc, 1qo6, 1vvd, 1vve, 1yug, 2k3c). Notable exceptions are: 2rgf, 2pas, 3mef and 1zda. For these proteins ResiCon did not find any significant structural transitions and achieved the lowest value of Q (see Figure 2.5) by assigning a single domain.

Dynamic domains found by ResiCon are generally larger (particularly: 2k3c, 1d1d, 2k0e and 2kr6). Conversely, GeoStaS often allocates flexible N- and C-terminal parts (e.g. 1adr, 1qo6, 1vve, 3egf) as quasi-rigid parts. The size of dynamic domains is also the main difference between ResiCon and PiSQRD. In case of small, static proteins (such as 1aey, 1pkt, 1pit, 2spz, 2ait, 3egf and 1zda), PiSQRD identifies numerous small and often trivial dynamic domains. ResiCon on the other hand detects no significant conformational changes (by analyzing an ensemble), and assigns a single dynamic domain. We observe that in many cases ResiCon identified a single domain which had the lowest value of Q among all partitionings (see 2rgf, 2pas, 1aey, 1pkt, 1pit, 2vil, 1aiw, 3mef, 2spz, 1leb, 2ait, 2pni, 2114). In these cases ResiCon correctly detected that no significant transitions were present in the protein. Therefore, unlike PiSQRD and GeoStaS, ResiCon can reliably indicate whether conformational changes occur in an ensemble of structures.

It is also noteworthy that ResiCon does not employ any post-processing procedures. This keeps the algorithm simple and clean, but results in a discontinuity of certain partitionings (see 4a5v, 1bf8 and 1vvd), which we refer to as the *zebra effect*. Although this seems to be an artifact of the clustering algorithm, we will take a closer look at this effect and show that it may also carry valuable information referring to the protein’s dynamics.

HIV-1 protease

An analysis of an MD trajectory of the HIV-1 protease showcases ResiCon’s capabilities. This protein undergoes substantial conformational changes associated with opening/closing of its structural parts, so-called *flaps* [40]. A database of X-ray-resolved structures, representing configurations which the protease can attain is available [76].

We examined a set of configurations of the HIV-1 protease acquired from a simulation carried out using the RedMD package [36]. This coarse-grained force field was designed to simulate intramolecular motions in proteins and nucleic acids. It has correctly predicted the flap-opening motion in the HIV-1 protease, which is known biological fact [40], and was independently confirmed by an all-atom MD simulation [66]. Roughly, about 1% of the trajectory seems to exhibit significant conformational transitions (flap opening events). This is a typical scenario in MD simulations, where a transition between meta-stable states is swift and fairly short. We needed a set of representative structures in order to find dynamic domains using ResiCon. From the whole trajectory, we extracted a set of 200 configurations

using a generic procedure facilitating the Principal Component Analysis (PCA), implemented in the R programming language (in the `bio3d` package [37]) – see *Supplementary Materials*.

Values of χ_{\min} for different numbers of clusters are given in Figure 2.3. The optimal number of clusters according to our procedure is 4. Figure 2.6 depicts the dynamic domains found by ResiCon and two representative configurations of the protease, as well as results from GeoStaS and PiSQRD. Because the sample of configurations was drawn according to the Boltzmann distribution, we were able to straightforwardly estimate the structural covariance matrix and provide PiSQRD with well-founded low-energy modes, and acquire a high-quality partitioning into two domains. Results produced by GeoStaS were acquired from the whole trajectory of the protease.

Dynamic domains identified by ResiCon have the highest value of the Q measure. Note that ResiCon does not try to minimize Q , but to produce a clustering with the optimal value of the χ_{\min} indicator. Consequently, at the cost of a slightly higher Q (ca. 1\AA) we arrive at a partitioning which corresponds to the biologically relevant sub-division of the protein. It is also worth mentioning that ResiCon’s partitioning into $k = 2$ clusters incidentally leads to an identical assignment as the one produced by PiSQRD.

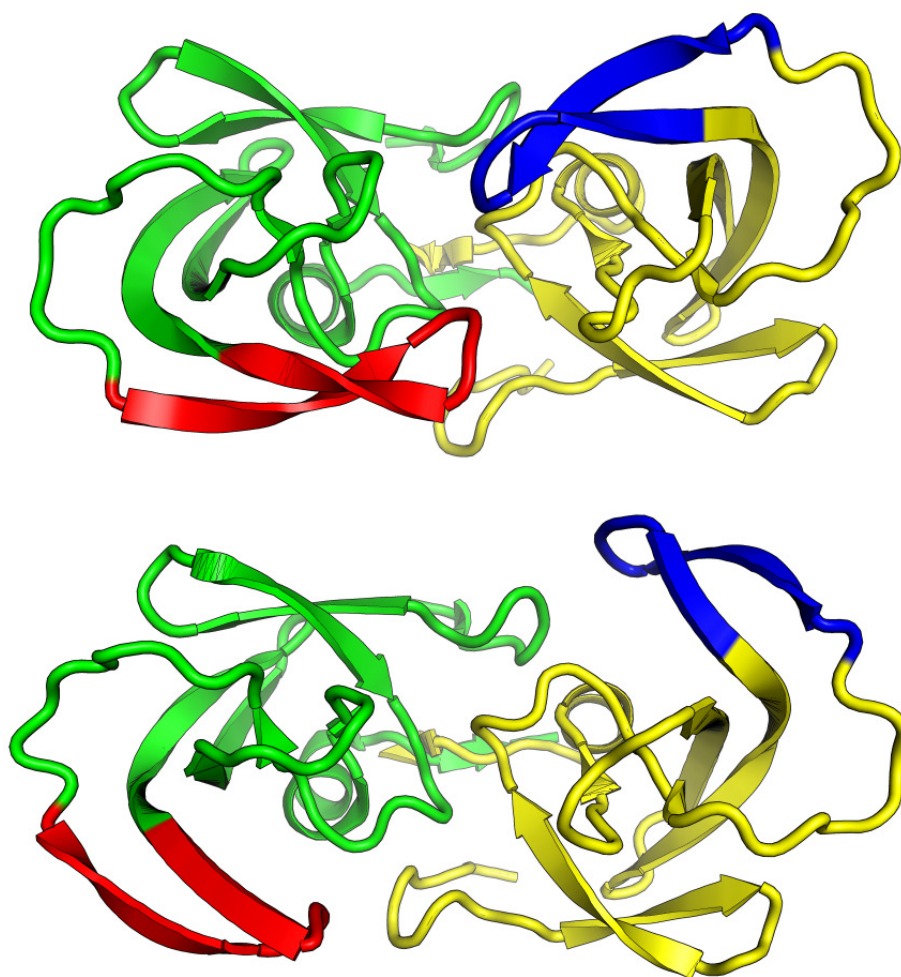
Motions of the flaps between the closed and open states are crucial in the functionality of the HIV-1 protease. It is also known that throughout their motion they remain quasi-rigid [30]. Therefore, using ResiCon, we successfully extracted a simplified picture of the mobility of HIV-1 protease, which agrees with experimental knowledge.

Alternative partitioning into 6 clusters (see Figure 2.7) also deserves interest. Apart from the flaps, additional dynamic domains resembling “arms” carrying the flaps can be observed. Note that conversely to the case of the four dynamic domains, the quasi-static regions are similar, but not ideally reflective. This indicates that motions of the two centrally symmetric sub-units of the HIV-1 protease in the provided trajectory were slightly different.

It can be also seen that domains are discontinuous. Especially residues belonging to the N-terminal lobe (yellow) and to the arm (cyan) are interleaved. In the next section we will analyze this effect, and show that this partitioning also carries valuable information.

The zebra effect

Dynamic domains found by ResiCon may include residues that seem to be pulled out from another quasi-rigid fragment. This is indicated as discontinuities (stripes) in Figure 2.7. Let us take a look at an example of such an extracted residue and try to understand the source of this effect. ResiCon assigned the ASN-83 residue to the arm (gray) dynamic domain (see Figure 2.8). However, its sequential neighbors belong to the lobe domain (yellow). The reason of this discontinuity is that, although ASN-83 has peptide bonds with its yellow neighbors, they are outweighed by contacts with the gray residues (see Figure 2.8). The discontinuity



Method	Clustering	Q
GeoStaS		9.500
ResiCon		10.409
PiSQRD		9.189

Figure 2.6: Two representative conformations of HIV-1 protease with flaps being closed and open, and summary of results produced by GeoStaS, ResiCon and PiSQRD. Structures are colored according to the dynamic domains detected by ResiCon.

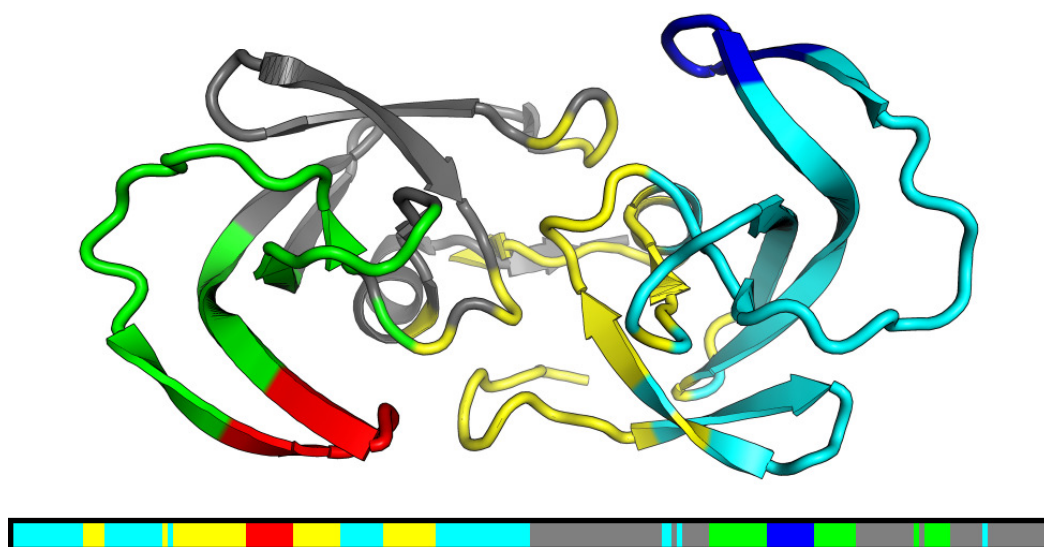


Figure 2.7: More subtle division emerges when the number of clusters is set to 6.

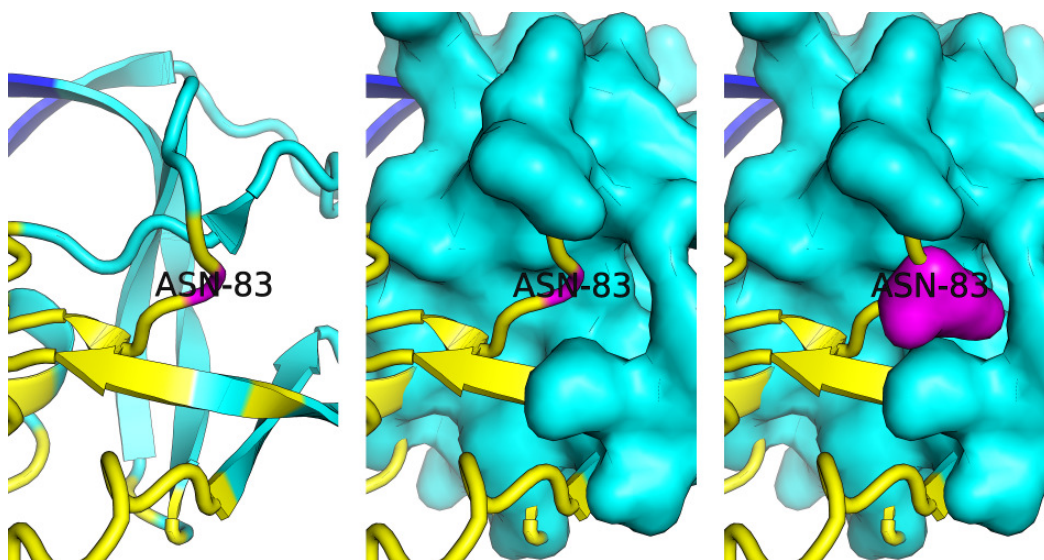


Figure 2.8: From the analysis carried out by ResiCon one can see that throughout the trajectory ASN-83 (magenta) not only remains close to residues in the N-terminal arm domain (gray) but also fills a cavity in that dynamic domain.

suggests that throughout the trajectory ASN-83 remained docked in the cavity of the arm domain, while its peptide bonds formed an axis of a hinge.

It seems that the zebra effect is not accidental, and that it can be used to find residues which act as “pivot points”. However, this effect is volatile and depends on small fluctuations in the input. Residues are assigned to dynamic domains based on the fuzzy membership matrix. At certain positions values of membership to different domains may be almost equal, and when discretized cause emergence of stripes. Therefore, to strengthen the identification of these “pivot point” residues, a more thorough analysis of the membership matrix is required.

2.5 Summary of results

Typically, protein structures are flexible and mobile, and their conformational changes may be crucial in facilitating signaling and metabolic processes in which they participate. Breaking a structure into dynamic domains may be compared to discovering gears and levers connected by cogs and pegs analogous to those found in classical machines. Such analyses allow us to better understand molecular mechanisms responsible for biological functions (e.g. [73]), facilitate MD simulations in large time-scales with simplified forcefields (e.g. [69]), or discover potential binding sites when designing inhibitors (e.g. [83]).

We have presented a universal tool for discovering dynamic domains in proteins. ResiCon is capable of analyzing a single structure, or an NMR ensemble of structures provided in a PDB format. It can also be applied to the set of independently obtained structures (e.g. X-ray crystallographic structures obtained under different conditions). In any case, it provides a complete set of results comprising partitioning of the molecule into dynamic domains with additional highlighting of residues composing hinges and interfacial regions. ResiCon also provides an indicator of partitioning quality, and suggests the optimal number of domains.

We tested our method using the reference set of NMR structures. It is comparable or better than the recently developed GeoStaS and PiSQRD. Apart from giving more compact dynamic domains, it is also capable of distinguishing structures composed of a single quasi-rigid region.

We have made ResiCon available online.¹ To make the analysis feasible and limit the number of uploads, queries may be rerun with changed parameters. Also, all queries and results are stored on our server.

¹<http://dworkowa.imdik.pan.pl/EP/ResiCon>

PDB code	Method	Clustering	Q	PDB code	Method	Clustering	Q
2rgf	GeoStaS		1.430	2k3c	GeoStaS		5.678
	ResiCon		1.275		ResiCon		3.676
	PiSQRD		1.421		PiSQRD		3.422
2pas	GeoStaS		4.926	1cfc	GeoStaS		12.658
	ResiCon		1.398		ResiCon		3.780
	PiSQRD		6.141		PiSQRD		2.294
1aey	GeoStaS		1.520	1a67	GeoStaS		4.478
	ResiCon		1.520		ResiCon		4.548
	PiSQRD		4.455		PiSQRD		5.887
1pkt	GeoStaS		1.670	3egf	GeoStaS		5.700
	ResiCon		1.670		ResiCon		5.713
	PiSQRD		5.147		PiSQRD		3.499
4a5v	GeoStaS		0.678	2pni	GeoStaS		16.195
	ResiCon		1.698		ResiCon		5.994
	PiSQRD		2.561		PiSQRD		6.993
1pit	GeoStaS		2.177	1zda	GeoStaS		9.610
	ResiCon		2.209		ResiCon		6.549
	PiSQRD		2.171		PiSQRD		6.798
2vil	GeoStaS		6.329	1adr	GeoStaS		6.241
	ResiCon		2.414		ResiCon		7.417
	PiSQRD		4.146		PiSQRD		7.139
1aiw	GeoStaS		8.066	1yug	GeoStaS		8.479
	ResiCon		2.636		ResiCon		8.230
	PiSQRD		4.527		PiSQRD		9.033
2ktf	GeoStaS		2.822	1d1d	GeoStaS		19.862
	ResiCon		2.693		ResiCon		8.368
	PiSQRD		0.636		PiSQRD		7.313
1vve	GeoStaS		3.299	2114	GeoStaS		10.284
	ResiCon		2.767		ResiCon		8.525
	PiSQRD		2.767		PiSQRD		10.411
3mef	GeoStaS		3.770	1bf8	GeoStaS		7.947
	ResiCon		3.086		ResiCon		9.153
	PiSQRD		4.765		PiSQRD		4.519
1vvd	GeoStaS		5.439	2htg	GeoStaS		10.514
	ResiCon		3.110		ResiCon		10.514
	PiSQRD		2.821		PiSQRD		8.665
2spz	GeoStaS		3.244	1qo6	GeoStaS		11.403
	ResiCon		3.212		ResiCon		11.057
	PiSQRD		8.071		PiSQRD		7.985
11eb	GeoStaS		6.885	2k0e	GeoStaS		12.771
	ResiCon		3.234		ResiCon		11.604
	PiSQRD		5.874		PiSQRD		7.927
2ait	GeoStaS		5.370	2kr6	GeoStaS		56.578
	ResiCon		3.310		ResiCon		54.160
	PiSQRD		3.871		PiSQRD		26.834

Table 2.1: Summary of results produced by GeoStaS, ResiCon and PiSQRD. Dynamic domains shown for PiSQRD are those for which the lowest value of Q was achieved.

	ResiCon vs. GeoStaS	ResiCon vs. PiSQRD			GeoStaS vs. PiSQRD			PiSQRD vs. PiSQRD		
	\mathcal{VI}	$r_{25\%}$	$r_{50\%}$	$r_{75\%}$	$r_{25\%}$	$r_{50\%}$	$r_{75\%}$	$\langle r_{25\%} \rangle$	$\langle r_{50\%} \rangle$	$\langle r_{75\%} \rangle$
2rgf	0.62	0.45	0.48	0.57	0.26	0.26	0.32	0.01	0.14	0.20
2pas	1.94	2.43	2.45	2.47	2.00	2.08	2.11	0.61	0.96	1.23
1aey	0.00	1.99	2.33	2.37	1.99	2.33	2.37	0.77	1.10	1.52
1pkt	0.00	2.07	2.24	2.36	2.07	2.24	2.36	1.38	1.63	1.84
4a5v	1.00	1.08	1.12	1.19	1.53	1.54	1.60	0.24	0.29	0.38
1pit	1.59	1.76	2.05	2.14	2.42	2.55	2.74	1.35	1.70	1.94
2vil	1.89	1.19	1.39	1.89	2.31	2.49	2.67	1.11	1.36	1.57
1aiw	1.64	1.66	1.80	2.13	2.58	2.69	2.77	1.30	1.53	1.90
2ktf	0.80	1.69	2.01	2.06	2.15	2.36	2.48	0.93	1.17	1.74
1vve	1.05	0.00	0.00	0.00	1.05	1.05	1.05	0.62	0.62	0.62
3mef	0.32	1.30	1.43	1.50	1.21	1.32	1.40	0.64	0.86	1.09
1vvd	1.24	0.21	0.21	0.21	1.26	1.26	1.26	0.31	0.31	0.31
2spz	0.13	1.42	1.62	1.94	1.37	1.59	1.96	1.28	1.60	1.73
1leb	0.61	2.05	2.27	2.41	2.33	2.37	2.45	0.92	1.21	1.52
2ait	0.71	1.26	1.65	1.82	1.49	1.67	1.99	0.99	1.31	1.52
2k3c	1.86	0.82	0.97	0.98	1.61	1.69	1.69	0.16	0.42	0.63
1cfc	1.60	0.41	0.44	0.97	1.41	1.48	1.86	0.77	0.92	1.30
1a67	1.07	1.90	1.96	2.06	2.22	2.38	2.69	1.58	1.93	2.12
3egf	0.14	1.30	1.40	1.43	1.29	1.41	1.45	0.63	0.91	1.14
2pni	1.60	1.16	1.58	1.78	2.00	2.25	2.35	1.18	1.44	1.65
1zda	0.79	1.37	1.55	1.96	0.83	1.28	1.78	1.26	1.54	1.75
1adr	0.84	0.80	1.30	1.81	1.11	1.64	2.22	1.19	1.57	1.94
1yug	1.79	0.90	1.00	1.31	1.98	2.12	2.15	0.70	0.99	1.29
1d1d	1.47	0.00	0.21	3.15	1.47	1.51	3.80	1.08	3.08	3.17
2l14	0.55	0.49	0.57	0.75	0.91	0.98	1.18	0.61	0.71	0.95
1bf8	1.61	0.74	1.27	3.22	1.69	1.89	4.20	1.13	1.74	2.99
2htg	0.00	1.36	1.58	1.91	1.36	1.58	1.91	0.84	1.13	1.30
1qo6	0.88	0.63	0.72	0.88	0.87	0.98	1.18	0.57	0.71	0.94
2k0e	0.97	0.59	0.66	0.82	1.05	1.16	1.31	0.72	0.87	1.03
2kr6	1.11	0.79	0.81	0.84	0.85	0.87	0.93	0.61	0.68	0.82

Table 2.2: Discrepancies in assignments expressed by radii of balls encompassing 25%, 50% and 75% of results. The ordering of the results is the same as in Figure 2.5, i.e. best-scoring results are first.

Chapter 3

Towards the identification of molecular cogs

The text of Chapter 3 is a verbatim citation of the work published in the *Journal of Computational Chemistry* [25]. The co-author, Bogdan Lesyng, participated in discussions referring to formulation of the problem and possible approaches to solve it. He also helped in preparing the publication.

Motivation and set-up

Computer simulations of molecular systems allow determination of microscopic interactions between individual atoms or groups of atoms, as well as studies of intramolecular motions. A comprehensive overview of conventional causality analysis methods of molecular structural transformations is presented in [22]. In general, the identification of causal relations hidden within such transformations is very difficult. In the thermodynamic analytical frame In this chapter we present a novel approach set in the thermodynamic analytical frame, in which structural and functional properties of molecules are related to their free energy changes. In order to better understand such properties, it is required to deepen our knowledge of free energy contributions arising from molecular subsystems in the course of structural transformations.

Below, we present a method of quantifying energetic contribution of each pair of atoms to the total free energy change along a given collective variable is presented. With the help of a genetic clustering algorithm, we propose a division of the system into two groups of atoms referred to as *molecular cogs*. Atoms which cooperate to push the system forward along a collective variable are referred to as *forward cogs*, and those which work in the opposite direction as *reverse cogs*.

3.1 Introduction

Molecular dynamics (MD) simulation methods and advanced algorithms for calculating free energy bring us closer to predicting the physical properties of biomolecules

[78, 16, 19]. However, computer simulations are not limited to interpreting experimental results. *In silico* one may also process MD data which can provide much more detailed information than that accessible in any experiment. The key to a deeper understanding of complex molecular systems is the extraction of valuable information from data produced in such simulations.

Biomolecules carry out their functions through conformational transitions between meta-stable states. Such phenomena can be simplified and described by a selected reaction coordinate which, in many cases, is a collective variable [17]. A valuable result of many simulation procedures (such as Umbrella Sampling [74], Thermodynamic Integration [49], Adaptive Biasing Force [19], and others[38]), is a free energy profile, the so-called Potential of Mean Force (PMF) [49]. In this study, the discussion is limited to one-dimensional PMFs, although it should be noted that the aforementioned methods can be formulated more generally and produce multi-dimensional free energy profiles. Of course, a one-dimensional PMF may be an oversimplification of what is going on in a complex system, but a meaningful collective variable usually leads to a free energy profile which makes the complicated transition more comprehensible [60, 16]. However, what the PMF does not provide is the information about what drives transitions between meta-stable states.

Various attempts to understand and describe the internal mechanics of molecular systems have already been reported [4, 6, 15, 51, 65, 1, 68]. We did not, however, find any general-purpose approach for studying a broader spectrum of cases, and in particular a methodology explaining the cause of a transition in terms of free energy contributions arising from certain parts of a molecule.

In this study we propose a new approach of analyzing the tendencies of a molecular system to undergo a selected structural transition. The main idea is to look at a shift along a collective variable as an effect of two opposing tendencies generated by interactions within the molecule. Our method indicates two groups of atoms – referred to as *molecular cogs* – which, through cooperative interactions, are the source of these tendencies. For this purpose, we construct undirected graphs with weights between nodes expressed by energetic contributions to the free energy change arising from pair-wise interactions between atoms. These graphs are then partitioned into subsystems – corresponding to molecular cogs – using a genetic clustering algorithm. We present results for small, model systems which served as case-studies for the identification of molecular cogs and for testing if their functioning agrees with our intuitions.

3.2 Methodology

Decomposition of the Helmholtz free energy, A , was investigated in the past, most notably by Karplus and coworkers[8, 10]. The aim was the determination of contributions to the free energy coming from components comprising the potential energy of the system. These potential energy components might come from different interaction types, or from cooperation between subsystems of the whole molecule. The strategy of expressing the potential energy of a system as a sum of terms,

$U = \sum_i U_i$, and computing the contribution of each of these terms to the free energy is ineffective because additivity in the potential energy components does not imply additive contributions to the entropy[23].

Two attempts of describing contributions to the free energy coming from parts of the system are worth mentioning. The first one was an approximate approach based on Free Energy Perturbation, in which higher order terms were neglected [10]. These terms are not, however, negligible, which severely limits the applicability of this method. An alternative route, employing Thermodynamic Integration, was also explored [8]. However, this approach is also limited, namely – values of the free energy contributions depend on the choice of the integration path.

We propose a different approach, in which pair-wise energetic contributions to the free energy are readily attained. Alas, as in the aforementioned methods of free energy decomposition, our current formulation struggles with a description of the entropic contributions, and at the present is not included in our method. The current implementation is primarily applicable to small molecules for which the role of entropy in structural transitions is negligible (see an example of a PMF in the *Results* section).

Our analysis originates from the following formula [13]:

$$A'(\xi^*) = \langle m_\xi \nabla U \cdot \mathbb{M}^{-1} \nabla \xi \rangle_{\xi^*} - \langle \mathbf{v} \cdot \nabla (m_\xi \nabla \xi) \mathbf{v} \rangle_{\xi^*}, \quad (3.1)$$

where A' is the derivative of the free energy with respect to the collective variable ξ , \mathbb{M}^{-1} is a diagonal matrix of inverse masses, and \mathbf{v} are velocities. The term m_ξ is defined by:

$$m_\xi := \left[\sum_i m_i^{-1} \left(\frac{\partial \xi}{\partial \mathbf{x}_i} \right)^2 \right]^{-1} \quad (3.2)$$

and can be interpreted as inertia of an effective point mass moving along the ξ coordinate.

On the right-hand side of Equation (3.1) we have, respectively, energetic and entropic contributions to the free energy change, both expressed as conditional averages, with the collective variable fixed at ξ^* . In the *Supporting Information* we briefly explain how these averages are estimated *via* constrained Molecular dynamics (cMD) simulations, and highlight an important limitation of this method.

Note, that if we consider the potential energy as a sum of interaction components:

$$U = U^{ele} + U^{vdw} + U^{bond} + \dots = \sum_i^I U^I$$

the energetic contribution in Equation (3.1) maintains this additivity:

$$\langle m_\xi \nabla U \cdot \mathbb{M}^{-1} \nabla \xi \rangle_{\xi^*} = \sum_i^I \langle m_\xi \nabla U^I \cdot \mathbb{M}^{-1} \nabla \xi \rangle_{\xi^*}, \quad (3.3)$$

as was noted by Chipot et al. [17].

In our numerical experiments we used a force field with the following set of interaction types:

- *ele* (electrostatic interactions);
- *vdw* (van der Waals interactions);
- *bond* (2-body bonded interactions);
- *angl* (3-body angle interactions);
- *tors* (4-body torsional interactions).

We shall also consider:

- *nbd* (non-bonded interactions, the sum of *ele* and *vdw* interactions);
- *conf* (conformational interactions, composed of *bond*, *angl* and *tors* interactions);
- *total* (the sum of *nbd* and *conf* interactions).

3.2.1 Decomposition of the energetic contribution

Our purpose was to identify the molecular cogs, i.e. sets of atoms, which cooperate and push the whole system forward/backward along a reaction coordinate. We approached this problem by converting it into the task of finding clusters in a graph. Nodes in such a graph correspond to atoms, whereas edges represent cooperation of pairs of atoms. A natural measure of such cooperation can be introduced by taking into account the energetic contribution to the free energy in Equation (3.1).

Electrostatic, van der Waals and two-atom chemical bond interactions can be readily transformed into weights in the graph. For example, electrostatic interactions between atoms α and β lead to the following contribution:

$$c_{\alpha\beta}^{ele}(\xi^*) := \frac{1}{m_\alpha} \left\langle m_\xi \frac{\partial U^{ele}}{\partial \mathbf{x}_\alpha} \cdot \frac{\partial \xi}{\partial \mathbf{x}_\alpha} \right\rangle_{\xi^*} + \frac{1}{m_\beta} \left\langle m_\xi \frac{\partial U^{ele}}{\partial \mathbf{x}_\beta} \cdot \frac{\partial \xi}{\partial \mathbf{x}_\beta} \right\rangle_{\xi^*}. \quad (3.4)$$

Alas, n -body potential energy components cannot, in general, be decomposed into a sum of pair interactions. However, because the free energy contributions are of the form $\nabla U \cdot \mathbb{M}^{-1} \nabla \xi$, such decomposition is not required.

To clarify, let us consider a 3-body potential energy component, e.g. $U^{angl}(\mathbf{x}_\alpha, \mathbf{x}_\beta, \mathbf{x}_\gamma)$. The weight $c_{\alpha\beta}^{angl}$ of an edge joining nodes α and β represents the contribution coming from atoms α and β which interact *via* U^{angl} , while the γ atom is kept at a fixed position. By asserting that the sum of pair-wise contributions is equal to the total contribution arising from $U^{angl}(\mathbf{x}_\alpha, \mathbf{x}_\beta, \mathbf{x}_\gamma)$, i.e.:

$$\sum_{i>j} c_{ij}^{angl}(\xi^*) = \sum_{i=\alpha,\beta,\gamma} \frac{1}{m_i} \left\langle m_\xi \frac{\partial U^{angl}}{\partial \mathbf{x}_i} \cdot \frac{\partial \xi}{\partial \mathbf{x}_i} \right\rangle_{\xi^*},$$

we arrive at the following definition of the cooperation term:

$$c_{\alpha\beta}^{angl}(\xi^*) := \frac{1}{2} \sum_{i=\alpha,\beta} \frac{1}{m_i} \left\langle m_\xi \frac{\partial U^{angl}}{\partial \mathbf{x}_i} \cdot \frac{\partial \xi}{\partial \mathbf{x}_i} \right\rangle_{\xi^*},$$

where the $1/2$ means that the overall *angl* contribution is evenly distributed between the cooperation terms: $c_{\alpha\beta}^{angl}$, $c_{\beta\gamma}^{angl}$ and $c_{\alpha\gamma}^{angl}$.

For the general case of an n -body energy component, U^I , we propose the following definition of the cooperation term between atoms α and β :

$$c_{\alpha\beta}^I(\xi^*) := \frac{1}{n-1} \sum_{i=\alpha,\beta} \frac{1}{m_i} \left\langle m_\xi \frac{\partial U^I}{\partial \mathbf{x}_i} \cdot \frac{\partial \xi}{\partial \mathbf{x}_i} \right\rangle_{\xi^*}.$$

Note that we propose to distribute the cooperation evenly among all pairs of atoms involved in the interaction I . With this definition we constructed graphs for all interaction types, I .

A matrix, $\mathcal{C}^I(\xi^*)$, such that $[\mathcal{C}^I(\xi^*)]_{\alpha\beta} := c_{\alpha\beta}^I(\xi^*)$ constructed for a particular value of the collective variable ξ^* and the interaction type I is referred to as the *transient cooperation matrix for I* (see Figure 3.1). For convenience, the transient cooperation matrix for the *total* interaction type does not contain any superscript:

$$\mathcal{C}(\xi_i) := \sum_I \mathcal{C}^I(\xi_i).$$

Note that the sum $\sum_{\alpha>\beta} [\mathcal{C}(\xi_i)]_{\alpha\beta}$, is the overall energetic contribution to $A'(\xi_i)$ in Equation (3.1).

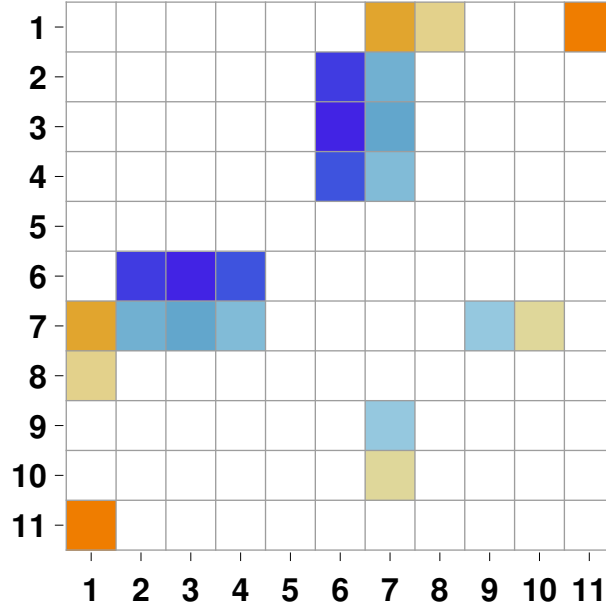


Figure 3.1: **Graphical representation of a transient cooperation matrix for *nbd* interactions.** Warm colors denote positive contributions, whereas blue – negative. White squares correspond to nil values.

Comment on the entropic contribution

The second term on the right-hand side of Equation (3.1) does not explicitly depend on U . It is possible to propose pair-wise contributions of this quantity in order to construct matrices which in turn could be used as input for the genetic clustering procedure described below. However, calculating these entropic contributions would require computation of the Hessian of the collective variable (the matrix of second-order derivatives with respect to atom coordinates) in each step of the simulation, which would make the computational cost of the procedure prohibitively high. We speculate on a possible solution of this problem in the *Summary of results* section.

3.2.2 Clustering

Data clustering is the procedure of finding disjoint subsets (clusters) of objects that share a high affinity, and are dissimilar to objects from other clusters. To carry out a clustering procedure we need a pair-wise affinity measure; thus, the data set is often represented by a weighted graph. Such a graph is encoded by an *affinity matrix* with elements corresponding to weights between nodes. In this study we were looking for groups of atoms whose cooperation decreased (increased) the free energy, thus pushing (pulling) the whole system forward (backward) along a reaction coordinate. Thus, we took the pair-wise energetic cooperation between atoms introduced earlier for the affinity measure.

We will refer to the clusters identified in a transient cooperation matrix as *transient molecular cogs*. The free-energy-reducing group is referred to as *forward cogs*, and the second group (which, conversely, increases the free energy gap) as *reverse cogs*. We, therefore, assume that a step taken by the molecule along a reaction coordinate is a consequence of a resultant tendency generated by the molecular cogs. But cross-cooperation between atoms from different groups can also occur, and we designate this unwanted effect 'gear grinding' (see the *Results* section).

Affinity Propagation algorithm

A considerable challenge associated with clustering is that different procedures produce different results. Therefore, it was crucial that the *right* clustering algorithm was chosen. We anticipated that the optimal algorithm should recover molecular cogs with as little gear grinding as possible, but we could not estimate how large this effect might be. To gain some insight into how molecular cogs might look like, we chose a clustering algorithm which is known to be successful in other applications.

Most clustering algorithms assume a non-negative affinity measure, whereas cooperation between atoms can be negative as well as positive. Among those algorithms which accepted negative values, Affinity Propagation (AP) was the procedure found to be the most promising[31]. The AP algorithm searches for *exemplars*, i.e. nodes within the graph around which non-exemplars are grouped, thus forming a cluster. It is an iterative procedure, where, in each step, "messages" between

nodes are exchanged to designate exemplars and their followers.

In our case, the sign of cooperation is arbitrary i.e. it depends on the direction of the reaction coordinate, ξ . It was crucial to ensure that molecular cogs found for a reaction coordinate $\xi' := -\xi$ were the same as those found for ξ . We resolved this problem by carrying out two clusterings with the AP method: one for $\mathcal{C}(\xi)$, and the second one for $-\mathcal{C}(\xi)$, in which the sign of all elements is changed, such as produced for ξ' . Description of the AP algorithm and our method for merging two clusterings can be found in *Supporting Information*.

One of the drawbacks of the AP method, similarly as in other clustering procedures, is that it requires several parameters, which influence the outcome. It appeared that small variations in parameters lead to markedly different results, and it was difficult to find a single set of parameters suitable for all cases (see the discussion in the *Supporting Information*). Nevertheless, the AP method gave us a first estimation of how molecular cogs look like, and it appears to be sufficiently good to initialize the genetic clustering procedure (see the *Genetic clustering* section).

Objective function

Molecular cogs identified by the AP algorithm were laden with low gear grinding, but that was not always a valuable finding. We noticed that in cases where no good partitioning was achievable, the AP method produced one-element clusters, which we considered to be artificial and incorrect.

The AP algorithm performs a clustering which maximizes the so-called *net similarity* [31]. This objective function, although helpful in many applications, does not carry any physical meaning. Note that cooperation, which we used to express affinity between atoms, translates into free energy differences, and thus into the tendency of the whole system to move along a collective variable. Molecular cogs should not only be laden with low gear grinding, but also cooperation generated by the cogs should cover the overall free energy change as much as possible.

To clarify, given a cooperation matrix \mathcal{C} , the sum of all negative (positive) elements is the overall tendency of the system to go forward (backward) along a reaction coordinate. It is desired for forward (reverse) cogs to cover as much of this overall tendency as possible. When there is only one atom in a cluster, there is no coverage, even though gear grinding might be small.

Let us denote by FC the set of atom indices, which were assigned to the forward cogs, and analogously by RC the set of atoms assigned to the reverse cogs. Given a cooperation matrix, $\mathcal{C} = [c_{ij}]$, the following three quantities are useful in measuring the quality of molecular cogs:

- Forward Cogs Rate:

$$\text{FCR} := \frac{\sum_{i,j \in \text{FC}} c_{ij}}{\sum_{c_{ij} < 0} c_{ij}} \quad (3.5)$$

- Reverse Cogs Rate:

$$\text{RCR} := \frac{\sum_{i,j \in \text{RC}} c_{ij}}{\sum_{c_{ij} > 0} c_{ij}} \quad (3.6)$$

- Gear Grinding Rate:

$$\text{GGR} := \frac{\sum_{\substack{i \in \text{FC} \\ j \in \text{RC} \\ c_{ij} > 0}} c_{ij}}{\sum_{c_{ij} > 0} c_{ij}} + \frac{\sum_{\substack{i \in \text{FC} \\ j \in \text{RC} \\ c_{ij} < 0}} c_{ij}}{\sum_{c_{ij} < 0} c_{ij}} \quad (3.7)$$

We assumed that both positive and negative elements exist in the cooperation matrix, \mathcal{C} , i.e. that $\sum_{c_{ij} < 0} c_{ij} \neq 0$ and $\sum_{c_{ij} > 0} c_{ij} \neq 0$. In cases in which the denominator is null, we substitute the whole fraction by 0.

FCR is the ratio of the contribution captured by the forward cogs to the total forward propensity found in \mathcal{C} . RCR is analogous, and both these measures have a maximum value of 1. Magnitude of misplaced contributions between atoms from different clusters is indicated by GGR, which has a maximum value of 2.

We propose the following molecular cogs quality measure:

$$\text{SCORE}(\text{FC}, \text{RC}, \mathcal{C}) := 0.5(\text{FCR} + \text{RCR} - \text{GGR}), \quad (3.8)$$

and $\text{SCORE} := 0$ for cases when any of the cogs consists of a single atom. The above scoring function was used in our genetic clustering procedure (described in the following section). See Figure 3.2 for an example of a clustering which maximizes the SCORE.

SCORE equals 1 if the corresponding molecular cogs cover the whole propensity of the system to move along a reaction coordinate, and no gear grinding occurs. We observed that a SCORE less than 0.5 often indicates that the subdivision of the system into molecular cogs is noisy.

Note that a trivial partitioning in which all atoms are assigned to one group, e.g. FC, indicates that the system as a whole has a tendency to move forward. We denote such trivial partitionings as “all FC” and “all RC”. The maximal value of SCORE is then 0.5, and although this is unfavorable, in many cases a trivial partitioning had the highest SCORE. In such cases, there was no convenient clustering into two groups, which means that the gear grinding was high.

Genetic clustering algorithm

Genetic algorithms are widely used in optimization problems when potential solutions are readily assessed and codified. They emulate the process of natural selection, promoting solutions – referred to as *specimens* – with higher scores. The codification of a solution is treated as its chromosome, which allows for *mutation* of a solution, and *crossover* with other chromosomes, thus “spawning” new specimens. At the end of each iteration of a genetic algorithm, there is a stage called *selection*, during which low-scoring solutions have a lesser chance of “survival”.

We used the genetic clustering algorithm to find molecular cogs with the highest SCORE, as defined in Equation (3.8) (see [18] for a good introduction to genetic

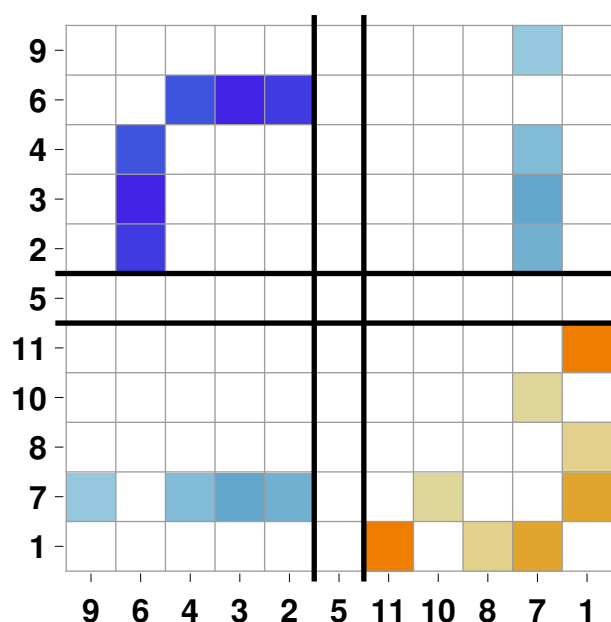


Figure 3.2: **Graphical representation of a result of a clustering procedure.** Rows and columns of the cooperation matrix in Figure 3.1 were reordered according to their assignments to: forward cogs, non-interacting atoms, and reverse cogs. In the upper left corner of this transformed matrix we have a block (square submatrix) with negative contributions, which come from the atoms 2, 3, 4 and 6 (from the forward cogs). In the lower right corner we have a block formed by pair contributions of the atoms in the reverse cogs. The gear grinding is small and results from contributions between the atom 7 and the atoms 2, 3, 4, 9.

clustering). The partitioning of a molecular system into molecular cogs was encoded by an array of numbers from the set $\{-1, 0, 1\}$, where the i th element of the array corresponded to the i th atom in a molecule. Values -1 and 1 translate into assignments to the forward and reverse cogs, respectively. An atom not belonging to any cluster was tagged by 0 ; this occurred when the atom did not cooperate with any other atom (see Figure 3.3).

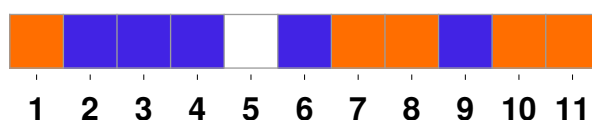


Figure 3.3: **Graphical representation of an exemplary partitioning:** $\{1, -1, -1, -1, 0, -1, 1, 1, -1, 1, 1\}$. Orange squares correspond to 1 (reverse cogs), blue to -1 (forward cogs), and white to 0 (non-interacting). The cooperation matrix in Figure 3.1 was reordered according to this assignment, yielding the transformed cooperation matrix in Figure 3.2.

To initialize the genetic clustering procedure, the AP algorithm was executed by applying five different methods of setting the diagonal elements in the affinity matrix (see *Supporting Information*). Next, it adds the “all FC” and “all RC” trivial solutions to the starting set. Following this, solutions are iteratively chosen, mutated and added to the set, until the starting population contained 200 candidate solutions.

Once the initial set is generated, the genetic procedure repeated the following steps:

1. Compose (randomly) 50 pairs of solutions to generate offspring using the crossover procedure.
2. Select (randomly) 20 solutions to generate offspring using the mutation procedure.
3. Calculate SCORE for the offspring and add it to the population.
4. Draw (randomly) 200 solutions from the population and discard the rest.
5. Choose the best scoring solution and check if there is any improvement in the SCORE. If there was none for 10 consecutive iterations, return the best solution. Otherwise, return to 1.

All random selections are done without repeats, so that a given solution with SCORE s is chosen with a probability proportional to e^{2s} . Although the number of parameters required in the genetic algorithm is daunting, changes to the majority of these parameters only influence the speed of arriving at the optimal solution (see the *Supporting Information* for a more detailed discussion of the parameter’s influence on the outcome).

Figure 3.4 shows that the genetic clustering finds the optimal solution for a range of transient cooperation matrices for *nbd* (complete results can be found in the *Results* section). The optimal solutions were found by means of a brute-force search, i.e. by producing all possible partitionings and calculations of their SCOREs. It is worth noting that in all cases there was a singular solution with the highest SCORE.

3.2.3 Trapezoidal rule for integrating A

The free energy difference between ξ_X and ξ_Y can be expressed as an integral:

$$\Delta A = \int_{\xi_X}^{\xi_Y} A'(\xi^*) d\xi^*. \quad (3.9)$$

The free energy derivative, $A'(\xi^*)$, can be estimated at a given ξ^* from a cMD simulations *via* Equation (3.1). The integral in Equation (3.9) can then be calculated using the trapezoidal rule [62]:

$$\widehat{\Delta A} \approx \frac{\Delta \xi}{2} \left\{ \widehat{A'(\xi_1)} + 2\widehat{A'(\xi_{i+1})} + \dots + 2\widehat{A'(\xi_{M-1})} + \widehat{A'(\xi_M)} \right\}, \quad (3.10)$$

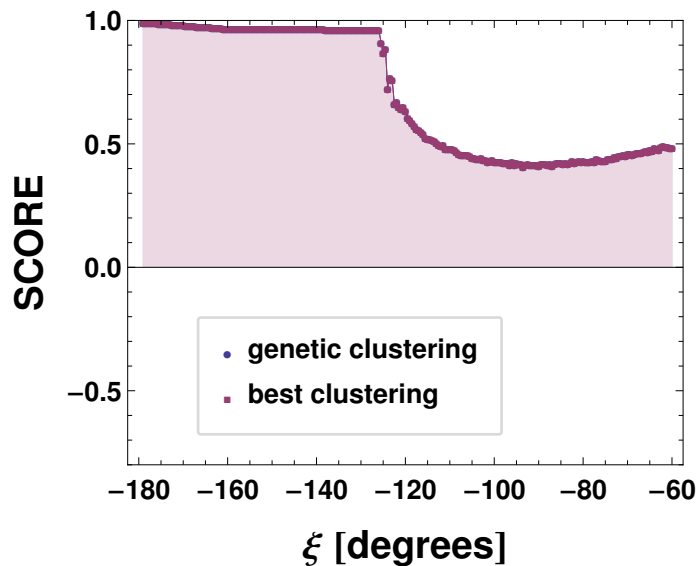


Figure 3.4: **SCORE** plot for *nbd* interactions. The plot shows that solutions found by the genetic clustering algorithm overlap perfectly with the highest scoring assignments to the transient molecular cogs.

where $\widehat{A'(\xi_i)}$ denotes an estimate of the free energy derivative at ξ_i . This requires independent cMD simulations at M values of the collective variable, $\{\xi_i\}_{i=1}^M$, with the grid size of $\Delta\xi$.

Each estimate $\widehat{A'(\xi_i)}$ has a corresponding variance, $\sigma^2[\widehat{A'(\xi_i)}]$, which in turn propagates to the variance of the integral estimate, i.e. $\sigma^2[\widehat{\Delta A}]$. In the following section we explain how we estimated the variances $\sigma^2[\widehat{A'(\xi_i)}]$. Because estimates $\widehat{A'(\xi_i)}$ are independent, the variance of the whole integral follows from Equation (3.10) straightforwardly:

$$\sigma^2[\widehat{\Delta A}] = \frac{\Delta\xi^2}{4} \{ \sigma^2[A'(\xi_1)] + 4\sigma^2[A'(\xi_2)] + \dots + 4\sigma^2[A'(\xi_{M-1})] + \sigma^2[A'(\xi_M)] \} \quad (3.11)$$

Details concerning estimation of the variances $\sigma^2[A'(\xi_1)]$ using a bootstrapping procedure can be found in the *Supporting Information*.

The same integration rule applies to all elements of the transient cooperation matrices. A matrix in which every element is a result of the above numerical integration is referred to as the *integrated cooperation matrix* or simply: *cooperation matrix*, and denoted by $\mathcal{C}(\xi_X \rightarrow \xi_Y)$. The sum $\sum_{\alpha>\beta} [\mathcal{C}(\xi_X \rightarrow \xi_Y)]_{\alpha\beta}$ is the energetic contribution to ΔA for the $\xi_X \rightarrow \xi_Y$ path. Molecular cogs found for this matrix are called *global molecular cogs*.

3.3 Results

In the first part of this section we present detailed results for a small, 11-atom molecular model, to validate the concept of our theoretical approach. We attempted to indicate molecular cogs to verify whether the genetic algorithm finds the optimal clustering for the scoring function proposed in the *Objective function* section.

In the second part of the *Results* we show molecular cogs for the *nbd*, *ele* and *vdw* interactions for three other molecules. These case-studies allowed for testing of the transferability of the parameters used in the genetic clustering algorithm, but also uncovered certain subtleties characteristic to our approach.

3.3.1 The [NH3+]CC(I)I molecular model

We were interested in finding molecular cogs propelling a structural transition between two meta-stable states, separated by a high free energy barrier. For our first case-study we required a system with a fairly natural collective variable, in which all types of interactions are significant (*conf* as well as *nbd*). We used the 2,2-diiodoethan-1-aminium molecule, which in the SMILES format is encoded as: [NH3+]CC(I)I (we use the SMILES representation throughout this article because of its conciseness). The dihedral angle between atoms N1-C5-C6-I7 was our collective variable of choice (see Figure 3.5). We used the Generalized Amber Force Field (GAFF) to model interactions between atoms, with partial charges assigned using an empirical procedure, AM1-BCC (Table 3.1).

[NH3+]CC(I)I		NCC(I)I		CCC(I)I		CClCC(I)I	
atom	partial charge	atom	partial charge	atom	partial charge	atom	partial charge
N1	-0.85	N1	-0.92	C1	-0.10	C1	0.03
H2	0.47	H2	0.36	H2	0.04	Cl2	-0.19
H3	0.47	H3	0.36	H3	0.04	H3	0.08
H4	0.47			H4	0.04	H4	0.08
C5	0.13	C4	0.17	C5	-0.07	C5	-0.08
C6	0.08	C5	0.20	C6	0.21	C6	0.20
I7	-0.08	I6	-0.19	I7	-0.19	I7	-0.19
I8	-0.08	I7	-0.19	I8	-0.19	I8	-0.19
H9	0.13	H8	0.05	H9	0.06	H9	0.08
H10	0.13	H9	0.05	H10	0.06	H10	0.08
H11	0.13	H10	0.12	H11	0.10	H11	0.12

Table 3.1: Partial charges assigned by the AM1-BCC procedure.

We focused on finding molecular cogs propelling the transition between the dihedral angles of $\xi_X := -172.5^\circ$ and $\xi_Y := -62.5^\circ$, which correspond to two PMF minima (Figure 3.5). The free energy barrier separating these minima is high, and interactions between the [NH3+] group and the iodine atoms provide strong *ele* and *vdw* contributions which influence this barrier.

Constrained MD simulations were carried out for fixed values of the ξ collective variable, each simulation with 10^5 fs timesteps, at $T = 300\text{K}$. We chose 239 points,

$\{\xi_i\}_{i=1}^{239}$, equally separated by $\Delta\xi = 0.5^\circ$, so that $\xi_1 = -179.0^\circ$ and $\xi_{239} = -60.0^\circ$. This was done in order to encompass the $[\xi_X, \xi_Y]$ interval; note that $\xi_{14} = \xi_X$ and $\xi_{234} = \xi_Y$.

The length of the block in the block bootstrap estimation of averages and their corresponding variances was set to 10^2 , which led to 10^3 blocks for each simulation data set. We chose the length of the block with the assumption that the auto-correlation after 10^2 steps is negligible in the case of our simple system.

The cMD procedure was implemented in the Python programming language (ver. 2.7.2), using the Open Babel[58] package (ver. 2.3.2) to model the molecule (see *Supporting Information*). All simulations for the $[\text{NH}_3^+]\text{CC}(\text{I})\text{I}$ model took about 30h on a desktop computer.

We focused on the free energy differences and energetic contributions for the $\xi_X \rightarrow \xi_Y$ path. For these end-points we obtain the free energy difference $\Delta A \approx -16.4 \frac{\text{kcal}}{\text{mol}}$, which is close to the energetic contribution, $\Delta E \approx -16.7 \frac{\text{kcal}}{\text{mol}}$. The residual $0.3 \frac{\text{kcal}}{\text{mol}}$ is the entropic contribution, which was negligible for our model.

3.3.2 Overview

Results for the *total*, *conf* and *nbd* interactions are juxtaposed in Table 3.2, and for 2-body interactions (*bond*, *ele*, *vdw*) in Table 3.3. The first row contains optimal partitionings of the integrated cooperation matrices, i.e. global molecular cogs (see Figure 3.3 for explanation), along with a picture of the model colored according to the clustering. In the second row we placed the integrated cooperation matrices rearranged in accordance with the clusterings depicted in the previous row of the table (see Figure 3.2 for explanation). This representation visualizes the “density” of cooperativity within molecular cogs, and gear grinding. The third row shows SCOREs for transient molecular cogs and compares genetic clusterings with optimal, brute-force partitionings. In the next row we see an illustration presenting transient molecular cogs, which is a concise summary of how cooperation within the molecule changes with ξ . This plot is helpful in judging whether the global molecular cogs are similar to the transient molecular cogs, and in assessing the consistency of cooperation within the molecule. Finally, the last row contains a PMF-like contribution profile which we call the *energetic contribution profile*; this is the most important result of our analysis. The green line represents the total energetic contribution of a given interaction type, orange and blue lines show contributions of the reverse and forward cogs, respectively, and the purple line – the magnitude of gear grinding. Gear grinding is quantified as the sum of absolute values of all misplaced contributions, i.e. from pairs of atoms from different clusters. Forward and reverse cogs contributions are calculated as the sum of cooperation between atoms assigned to FC and RC, respectively.

3.3.3 Results for the $[\text{NH}_3^+]\text{CC}(\text{I})\text{I}$ molecular model

In this section we comment on the results presented in Tables 3.2 and 3.3. We explain the Tables row by row, highlighting important aspects of the analysis,

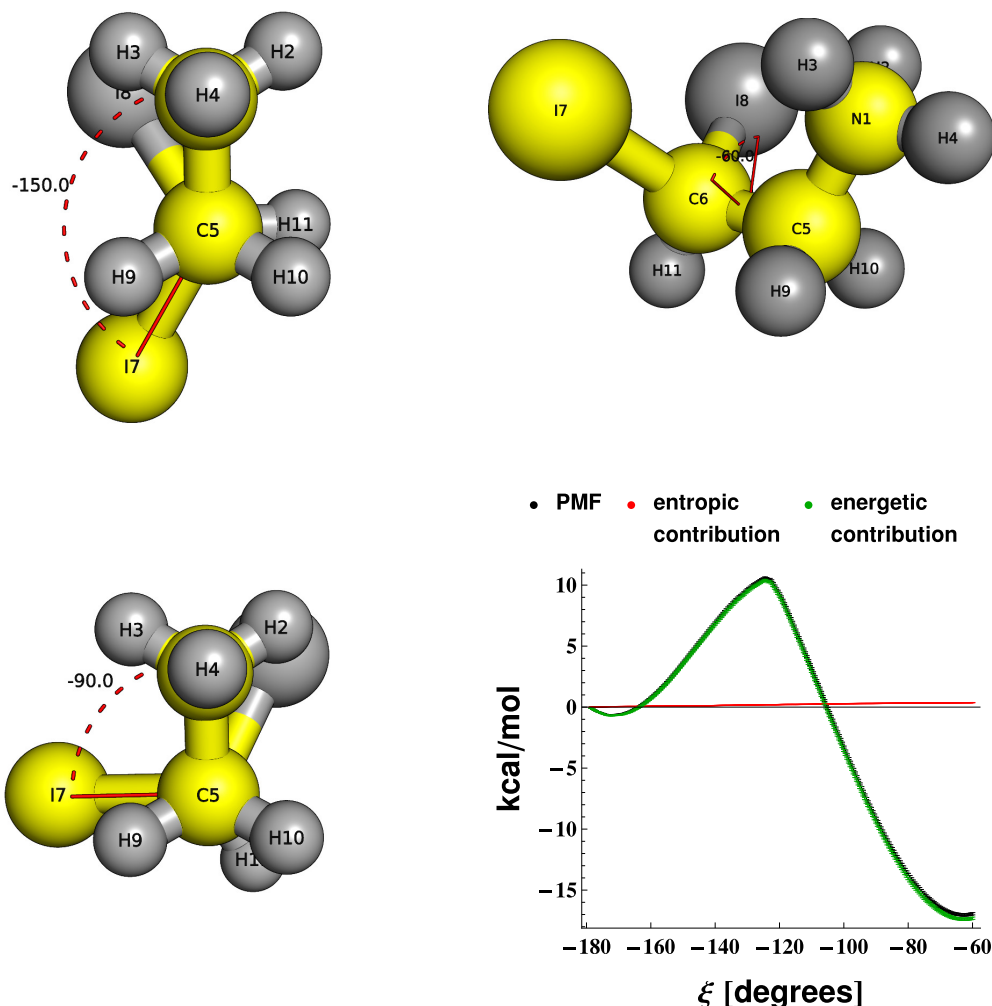


Figure 3.5: $[\text{NH}_3^+]\text{CC}(\text{I})\text{I}$ molecule summary. Three configurations of the molecule are shown, and the PMF with energetic and entropic contributions (for $T = 300$ K). Atoms N1 - C5 - C6 - I7 (yellow) were used to define the dihedral angle, ξ .

starting with Table 3.2 and then proceeding with Table 3.3.

In the first row of Table 3.2, global molecular cogs for *total* and *conf* interactions are trivial (“all FC”), with very low SCOREs. Thus, no interesting cooperation was detected for these interaction types – the whole system has a general propensity for preferring the state around ξ_γ . On the other hand, global molecular cogs for the *nbd* interactions have a high SCORE. The clustering suggests, in particular, that non-bonded interactions between the nitrogen atom and the iodine atoms hinder the transformation (as expected). A more surprising implication was that

the hydrogen atoms from the [NH3+] group cooperate with the C6 atom.

Clustered cooperation matrices are placed in the second row of Table 3.2. For the *total* and *conf* interactions, for which the partitioning was trivial, the matrices kept their original form, whereas for the *nbd* interactions rows and columns were rearranged in accordance with the partitioning. Matrices for *total* and *conf* are almost indistinguishable, which suggests that the transition is mainly governed by the *conf* interactions. It is also clear that the cooperation matrix for *nbd* is less “dense”, with little gear grinding.

In the next row we see that the partitionings into transient molecular cogs for *nbd* have consistently higher SCOREs than those found for *total* and *conf*. There is a drop in SCOREs corresponding to the crossing of the free energy maximum at about -115° dihedral angle. SCOREs for *nbd* molecular cogs vary significantly for different values of the collective variable, and the division into forward and reverse cogs becomes slightly more difficult after crossing the free energy maximum.

The transient molecular cogs depicted in graphs in the fourth row of the table show that the transient molecular cogs for *total* and *conf* exhibit no interesting structure and, in most cases, are of the “all FC” or “all RC” type. Transient molecular cogs for *nbd* are fairly consistent with global molecular cogs. Interestingly, the previously noted *nbd* cooperation between hydrogen atoms H2, H3, H4 and the C6 atom persists throughout the transition.

In the last row of the table we placed the energetic contribution profiles. Because the global molecular cogs for the *total* and *conf* interactions were of the “all FC” form, the only contributions come from the forward cogs. For the *nbd* interactions, the plot shows a beautiful separation of contributions coming from the forward ($-4.2 \frac{\text{kcal}}{\text{mol}}$) and the reverse cogs ($3.7 \frac{\text{kcal}}{\text{mol}}$) for non-bonded interactions, and low gear grinding for the $\xi_X \rightarrow \xi_Y$ transition.

Let us now look at the results in Table 3.3. In all three cases global molecular cogs were non-trivial, although the SCORE for *bond* is significantly lower than for *ele* and *vdw*. The partitioning for *bond* suggests that there is an impediment arising from interactions of the C6, I7 and I8 atoms, whereas the rest of the system favors the state around ξ_Y . Moving on to *ele* and *vdw* interactions we see that their global molecular cogs share a common pattern, although the SCORE for the latter is slightly lower. We can also see that the cooperation of the H2, H3, H4 and C6 atoms (indicated earlier for *nbd*) is caused by the *ele* interactions.

The next row of Table 3.3 shows graphical representations of the clustered cooperation matrices. Not surprisingly, the *bond* matrix is more sparse than any other matrix, however judging by the low SCORE for global molecular cogs, this was not sufficient to ensure a clear division into the forward and reverse cogs. Conversely, for *ele* and *vdw* there seems to be a higher degree of gear grinding, yet the SCOREs were higher. This is due to the fact that the cooperation within molecular cogs is much stronger than gear grinding between them.

SCOREs in the third row of the table show that the transient molecular cogs for the *bond* interactions were consistently low. We see the opposite for *ele*, and a completely different situation for the *vdw* transient molecular cogs. The case of the *vdw* cooperation is particularly interesting because it shows that the global

molecular cogs can have a high SCORE despite the fact that most of the transient molecular cogs have SCOREs below 0.5.

In the fourth row of the table we see that the transient molecular cogs for *bond* are consistent with their global molecular cogs for dihedral angles in the $[-150^\circ, -100^\circ]$ interval (i.e. around the free energy maximum at $\xi \approx -115^\circ$) For the *ele* interactions we see a stable cooperation between the H1, H2, H3 and C6 atoms, occasionally aided by atoms: H9 and I8. The reason behind the shape of the *vdw* SCORE plot becomes clearer once we see that the cooperation for this interaction type has two stages – before and after crossing the free energy maximum. Transient molecular cogs for *vdw* on the left side of the free energy maximum are mainly trivial (“all RC”) with a SCORE of about 0.5. To the right of the maximum there is a change in cooperation; we see a steady partitioning into forward cogs composed of atoms: N1, I7, I8, H9, H10 and H11, and reverse cogs (atoms: H2, H3, H4 and C6).

In the last row of the table we see that the *bond* interactions lead to molecular cogs with high gear grinding, which is the cause of low SCOREs. The profile shows that these interactions lower the free energy gap in the $\xi_X \rightarrow \xi_Y$ transition by $-9.9 \frac{\text{kcal}}{\text{mol}}$. However, we can also see that the contribution from atoms C6, I7 and I8 alone increases this gap by $5 \frac{\text{kcal}}{\text{mol}}$. For the *ele* interactions the separation into molecular cogs was clean (low gear grinding), with a $-4.3 \frac{\text{kcal}}{\text{mol}}$ contribution from the forward cogs, and a $2.8 \frac{\text{kcal}}{\text{mol}}$ contribution from the reverse cogs due to the transition. For *vdw*, the separation has also led to low gear grinding, however the overall contribution from the forward cogs is much smaller than the one from the reverse cogs. This is an important observation which occurs again in the next section, where we analyze global molecular cogs determined for other model systems.

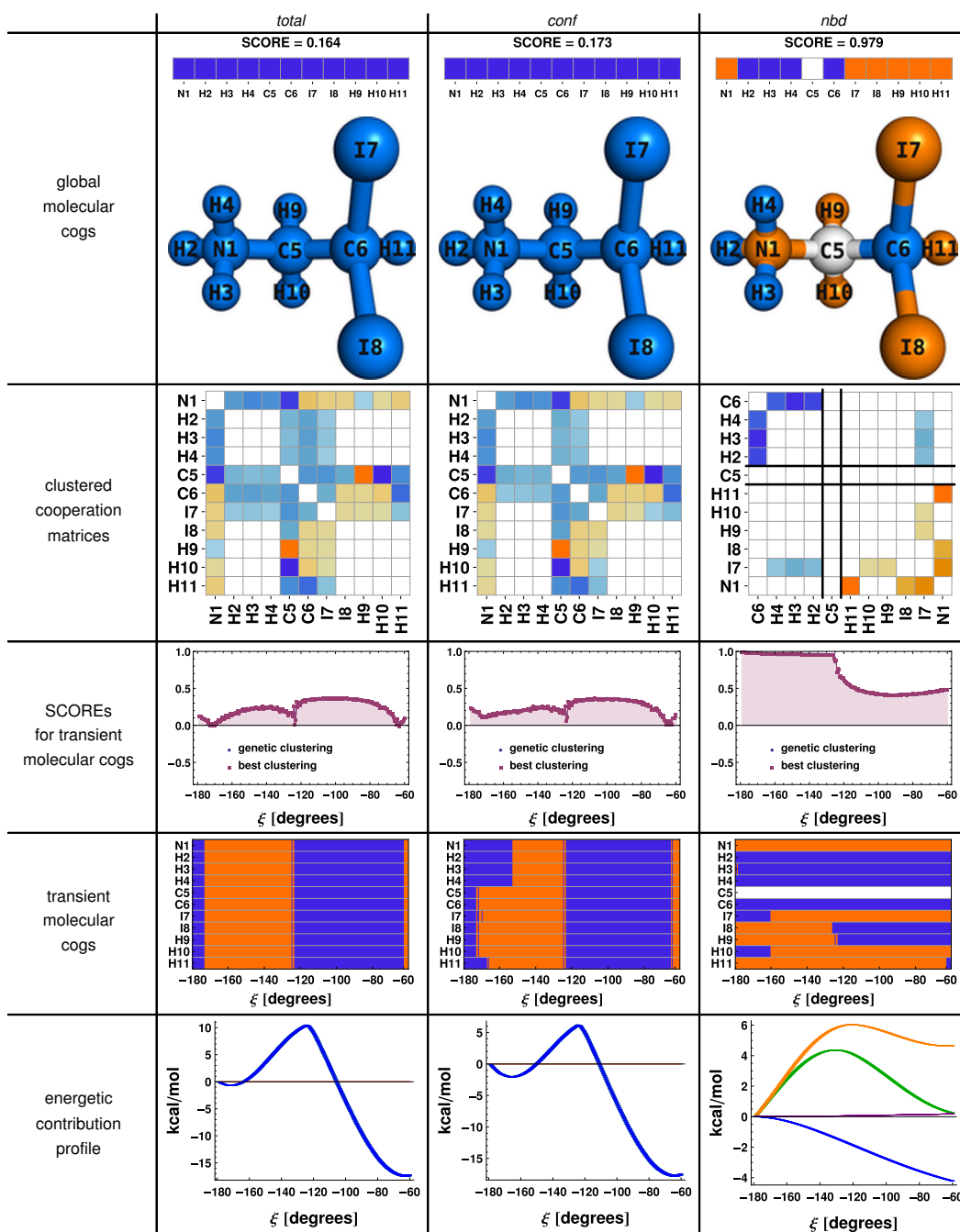


Table 3.2: Results summary for interactions: *total*, *conf* and *nbd*. Out of the triple: *total*, *conf* and *nbd*, only the last one leads to a high SCORE partitioning, yielding non-trivial molecular cogs. Colors: green, blue, orange and purple in the last row of the table denote contributions from the: whole system, forward cogs, reverse cogs and gear grinding, respectively (see the *Overview* section for details).

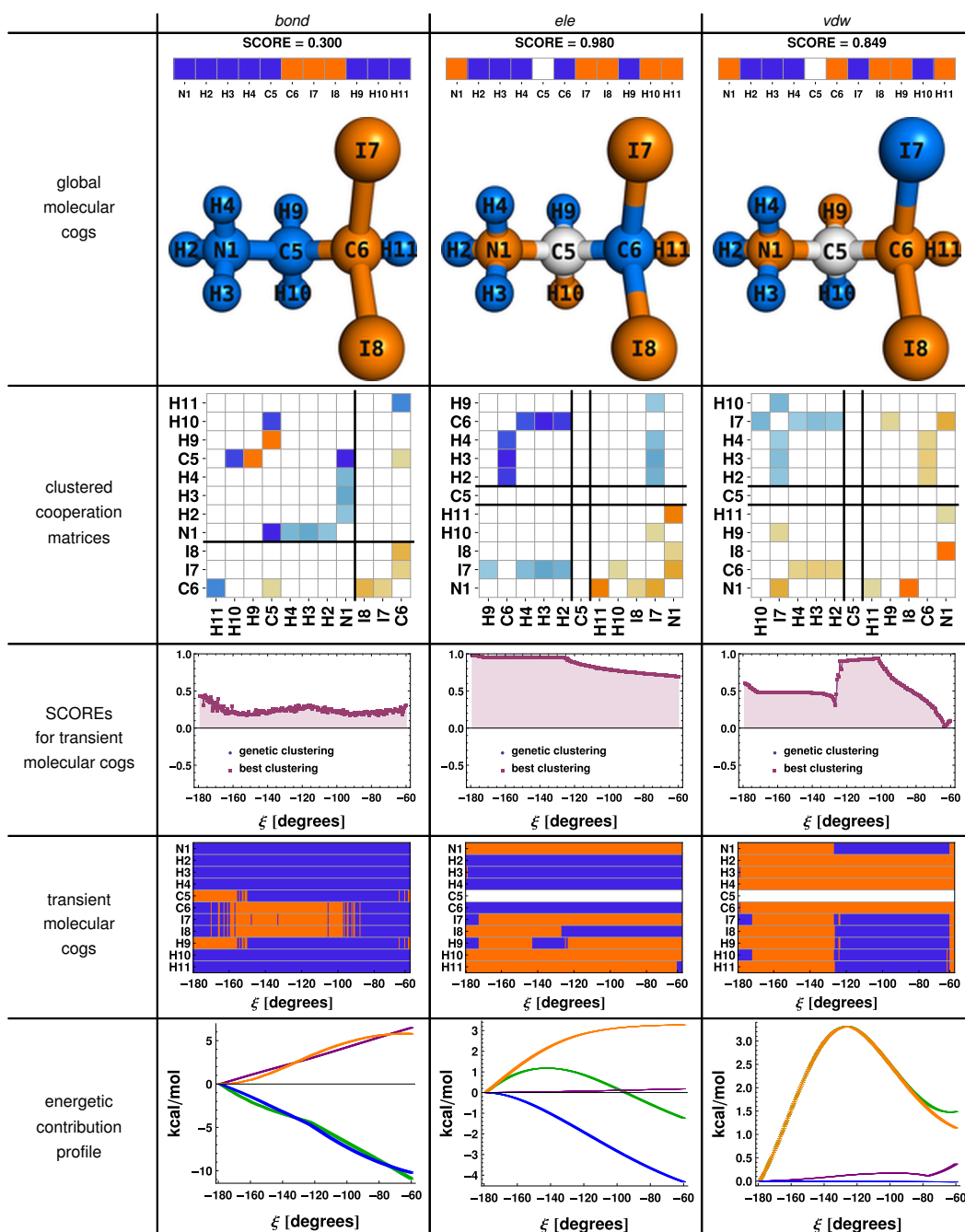


Table 3.3: Results summary for interactions: *bond*, *ele* and *vdw*. All three 2-atom interaction types lead to decompositions into non-trivial molecular cogs. However, the transient molecular cogs have high SCOREs only for the *ele* interactions. Colors: green, blue, orange and purple in the last row of the table denote contributions from the: whole system, forward cogs, reverse cogs and gear grinding, respectively (see the *Overview* section for details).

3.3.4 Results for related molecules

To better understand our approach, it is valuable to identify molecular cogs for other systems, related to the $[\text{NH}_3^+]\text{CC}(\text{I})\text{I}$ molecule scrutinized above. It was also valuable from the perspective of testing the transferability of the parameters used in the genetic clustering algorithm (as shown in the full results in the *Supporting Information*). We took into account molecules which instead of the $[\text{NH}_3^+]$ group had the: NH_2 , CH_3 and CH_2Cl groups respectively, i.e. molecules: $\text{NCC}(\text{I})\text{I}$, $\text{CCC}(\text{I})\text{I}$ and $\text{CClCC}(\text{I})\text{I}$. Partial charges assigned by the AM1-BCC procedure for these molecules are listed in Table 3.1. In this section we report and shortly discuss global molecular cogs for the *nb*, *ele* and *vdw* interactions (see Table 3.4).

Because we analyzed closely related molecules, it was expected that molecular cogs for the *nb* interactions would be comparable. All molecules presented here, except for $\text{CClCC}(\text{I})\text{I}$, have partial charges of the same sign for corresponding atoms, therefore global molecular cogs for *ele* look similar. The more interesting outcome was related to discrepancies in molecular cogs identified for the *vdw* interactions.

As noted earlier, in the case of the $[\text{NH}_3^+]\text{CC}(\text{I})\text{I}$ molecule the overall *vdw* contribution is almost entirely explained by cooperation within the reverse cogs. The forward cogs were identified merely because their contribution was non-positive. Nevertheless, this was a valuable insight – we have learned that the *vdw* steric effects are due to interactions of particular atoms: those which comprise the reverse cogs. On the other hand, the *vdw* molecular cogs for the $\text{NCC}(\text{I})\text{I}$ molecule are trivial (“all RC”) and carry no such information. No non-trivial partitioning with a higher SCORE was found because there were no legitimate forward cogs, even as ineffective as those discovered in $[\text{NH}_3^+]\text{CC}(\text{I})\text{I}$. This then suggests that we gained a simplified picture of *vdw* cooperation in the $[\text{NH}_3^+]\text{CC}(\text{I})\text{I}$ molecule simply because we were fortunate enough to have analyzed a system in which there had been at least a minimal contribution from the forward cogs. This is a consequence of the underlying assumption that a molecule’s tendency to undergo a transition is a result of two opposing cooperations. Perhaps we should approach the problem differently for instances in which the molecule as a whole has the propensity to move forward/backward (as we also saw for the *total* and *conf* interactions).

Molecular cogs for *vdw* interactions for the $\text{CCC}(\text{I})\text{I}$ molecule are similar to those of $[\text{NH}_3^+]\text{CC}(\text{I})\text{I}$. This might seem natural; the $[\text{NH}_3^+]$ and CH_3 groups share common properties. However, the energetic contribution profile for the *vdw* interactions (see *Supporting Information*) reveals that the forward cogs in $\text{CCC}(\text{I})\text{I}$ have a contribution comparable to that of the reverse cogs. Although *vdw* molecular cogs for $\text{CCC}(\text{I})\text{I}$ and $[\text{NH}_3^+]\text{CC}(\text{I})\text{I}$ are identical, the underlying mechanism is different.

The $\text{CClCC}(\text{I})\text{I}$ model was designed to lower the *ele* barrier. However, the resulting molecular cogs for *ele* became trivial and, as in the case of *vdw* for $\text{NCC}(\text{I})\text{I}$, we do not know which atoms are the main source of this effect. This again suggests that perhaps an alternative method of finding molecular cogs should be considered. In cases of trivial partitionings, such method should extract information about parts

of the molecule which are the main source of the free energy difference. However, we leave investigation of properties and characteristics of alternative scoring functions for future studies.

	[NH3+]CC(I)I	NCC(I)I	CCC(I)I	CC1CC(I)I
<i>nb</i>	SCORE = 0.979 	SCORE = 0.980 	SCORE = 0.995 	SCORE = 0.748
<i>ele</i>	SCORE = 0.980 	SCORE = 0.981 	SCORE = 0.979 	SCORE = 0.381
<i>vdw</i>	SCORE = 0.849 	SCORE = 0.493 	SCORE = 0.989 	SCORE = 0.968

Table 3.4: **Global molecular cogs for the following molecules:** [NH3+]CC(I)I, NCC(I)I, CCC(I)I and CC1CC(I)I. We focused our discussion on partitionings into forward and reverse cogs for the *nb*, *ele* and *vdw* interactions. Full results are detailed in the *Supporting Information*.

3.4 Discussion

Note that in all cases the genetic clustering algorithm gave clusterings with the best possible SCOREs. However, the efficiency of the genetic algorithm depends on the starting point, and without the help of the AP-generated initial population it took, on average, about 30 times longer, and the best result was not always achieved. To generate these initial solutions we adapted the AP clustering procedure (see *Supporting Information*).

The reason why conformational interactions lead to low-quality molecular cogs is that these are short-ranged interactions and our test molecular system is small. Partitioning of a graph into clusters is laden with a cost which depends on the weights of cross-cluster edges. But, as can be seen in Table 3.2, the integrated cooperation matrices for non-bonded interactions are more sparse than for conformational interactions. Consequently, the cost of partitioning is greater for more “dense” matrices. But this cost decreases with increasing dimensionality of a matrix. It is therefore possible that, for more complex systems, molecular cogs for conformational interactions will have a higher SCORE.

The non-bonded interactions in the GAFF force field for atoms separated by three bonds or fewer are zero. This, and the fact that our model system is small, led to a sparse *nb* cooperation matrix. Note, however, that a larger system may yield matrices that would be more “dense” for non-bonded interactions than for short-ranged conformational interactions. Whether a partitioning of these matrices would result in higher gear grinding remains an unanswered question.

We also anticipate that the entropic contribution in Equation (3.1) should play a more notable role for larger systems. Nonetheless, we have not yet considered clustering a graph with edges weighted by the entropic contributions of pairs of atoms. The reasons are twofold: practical (we want to avoid calculating Hessians

of collective variables) and conceptual (the entropic cooperation of an atom with itself is non-zero). Entropic contributions need to be considered, but this should be the subject of future studies.

In the second part of the *Results* section we identified and analyzed molecular cogs for three additional molecules, related to the $[\text{NH}_3^+]\text{CC}(\text{I})\text{I}$ model. We discovered that trivial molecular cogs, which carry less information than any other partitioning, may occur whenever there is a lack of competition of cooperation within the molecule. This is a consequence of the proposed approach of dividing the system into two competing subsystems. In many cases in which global molecular cogs were trivial, we simply did not gain any insight into what is propelling the transformation. However, it seems that in such cases the analysis should be aimed at finding parts of a molecule that play the dominant role in its mobility.

Comparison with qualitative expectations

Results for the $[\text{NH}_3^+]\text{CC}(\text{I})\text{I}$ model, especially for the *ele* interactions, were useful in checking our intuitions against what was shown by the analysis. In this section we shortly discuss several sanity checks related to results for the *ele* interactions that were helpful in verifying whether our implementation had any critical errors, and whether the proposed approach leads to reasonable assessments.

From the plot depicting transient molecular cogs for *ele* (Table 3.3) we see that atoms N1 and I7 consistently hinder the transformation. These two atoms have negative partial charges (-0.85 and -0.08 , respectively; see Table 3.1), and therefore repel each other throughout the transition. But the I8 atom (partial charge of -0.08) behaves differently, i.e. for ξ lower than -115° (free energy maximum) it impedes the process by repelling the N1 atom, and aids it for ξ larger than -115° . This effect was correctly captured by the above analysis, because the I8 atom finds itself in the reverse cogs in the first part of the transition, and in the forward cogs in the second part (as shown by the graph in the sixth row of Table 3.3). From the integrated contribution matrix for *ele* (second row in Table 3.3) we see, however, that the cumulative contribution from the N1-I8 atoms is positive, which results in assigning them to one cluster.

The analysis revealed that atoms: H2, H3, H4 and C6 were consistently cooperating electrostatically, lowering the free energy barrier. This conclusion was more unexpected than the one concerning atoms: N1, I7 and I8, but was also compatible with our intuitions, taking into account the partial charges in Table 3.1.

Our method can suggest a qualitative interpretation of the cause behind a structural transition. However, it should be noted the most valuable information gained from this type of analysis is the quantitative description of the molecular cogs *via* the energetic contribution profile.

3.5 Summary of results

The aim of this article was to introduce a new methodology of identifying molecular cogs – parts of a molecule that propel structural transitions in forward/backward

directions along a collective variable. The current framework allowed us to track energetic contributions to the free energy, leaving the problem of including entropic terms for future developments. Results show that with the use of the genetic clustering algorithm we can successfully divide small molecules and identify forward and reverse molecular cogs associated with non-bonded interactions.

In particular, we proposed the approach of defining free energy contributions originating from pairs of atoms, and a method of dividing a molecule into molecular cogs. We showed that the proposed genetic clustering algorithm efficiently finds the optimal cogs, leading to high-quality partitionings for non-bonded interactions. However, we also found that conformational interactions lead to low-scoring molecular cogs, and that the system as a whole favored one meta-stable state over the other.

Currently, our method is based on cMD simulations for computing conditional averages (Equation (3.1)). Unfortunately, this solution has a critical drawback (see *Supporting Information*), but also, in order to determine the entropic contributions, requires computing second-order derivatives of the collective variable. To resolve this problem we should facilitate the Adaptive Biasing Force (ABF) scheme for calculating the PMF [19]. Specifically, our future work is aimed at reformulating the procedure as a plugin to the NAMD package (in which the ABF has already been implemented), to include entropic contributions and to assure scalable performance. Once this is done, numerous new opportunities will become available, some of which we mention below.

We constructed graphs in which nodes corresponded to atoms. Note, however, that they may also be assigned to amino acids (or more sizable objects) to construct graphs, which would lead to a more mosaic clustering. It could also be valuable to represent a whole ligand as a single object in a protein-ligand complex. One could also consider the role of functional groups comprising a ligand and amino acids in a binding pocket; it might then be helpful to represent the rest of the protein as a single node in the graph. Another example is a possible treatment of solvent molecules, for example: a particular group of interesting water molecules could be transformed into separate nodes in the graph, while others reduced to a single node.

Our method may prove helpful in understanding why wild-type proteins perform better than mutants, or in explaining why certain drugs perform better than others, despite their structural similarity. It would also be interesting to consider a multi-stage induced fit docking, and to study cooperativity between amino acids and the ligand along a collective variable.

This research study demonstrates how to extract pair-wise energetic contributions to the free energy change along a reaction coordinate. In its present form¹, we could only use our method to analyze small, model molecules. However, we were able to verify the efficacy of the genetic clustering algorithm, and to learn what can be expected from the new notion of "molecular cogs". Our future aim is to analyze more complex systems, and to develop a publicly available implementation

¹We made the source code available at: <http://github.com/ponadto/molecular-cogs>

of our method for practical applications.

Chapter 4

Conclusions

In this closing chapter we re-iterate the most important findings of the methods presented in Chapters 2 and 3, and also propose several refinements to them. Both studies exemplify projects in which physical knowledge works side by side with the unsupervised machine learning approaches. Physical insight played a predominant role in choosing informative similarity measures used in both applications. Although unsupervised machine learning is becoming more and more powerful, without a reliable similarity measure, even the best clustering algorithms would be ineffective.

4.1 Dynamic domains

ResiCon was built upon several ideas, some of which were essential for its high-quality results. The first one made use of the concept of a *contact* between residues (that had been proven to be advantageous in an earlier application [20]) to define the *geometrical variability*. Thereupon, the second idea was to propose a specific *spectral clustering* algorithm, whose output were clusters corresponding to dynamic domains of a given protein structure. In its current implementation, ResiCon requires information about all pairs of configurations to calculate the geometrical variability.

The similarity measure derived from geometrical variability was intended to express not only structural shifts, but – more importantly – the strength of physical interactions between pairs of amino-acids. This similarity measure was indeed effective, as it led to compact, high-quality dynamic domains, but as we discuss in Chapter 2, its interpretability proved to be especially useful. As we investigated peculiar artifacts in the results of the spectral clustering algorithm, we encountered discontinuities in the clusters. Because of the similarity measure’s interpretability, these “artifacts” lead to valuable observations about the mechanics of structural transitions of the HIV-1 protease.

The spectral clustering method offered a criterion for selecting an optimal number of clusters. One should note that popular methods – GeoStaS and PiSQRD – were vulnerable to the slightest changes in parameter values, often yielding absurd

clusterings, whereas ResiCon was able to discover relevant, rigid domains, or a lack of such. ResiCon is available at <http://dworkowa.imdik.pan.pl/EP/ResiCon>.

Potential improvements

To analyze MD simulations, ResiCon made use of the *principal components analysis* (PCA) to select representative configurations. Applying PCA was fully justified – otherwise, in order to construct the similarity matrix, every pair of configurations in the trajectory would have to be tracked. However, the PCA relies on linearity assumptions, which in case of structural transitions of biomolecular systems may seem overly optimistic. It is possible to amend this obstacle by employing a non-linear algorithm (e.g., NLPCA, which stands for *non-linear PCA*), but at prohibitively high computational cost.

In practice, only a few pairs contribute (or, more precisely: *could* have contributed) to the similarity matrix. It would certainly be worthwhile to test a sub-procedure that might substitute PCA by finding relevant pairs of configurations. In particular, the *locality-sensitive hashing* (LSH) [33] algorithm offers this kind of functionality.

4.2 Molecular cogs

In Chapter 3 we presented a novel methodology for extracting knowledge on internal mechanics of small molecules undergoing structural transitions. This was intended as a “proof of concept”, with a stronger focus on potential capabilities of the proposed methodology and indicating technical problems and pitfalls. We have learned that the similarity measure used for identifying molecular cogs leads to reasonable results, and often highlights unexpected properties of the system at hand. But – perhaps more importantly – the similarity measure has a clear interpretation in terms of free energy change, associated with a structural transition.

Thus, we got a meaningful similarity measure, for which we could apply an adequate clustering method. For that purpose, we proposed a custom objective function, for which we found optimal solutions using a genetic clustering algorithm.

Potential improvements

The fundamental concept of the molecular cogs methodology presented in Chapter 3 was that there are two, *distinct* groups of atoms, cooperating to propel the system forwards (or backwards) along a given collective variable. However, we concluded that the clustering procedure has its limitations in cases in which the molecular system *as a whole* has a general propensity to move in a certain direction. In such cases, a more valuable information should be extracted, namely: which group of atoms plays the dominant role in such an effect.

Instead of using a definitive approach of clustering, one might consider employing a more general idea in which each element can be a member of a more than just one group. *Community detection* offers such functionality, but also allows for

the identification of hierarchical structures. With the help of a community detection algorithm, we could, for example, search for groups of atoms with different degrees of contribution to the system's tendency to make a transition.

From a different stand point, the method presented in Chapter 3 employed *constrained molecular dynamics* (cMD) to estimate local changes in the Helmholtz free energy. However, as we noticed in Chapter 3, the cMD has limitations, when applied to larger systems. An alternative sampling technique – most notably: the *adaptive biasing force* (ABF) method – could remedy this impediment, but at the cost of a more complex and time-demanding implementation. We left the ABF-driven improvement for a future molecular cogs identification scheme.

The ABF sub-procedure would also alleviate the problem of estimating entropic contributions to the molecular cogs. Nevertheless, we are not yet certain how the entropic contribution matrix should be partitioned. We need to re-evaluate the meaning of molecular cogs altogether, and decide on a strategy that would treat entropic and energetic contributions on an equal footing.

The molecular cogs setting in its current formulation suggests a more subtle analysis of the causality hidden in complex molecular transformations. We noticed that by modeling the contributions to the PMF we are close to modeling the kinetics of a transition, as well. That is, we can modify our approach to identify contributions of a given subsystem to the *reaction rate constant*. For that, we would need to equip our analytical toolbox with yet another method called the *Markov state models* (MSM) [67]. Proposed by Frank Noe, the MSM extracts kinetic properties of a system from the PMF. It was proven to give reliable estimates of the reaction rate constant for small enzymes, and seems to be an interesting future extension to our molecular cogs methodology. We might use MSM to discover the influence of a given atomic subset on the kinetics of a structural transition, as well as to identify those subsets whose impact is essential for a given biomolecule's functionality.

4.3 Outcome

This work presented the difficulties and benefits of adapting unsupervised machine learning to analyzing biophysical data. The examples provided in Chapters 2 and 3 showed solutions to three main problems that are likely to be encountered in such investigations:

1. choice / definition of a similarity measure,
2. right clustering procedure and the optimal number of clusters,
3. verification of results.

Our similarity measures were strongly based on physical properties of the analyzed systems. This was crucial as it brought well defined meaning to the resulting partitionings. Next, we chose the appropriate clustering algorithms that worked well with the proposed similarity measures. However, we were not be able to judge

the quality of the clustering (and thus, the clustering procedures) without an external method of verifying the results. Therefore, these three aspects of an analysis involving unsupervised machine learning are interrelated and cannot be considered separately.

Summarizing, in this work we have proven that clustering can be a powerful tool for solving biophysical problems. Chapter 2 describes our procedure for identifying *dynamic domains* – quasi-rigid parts of proteins that undergo structural changes involving relative movements of these parts. In Chapter 3, we presented the concept of *molecular cogs* – an innovative approach to describing and discovering molecular subsystems that determine the propensity of a molecule to go through a conformational transition. In both cases, we were able to transform complex biophysical data into simpler and meaningful sets of structural and functional properties of molecular systems. We hope that these numerical experiments will serve as guideposts for researchers willing to adapt machine learning in their areas of study.

4.4 Final comments

Complex data produced in biophysical simulations and/or experiments (both, *in silico* and *in vivo*) may often contain valuable information whose extraction requires efficient statistical methods. Such information can lead to unexpected results, like for example identification of the “pivot point” residues revealed by the analysis of the HIV-1 protease trajectory presented in Chapter 2. Therefore, it is of paramount importance that such methods should convey easily interpretable, unambiguous, and visually appealing results. These conditions are indispensable for any such method to be useful.

Appendices

Appendix A

Supplementary Materials for Chapter 2

The following text was published verbatim in the Oxford Journal *BIONFORMATICS* in the form of Supplementary Materials [24].

A.1 Hierarchical clustering

We show in the main article that the spectral clustering algorithm employed by ResiCon leads to high-quality results. This algorithm has many advantages, one of which is a versatile internal indicator of clustering quality. This is important because similarity matrices used as input for the clustering procedure differ in size and density. In this paragraph we highlight the problem of choosing the right number of clusters in case of a standard agglomerative hierarchical clustering – UPGMA [71].

UPGMA starts out with all elements in separate groups. The algorithm proceeds iteratively, at each stage joining the two nearest clusters, continuing until there is just a single cluster. In the UPGMA the distance between two sets of points is the arithmetic average of distances between elements from one set and elements from the second. As a result, UPGMA constructs a rooted tree (dendrogram) encoding clusterings into different numbers of clusters. By cutting the tree at a given height h one can retrieve a clustering into groups separated by at least the distance equal to h . Therefore, h can be understood as a parameter determining the number of clusters.

To perform UPGMA clustering we needed a distance matrix, instead of a similarity matrix. A similarity matrix W was transformed into a distance matrix by taking $\mathbf{1} - W$, where $\mathbf{1}$ is a matrix with the same dimensions as W , whose elements are all equal to 1.

In Table A.1 we show how different values of the height parameter lead to different numbers of clusters for the examples used in the main article. The values provided in the table are the minimal heights at which cutting the tree yielded a given number of clusters. Bold font indicates the number of clusters found by Resi-

Con’s spectral algorithm; we used the corresponding values of the height parameter to order the rows in the table.

	Number of clusters					
	1	2	3	4	5	6
2htg	0.907	0.730	0.574	0.494	0.423	0.257
1aey	0.967	0.962	0.933	0.924	0.918	0.898
3mef	0.967	0.950	0.944	0.921	0.914	0.845
1leb	0.974	0.953	0.951	0.935	0.912	0.868
1pkt	0.975	0.958	0.946	0.938	0.931	0.930
1zda	0.976	0.904	0.869	0.865	0.814	0.745
1pit	0.979	0.948	0.948	0.908	0.886	0.871
1aiw	0.983	0.959	0.952	0.912	0.898	0.829
2rgf	0.985	0.983	0.967	0.956	0.953	0.947
2pas	0.985	0.980	0.962	0.958	0.954	0.941
2ktf	0.986	0.977	0.971	0.966	0.964	0.945
2spz	0.986	0.953	0.922	0.904	0.898	0.883
3egf	0.990	0.959	0.938	0.900	0.870	0.868
2vil	0.992	0.986	0.979	0.974	0.970	0.955
2ait	0.992	0.972	0.962	0.949	0.932	0.911
1a67	0.993	0.986	0.969	0.965	0.948	0.948
2pni	0.994	0.991	0.970	0.953	0.947	0.933
2l14	0.997	0.996	0.995	0.992	0.972	0.969
1yug	0.971	0.917	0.916	0.883	0.880	0.861
1vvd	0.996	0.979	0.967	0.952	0.951	0.944
1vve	0.995	0.983	0.973	0.954	0.952	0.942
2k0e	0.999	0.986	0.959	0.935	0.907	0.895
4a5v	0.994	0.986	0.976	0.975	0.973	0.965
1cfc	1.000	0.989	0.988	0.985	0.978	0.969
2kr6	1.000	0.997	0.994	0.991	0.986	0.973
1d1d	1.000	0.998	0.997	0.992	0.986	0.974
2k3c	0.991	0.967	0.892	0.882	0.837	0.760
1adr	0.997	0.972	0.937	0.937	0.912	0.902
1qo6	0.999	0.970	0.964	0.946	0.932	0.920
1bf8	0.999	0.994	0.981	0.981	0.981	0.968

Table A.1: **Heights in the UPGMA dendrogram leading to the provided number of clusters.** The rows are sorted according to the bold values (indicating numbers of clusters as found by ResiCon).

Treating the number of clusters found by the spectral procedure as baseline, there is no value of the height parameter that could reproduce ResiCon’s original results. This is because the relation between the number of clusters and h depends on the size of the distance matrix, and its density. The reason why the parameter h is so volatile is that hierarchical clustering algorithms require a definition of distance between groups (e.g. UPGMA uses the arithmetic average of distances between points, but other measures are also commonly used). If we choose the distance matrix to be $\mathbf{1} - W$, then most of the edges between nodes will be equal to 1. This is the reason why the distance between clusters becomes closer to 1 as their sizes increase. In most cases, we are interested in a fairly small number of

clusters (typically, 3 at most), for which the discrepancies between corresponding h values is high. Without employing a measure of clustering quality we would not be able to straightforwardly choose the optimal number of clusters.

In order to use an agglomerative hierarchical clustering algorithm, it would have been necessary to either abandon the proposed definition of a contact matrix based on geometrical variability, or introduce a more complex transformation leading to the distance matrix. We decided to use a less known, spectral clustering algorithm which overrides the problem of finding optimal numbers of clusters in graphs with different characteristics. We also think that it is a more natural approach in the context of dynamic domains deduction based on a contact matrices.

A.2 Quality analysis for PiSQRD

In the main article we compare results produced by ResiCon, GeoStaS and PiSQRD. We chose PiSQRD as a representative method for the physics-based approaches to dynamic domains identification. Its premise is that the eigenvectors of the structural covariance matrix (Equation A.1), referred to as *low-energy modes*, contain the information required for the identification of global, domain-like movements in a protein.

The structural covariance matrix is defined as:

$$C_{ij} := \langle \delta r_i, \delta r_j \rangle, \quad (\text{A.1})$$

where δr_i is the displacement of the i th coordinate (assuming that a molecule comprising N C^α -atoms is described using $3N$ coordinates), and the angle brackets denote a canonical ensemble averaging. A canonical ensemble average can be approximated, for example, using all-atom NVT molecular dynamics (MD) simulations, i.e. by producing a sample of configurations according to the Boltzmann distribution. It may also be approximated from a single configuration using a physical model, e.g. an elastic network approach, as is being done in PiSQRD.

The input protein models used in our study for method comparison and evaluation were NMR ensembles in the PDB format. In general, these models are *not* a sample from the canonical ensemble of configurations. Additionally, in many cases the coordinates of a model in a PDB file are provided with respect to a reference frame unrelated to the protein's structure. As a result, the average configuration in such cases may be non-physical (improbable due to extremely high energy). Nevertheless, we took an effort of approximating the structural covariance matrix using the models acquired from an NMR experiment and used the eigenvectors extracted from it as input for PiSQRD. The aim of this analysis was to show a potential user (willing to assume that the NMR structures are sufficient to estimate the structural covariance matrix) what can be expected from the results.

In order to produce a single assignment to dynamic domains from a set of structures, the PiSQRD server requires as input: a reference structure, 10 largest eigenvalues of the covariance matrix, and their corresponding eigenvectors. We tested four methods of choosing the reference structure and estimating the structural covariance matrix:

1. take the average configuration (arithmetic average of all coordinates) as the reference structure, and calculate the covariance matrix;
2. take the average configuration as the reference structure, superimposing all models on it, and then calculate the covariance matrix;
3. take the model whose cumulative RMSD to all other models is the smallest as the reference structure, superpose all models on it, and then calculate the covariance matrix;
4. use the mRMSD algorithm (see below) to structurally align all models, take the average configuration as the reference structure, and then calculate the covariance matrix.

In all cases superpositions were done with the Kabsch algorithm, using only the C^α -atoms. The displacements δr_i needed for the calculations of the covariance matrices were calculated with respect to the reference structures.

The first method makes sense in case of a sample of structures produced in the course of a MD simulation, where the reference frame is set so that the momentum and angular momentum are both null. The second method was assumed to work better in cases in which the reference frame with respect to which the coordinates are given in a PDB file is not related to the protein structure. In such cases, the orientation and relative position of the models seems random. The third method guarantees that the reference structure is physical, and that the displacements δr_i are reasonable. The fourth method is an improvement over the second method, in which we use the mRMSD algorithm to structurally align all models.

The mRMSD algorithm is an iterative procedure in which:

1. superimpose all models onto the average structure;
2. compute the new average structure;
3. calculate the RMSD of new average structure with respect to the old average structure;
4. if the RMSD is greater than 0.01 go back to point 1.

For all four methods we extracted the eigenvectors of the covariance matrix and uploaded them along with the reference structures as input for the PiSQRD server. Figure A.1 summarizes the quality of dynamic domains found using PiSQRD, and ResiCon. It is difficult to choose the best method of producing input for PiSQRD based solely on the dynamic domains quality. We, therefore, decided to pick the third method as worth mentioning in the main article because, in our opinion, it is the most natural one, and because the reference structures are in all cases physically viable.

A.3. PROCEDURE FOR SELECTING REPRESENTATIVE CONFIGURATIONS FROM A TRAJECTORY

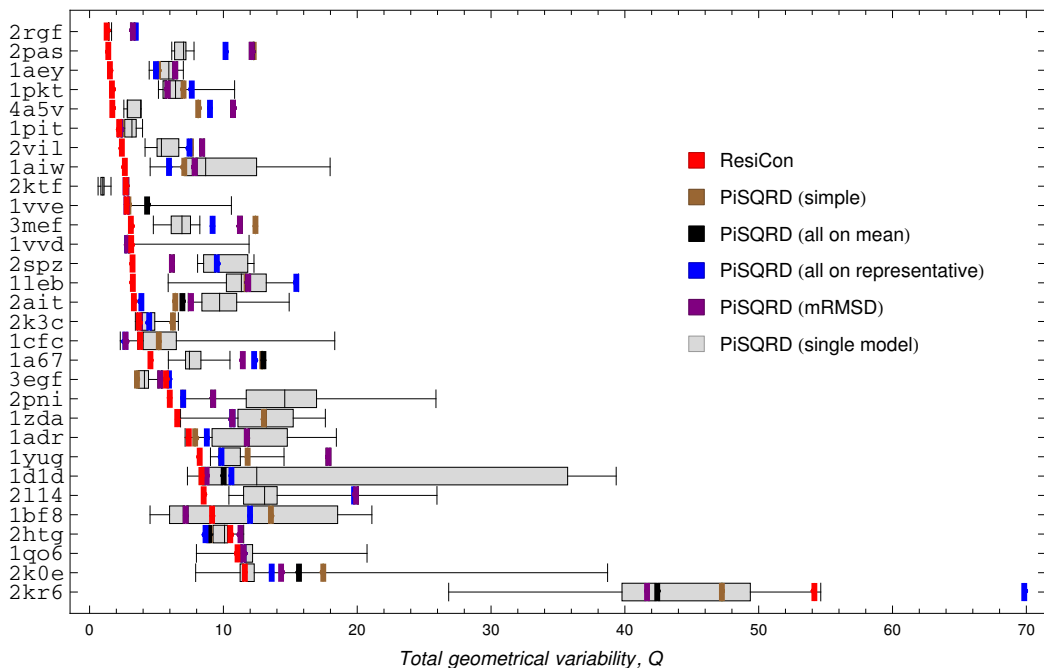


Figure A.1: **Dynamic domains quality for ResiCon and PiSQRD.** We tested the single dynamic domain assignment produced by PiSQRD for four methods of estimating the structural covariance matrix.

A.3 Procedure for selecting representative configurations from a trajectory

We needed to extract an ensemble of representative configurations of the HIV-1 protease from its MD trajectory encoded in a DCD file. For this purpose we have developed a greedy algorithm. Greedy approaches are fairly common whenever there is a need to extract a subset of representative examples from a larger set. In this particular case, however, we wanted to identify exactly m representatives at which spheres of radius R would be centered, thus excluding neighbors of these representatives. Therefore, we had to identify the appropriate radius which would cause the greedy search to yield the required number of samples. We assumed that the number of samples is an “almost monotonous” function of the sphere radius, and decided to use a “softened” binary search method (see Algorithm 1). This heuristic turned out to be effective for the trajectory being analyzed.

Function `ExtractRepresentativeConfigurations` (Algorithm 1) carries out the aforementioned procedure in the following steps. First, we reduce the dimensionality of the configurations in X using Principal Component Analysis (PCA) with p components, yielding a set of points X_p centered around 0. In the case of the HIV-1 protease we used $p = 4$ retaining 90% of variability. Next, the `ChooseConfigurations` procedure (Algorithm 2) selects a set of representative points for a given minimal distance between representatives. The `ExtractRepresentativeCon-`

Algorithm 1: ExtractRepresentativeConfigurations

Input:

X – a set containing N configurations (points in \mathbb{R}^{3n} , where n is the number of atoms)

p – number of principal components

m – target number of representative configurations ($m < N$)

Result: ensemble of m configurations

begin

$PCs \leftarrow$ PrincipalComponents(X)

$X_p \leftarrow$ Projection(X, PCs, p) /* X_p is a set of N p -dimensional points centered around 0 */

$R_{\min} \leftarrow 0$

$R_{\max} \leftarrow \max_{x \in X_p} \|x\|$

$Conf \leftarrow \emptyset$ /* set of representative configurations */

while $|Conf| \neq m$ **do**

$R \leftarrow (R_{\min} + R_{\max})/2$

$Conf \leftarrow$ ChooseConfigurations(R, X_p)

$k \leftarrow |Conf|$

if $|Conf| > m$ **then**

$R_{\min} \leftarrow R_{\min} + (R - R_{\min})/4$

end**if** $|Conf| < m$ **then**

$R_{\max} \leftarrow R_{\max} - (R_{\max} - R)/4$

end**end****end**

return ExtractConfigurations($X, Conf$)

figurations procedure iteratively narrows the upper and lower bounds on the sphere radius, until the number of representative points found by ChooseConfigurations-Greedily is equal to the desired ensemble size m . We used a “softened” binary search for selecting the radius, which narrowed the search area exponentially by a factor of $1/4$.

The ChooseConfigurations procedure is defined in Algorithm 2. The procedure start off by initializing sets Rep and Rem , containing representative and remaining configurations respectively. Next, the algorithm iteratively select a random point x from the Rem set, center a sphere of radius R at that point and finds neighboring points within that sphere. Element x is then added to the Rep set, and the set of neighboring points is subtracted from the Rem set. These steps are repeated until there are no more elements in Rem set, and return Rep as output.

Algorithm 2: ChooseConfigurations

Input: X_p – set of N p -dimensional points R – radius of spheres built around representative points**Result:** set of representative points**begin**

```

   $Rep \leftarrow \emptyset$  /* set of representative points          */

```

*/

```

   $Rem \leftarrow X_p$  /* set of remaining points              */

```

*/

```

  while  $|Rem| > 0$  do

```

```

     $x \leftarrow$  any  $y \in Rem$ 

```

```

     $Neighbors \leftarrow \{y \in Rem : \|x - y\| \leq R\}$ 

```

```

    /* a sphere of radius  $R$  is centered at point  $x$           */

```

*/

```

     $Rem \leftarrow Rem \setminus Neighbors$ 

```

```

     $Rep \leftarrow Rep \cup \{x\}$ 

```

```

  end

```

```

  return  $Rep$ 

```

```

end

```

We have implemented the above procedures in the R language, using `bio3d` package (v. 1.1-6). The principal component analysis was carried out with the `pca.xyz` function (provided in the `bio3d` package), which by default centers and scales configurations.

A.4 Complete set of results

In order to explain our procedure of comparing results of methods tested we discuss as an example the 1d1d protein molecule. To quantify the agreement between dynamic domains assignments, we used the Variation of Information metric (\mathcal{VI} , see *Methods*). In the case of 1d1d ResiCon and GeoStaS exhibit a relatively large value (1.47) of \mathcal{VI} (see Figure A.2), which means that they produced dissimilar partitionings.

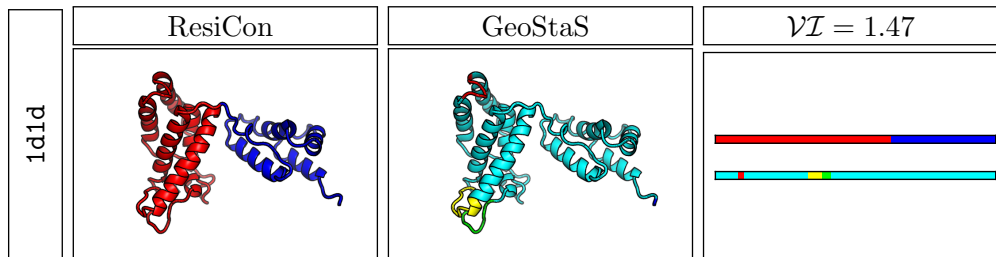


Figure A.2: Dynamic domains assignments by ResiCon and GeoStaS, and their agreement for 1d1d.

Note that GeoStaS identified the C-terminus and a loop as separate dynamic domains because of their flexibility. However, GeoStaS failed to identify the relative motion of the two large subunits of the protein. ResiCon found dynamic domains according to that motion, thus dynamic domains are larger.

In the case of ResiCon and GeoStaS all structures in a PDB file are considered and a single clustering into dynamic domains is being produced. However, PiSQRD analyzes only one (the first) model from several available in the NMR ensemble. It might seem that by requiring only a single structure, PiSQRD has an advantage over methods like ResiCon or GeoStaS. But this would be the case only if PiSQRD gave the same results, regardless of the configuration provided as input. As we show below, dynamic domains assigned by PiSQRD vary significantly.

Because of a large number of models it was not possible to provide a graphical representation for each partitioning. We therefore focused on values of the agreement measure \mathcal{VI} between results produced by the three methods. The histograms of \mathcal{VI} for 1d1d are presented in Figure A.3.

Results presented in the following tables are presented in the same order as in the main article, i.e. best-scoring results are first.

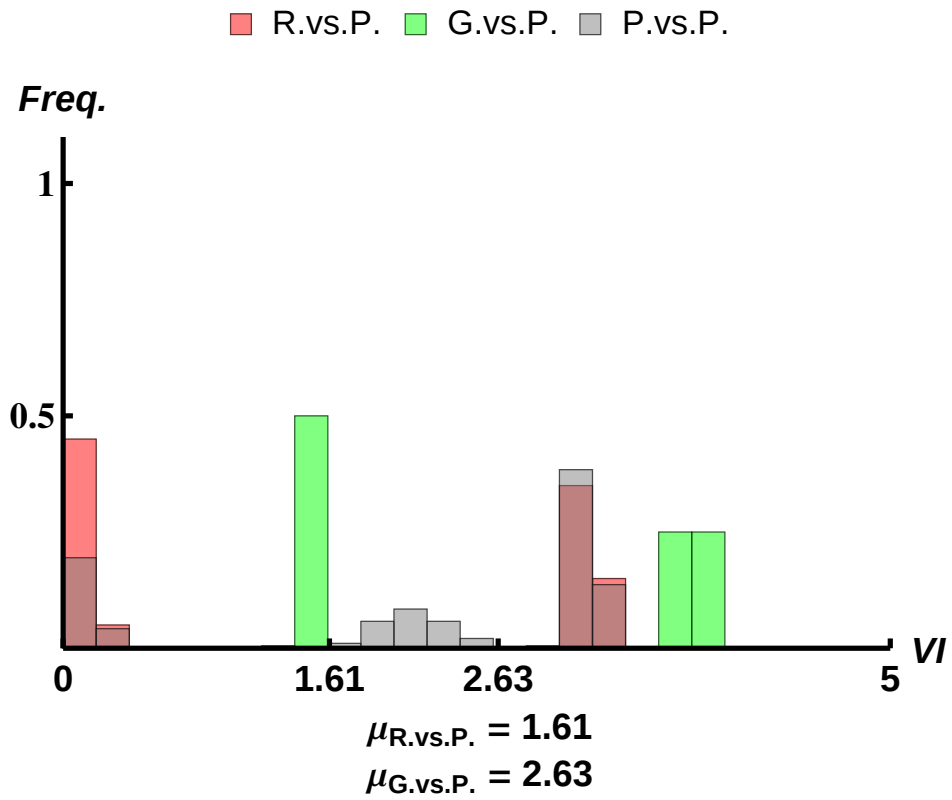


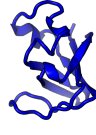




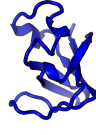




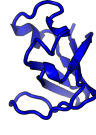




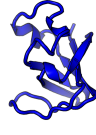




Figure A.3: Histogram of \mathcal{VI} between ResiCon and PiSQRD (R. vs. P.), GeoStaS and PiSQRD (G. vs. P.), and PiSQRD with itself (P. vs. P.) for 1D1D.

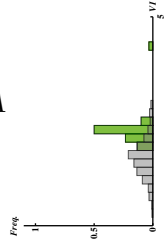
2rgf		R	G	P1	P2	ResiCon	GeoStas	P1:SQRD (P1)	P1:SQRD (P2)	
R		0	0.62	0.45	0.57					
G		0.62	0	0.32	0.26					
P1		0.45	0.32	0	0.21					
P2		0.57	0.26	0.21	0					

2pas		R	G	P1	P2	ResiCon	GeoStas	P1:SQRD (P1)	P1:SQRD (P2)	
R		0	1.94	2.39	2.45					
G		1.94	0	2.00	2.08					
P1		2.39	2.00	0	1.68					
P2		2.45	2.08	1.68	0					





















1aey		R	G	P1	P2	ResiCon	GeoStas	P1:SQRD (P1)	P1:SQRD (P2)	
R		0	0.00	1.90	2.10					
G		0.00	0	1.90	2.10					
P1		1.90	1.90	0	1.29					
P2		2.10	2.10	1.29	0					

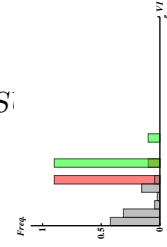
1pkt		R	G	P1	P2	ResiCon	GeoStaS	PiSQRD (P1)	PiSQRD (P2)
R		0	0.00	4.20	1.85				
G		0.00	0	4.20	1.85				
P1		4.20	4.20	0	2.87				
P2		1.85	1.85	2.87	0				





















A

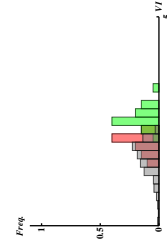


ETE SET OF RES

4a5v		R	G	P1	P2	ResiCon	GeoStaS	PiSQRD (P1)	PiSQRD (P2)
R		0	1.00	1.05	1.59				
G		1.00	0	1.53	2.07				
P1		1.05	1.53	0	1.00				
P2		1.59	2.07	1.00	0				



1pit		R	G	P1	P2	ResiCon	GeoStaS	PiSQRD (P1)	PiSQRD (P2)
R		0	1.59	2.14	2.06				
G		1.59	0	2.70	2.98				
P1		2.14	2.70	0	2.20				
P2		2.06	2.98	2.20	0				



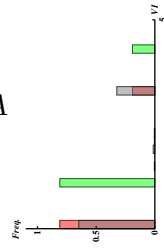
77

2v11		R	G	P1	P2	ResiCon	GeoStas	P1:SQRD (P1)	P1:SQRD (P2)	
R		0	1.89	1.19	1.66					
G		1.89	0	2.43	2.49					
P1		1.19	2.43	0	0.91					
P2		1.66	2.49	0.91	0					

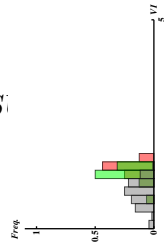
1aiw		R	G	P1	P2	ResiCon	GeoStas	P1:SQRD (P1)	P1:SQRD (P2)	
R		0	1.64	4.24	1.46					
G		1.64	0	3.76	2.64					
P1		4.24	3.76	0	3.24					
P2		1.46	2.64	3.24	0					

2kttf		R	G	P1	P2	ResiCon	GeoStas	P1:SQRD (P1)	P1:SQRD (P2)	
R		0	0.80	1.68	2.10					
G		0.80	0	1.82	2.37					
P1		1.68	1.82	0	1.77					
P2		2.10	2.37	1.77	0					

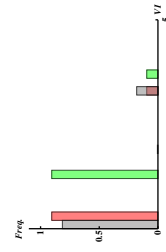
1vve		R	G	P1	P2	ResiCon	GeoStaS	PiSQRD (P1)	PiSQRD (P2)	A ETE SET OF RES
R		0	1.05	0.00	3.27					
G		1.05	0	1.05	4.25					
P1		0.00	1.05	0	3.27					
P2		3.27	4.25	3.27	0					








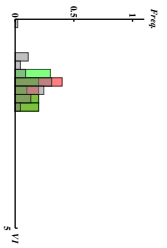



3mef		R	G	P1	P2	ResiCon	GeoStaS	PiSQRD (P1)	PiSQRD (P2)	A ETE SET OF RES
R		0	0.32	1.13	0.61					
G		0.32	0	1.02	0.66					
P1		1.13	1.02	0	0.72					
P2		0.61	0.66	0.72	0					



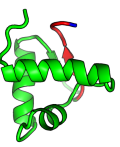

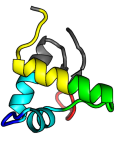
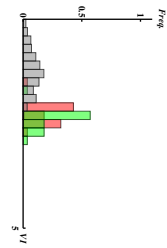







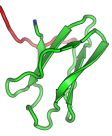


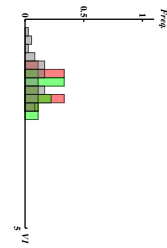



1vvd		R	G	P1	P2	ResiCon	GeoStaS	PiSQRD (P1)	PiSQRD (P2)	A ETE SET OF RES
R		0	1.24	0.21	3.36					
G		1.24	0	1.26	3.74					
P1		0.21	1.26	0	3.28					
P2		3.36	3.74	3.28	0					



79

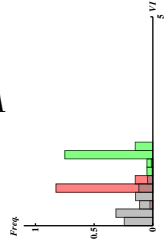
2spz		R	G	P1	P2	ResiCon	GeoStas	PiSQRD (P1)	PiSQRD (P2)	r_{res}
R		0	0.13	1.42	1.94					
G		0.13	0	1.37	1.96					
P1		1.42	1.37	0	2.05					
P2		1.94	1.96	2.05	0					

1leb		R	G	P1	P2	ResiCon	GeoStas	PiSQRD (P1)	PiSQRD (P2)	r_{res}
R		0	0.61	2.47	2.43					
G		0.61	0	2.60	2.61					
P1		2.47	2.60	0	2.25					
P2		2.43	2.61	2.25	0					

2ait		R	G	P1	P2	ResiCon	GeoStas	PiSQRD (P1)	PiSQRD (P2)	r_{res}
R		0	0.71	2.16	1.97					
G		0.71	0	2.07	2.22					
P1		2.16	2.07	0	2.09					
P2		1.97	2.22	2.09	0					

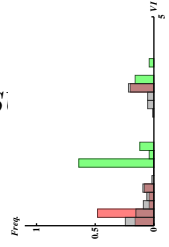
2k3c		R	G	P1	P2	ResiCon	GeoStaS	PiSQRD (P1)	PiSQRD (P2)
R		0	1.86	1.13	1.08				
G		1.86	0	1.35	1.87				
P1		1.13	1.35	0	1.52				
P2		1.08	1.87	1.52	0				

A

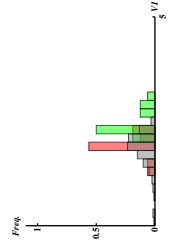


ETE SET OF RES






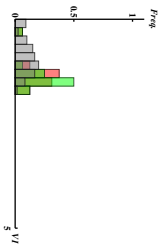



1cfc		R	G	P1	P2	ResiCon	GeoStaS	PiSQRD (P1)	PiSQRD (P2)
R		0	1.40	0.21	3.29				
G		1.40	0	1.48	3.56				
P1		0.21	1.48	0	3.32				
P2		3.29	3.56	3.32	0				






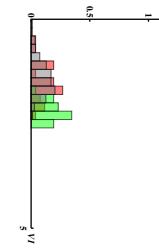







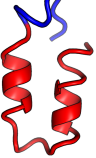
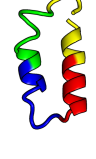

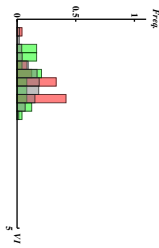



1a67		R	G	P1	P2	ResiCon	GeoStaS	PiSQRD (P1)	PiSQRD (P2)
R		0	1.07	1.94	2.29				
G		1.07	0	2.22	2.69				
P1		1.94	2.22	0	2.59				
P2		2.29	2.69	2.59	0				

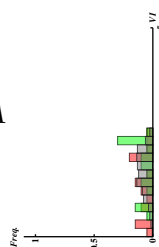


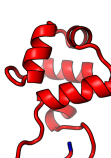
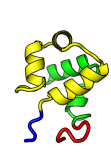













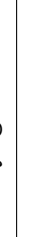
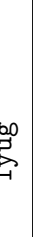
81

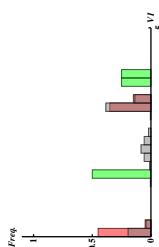







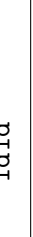
3e6gf		R	G	P1	P2	ResiCon	GeoStas	P1:SQRD (P1)	P1:SQRD (P2)	$\frac{F_{res}}{s}$
R		0	0.14	1.70	1.17					
G		0.14	0	1.70	1.14					
P1		1.70	1.70	0	1.70					
P2		1.17	1.14	1.70	0					

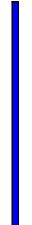




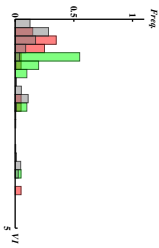

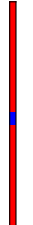

2pni		R	G	P1	P2	ResiCon	GeoStas	P1:SQRD (P1)	P1:SQRD (P2)	$\frac{F_{res}}{s}$
R		0	1.60	2.15	1.62					
G		1.60	0	2.14	2.25					
P1		2.15	2.14	0	2.23					
P2		1.62	2.25	2.23	0					

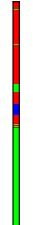




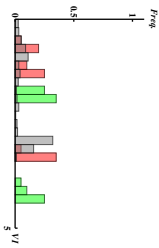



1zda		R	G	P1	P2	ResiCon	GeoStas	P1:SQRD (P1)	P1:SQRD (P2)	$\frac{F_{res}}{s}$
R		0	0.79	1.99	1.81					
G		0.79	0	2.25	1.36					
P1		1.99	2.25	0	2.29					
P2		1.81	1.36	2.29	0					

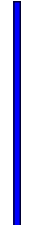

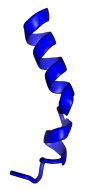


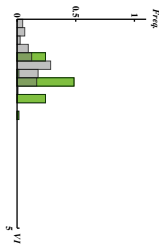
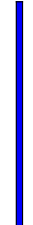

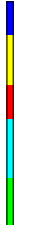
1adr		R	G	P1	P2	ResiCon	GeoStaS	PiSQRD (P1)	PiSQRD (P2)	A <i>ETE SET OF RES</i> 
R		0	0.74	0.82	2.27					
G		0.74	0	1.56	2.34					
P1		0.82	1.56	0	2.57					
P2		2.27	2.34	2.57	0					

1yug		R	G	P1	P2	ResiCon	GeoStaS	PiSQRD (P1)	PiSQRD (P2)	
R		0	1.82	1.07	1.16					
G		1.82	0	2.56	1.57					
P1		1.07	2.56	0	1.80					
P2		1.16	1.57	1.80	0					

1d1d		R	G	P1	P2	ResiCon	GeoStaS	PiSQRD (P1)	PiSQRD (P2)	
R		0	1.47	3.29	0.21					
G		1.47	0	3.76	1.51					
P1		3.29	3.76	0	3.17					
P2		0.21	1.51	3.17	0					

2114		R	G	P1	P2	ResiCon	GeoStas	P1:SQRD (P1)	P1:SQRD (P2)	F_{res}
R		0	0.55	0.31	4.01					
G		0.55	0	0.84	3.66					
P1		0.31	0.84	0	3.74					
P2		4.01	3.66	3.74	0					

1bf8		R	G	P1	P2	ResiCon	GeoStas	P1:SQRD (P1)	P1:SQRD (P2)	F_{res}
R		0	1.60	0.57	3.33					
G		1.60	0	1.69	4.26					
P1		0.57	1.69	0	3.33					
P2		3.33	4.26	3.33	0					

2htg		R	G	P1	P2	ResiCon	GeoStas	P1:SQRD (P1)	P1:SQRD (P2)	F_{res}
R		0	0.00	1.56	2.29					
G		0.00	0	1.56	2.29					
P1		1.56	1.56	0	1.68					
P2		2.29	2.29	1.68	0					

1qo6		R	G	P1	P2	ResiCon	GeoStaS	PiSQRD (P1)	PiSQRD (P2)	<p>A</p> <p>ETE SET OF RES</p>
R		0	0.96	3.07	0.61					
G		0.96	0	3.09	0.62					
P1		3.07	3.09	0	3.12					
P2		0.61	0.62	3.12	0					

2k0e		R	G	P1	P2	ResiCon	GeoStaS	PiSQRD (P1)	PiSQRD (P2)	
R		0	0.97	0.50	3.29					
G		0.97	0	1.27	2.76					
P1		0.50	1.27	0	3.27					
P2		3.29	2.76	3.27	0					

2kr6		R	G	P1	P2	ResiCon	GeoStaS	PiSQRD (P1)	PiSQRD (P2)	
R		0	0.78	3.21	0.30					
G		0.78	0	3.10	0.89					
P1		3.21	3.10	0	3.25					
P2		0.30	0.89	3.25	0					

Appendix B

Supplementary Materials for Chapter 3

The following text was published verbatim in the *Journal of Computational Chemistry* in the form of Supplementary Material [25].

B.1 Constrained molecular dynamics

In the blue moon ensemble method [13], the conditional expected value of quantity f , given $\xi = \xi^*$ can be computed using the *Fixman correction*:

$$\langle f \rangle_{\xi^*} = \frac{\langle m_{\xi}^{1/2} f \rangle_{\xi^*, \dot{\xi}},}{\langle m_{\xi}^{1/2} \rangle_{\xi^*, \dot{\xi}}}, \quad (\text{B.1})$$

where $\langle \cdot \rangle_{\xi^*, \dot{\xi}}$ denotes an average at $\xi = \xi^*$, and $\dot{\xi} = 0$. Also, one assumes that $\nabla \xi \neq 0$. This average is approximated *via* a constrained Molecular Dynamics (cMD) simulation, i.e. by applying an additional force proportional to $\nabla \xi$ that holds the system at ξ^* . Thus, the simulation is carried out using the modified Hamiltonian:

$$\mathcal{H}_{\xi^*} := \mathcal{H} + \lambda(\xi^* - \xi),$$

where λ is chosen such that $\xi = \xi^*$ and $\dot{\xi} = 0$:

$$\lambda = m_{\xi} \left(\sum_i \frac{1}{m_i} \frac{\partial \xi}{\partial \mathbf{x}_i} \frac{\partial U}{\partial \mathbf{x}_i} - \mathbf{v} \cdot \mathbb{H} \mathbf{v} \right).$$

The Andersen thermostat [59] for standard NVT sampling was implemented in the Python programming language, and the Open Babel package (ver. 2.3.2) to model the molecule.

However, we observed that rotations around the N1--C5 bond were much rarer than in a standard NVT simulation. Consequently, the interactions of hydrogens from the [NH3+] with the rest of the molecule were inequivalent, and thus the clustering procedures assigned these hydrogens to different clusters. Consequently,

although the hydrogens from the [NH3+] group should have been equivalent, their interactions with the rest of the molecule were different.

To streamline the sampling, we added a multidimensional replica-exchange scheme [72] to the Andersen thermostat. We initiated each simulation with 3 models, with the dihedral angle C6-C5-N1-H2 set to three different values (spanning the range of 360° evenly), all at the same temperature $T = 300$ K. The frequency of replica swapping was set to 1 in every 10^3 steps of the cMD simulation, for randomly chosen pairs of copies of the molecule. If the swapping was accepted, the system was equilibrated for 10^2 steps, keeping $\xi = \xi^*$ constant.

Sampling of the troublesome degree of freedom associated with rotations around the N1--C5 bond required additional care, but was fairly simple. However, as was noted by Darve in [75], sampling at $\xi = \xi^*$ using cMD may be, in general, impractical because energy barriers separating transitional pathways become virtually impassable. Therefore, cMD simulations in a more general setting are likely to be prohibitively complicated. This drawback is the main disadvantage of our current molecular cogs finding procedure. We believe, however, that the Adaptive Biasing Force [75] should help alleviate the limitations of cMD, and adapting this methodology to our approach should be the natural future approach in finding molecular cogs in more complex systems.

B.2 Error estimation using a bootstrapping procedure

The conditional expected value of quantity f given $\xi = \xi^*$, is estimated from a cMD simulation, as suggested by Equation B.1. If the data points were uncorrelated, one could estimate $\langle f \rangle_{\xi^*}$ simply by taking a mean over the data. Then, assuming a Gaussian distribution, one might approximate the error of this estimate by the standard deviation. An alternative approach, called *bootstrapping*, does not require any assumptions about how the data points are distributed.

Bootstrapping allows estimation of a given statistical quantity (e.g. average) along with its accuracy measure (e.g. variance). Given a set of points $S := \{x_i\}_{i=1}^K$, new sets $S_j := \{x_i^j\}_{i=1}^K$ are resampled from S with replacement. The basic idea of bootstrapping is to calculate a given statistical quantity independently for data sets $\{S_j\}_{j=1}^K$, and, based on these results, extract the measure of accuracy. In its original form proposed by Efron et al. [26], data points in set S are assumed to be uncorrelated. However, consecutive data points acquired from a cMD simulation are correlated. We, therefore, applied a *block bootstrap*, an extension of the original bootstrapping procedure, in which correlation within the data is allowed and accounted for [12]. The block bootstrap divides the set S into b non-overlapping blocks¹ of length k , such that $bk = K$. The i th block, B_i , contains b consecutive data points, such that $B_i = (x_{(i-1)k+1}, \dots, x_{ik})$, for $1 \leq i \leq b$. Next, the procedure draws randomly b blocks with replacement, combines them into a new data set and with its use calculates a given statistical quantity. This procedure is repeated

¹This is the *Carlstein's blocking rule*; for a review of this and other methods see [39, 50].

many times, yielding an array of estimates from which measures of accuracy are inferred.

In our case, the statistical quantity of interest was the average over sets of points acquired from a cMD simulation. Using the block bootstrap we estimated averages and their corresponding variances. The errors reported in the plots in the *Results* section are the standard deviations of the estimated averages.

B.3 Merging two clusterings from the AP method

Below we give a short description of the Affinity Propagation (AP) method and explain how we used this approach to produce an initial set of solutions for the genetic clustering algorithm.

Given an affinity matrix, \mathcal{C} , the AP algorithm constructs the *availability*, \mathcal{A} , and *responsibility*, \mathcal{R} , matrices (see Figure B.1; a good description of the AP method can be found in [26]). Then, for object i , the value of k which maximizes $[\mathcal{A}]_{ik} + [\mathcal{R}]_{ik}$ indicates whether i is an exemplar (if $i = k$), or identifies the exemplar to which i is assigned. An object is chosen to be an exemplar if it is indeed the best representative of a group, or if it is dissimilar to all other objects. In the latter case, such an exemplar is in fact an outcast, having a negative or null affinity to other objects.

It is important to note, that the cooperation matrix (introduced in the main article) has zeros on the diagonal (an atom does not interact with itself), whereas in the AP method a value on the diagonal indicates the predisposition of an object to become an exemplar. Frey and Dueck suggest setting the diagonal with a mean affinity of a given object to all other objects, or simply setting the whole diagonal with equal values [31].

We needed to produce a single partitioning independent of the sign of the collective variable. For this purpose, two sets of exemplars, E_+ and E_- , were inferred from \mathcal{C} and $-\mathcal{C}$ affinity matrices. Our aim was to merge these two sets of exemplars, and their followers, into one clustering, i.e. to distribute all objects between two sets, P_+ and P_- . The P_+ group corresponds to the indices of atoms comprising the reverse cogs, and the P_- to the forward cogs (denoted in the main article by RC and FC, respectively).

First, if an object was an exemplar with no followers in E_- , we dispatched it to P_+ (and analogously for E_+ and P_-), because singular exemplars were outcasts with no positive-weighted edges to any other nodes. Non-singular exemplars in E_- were dispatched to P_- (analogously, in the other way around). We canceled objects which had a null affinity to all other objects, therefore a situation in which an object is a singular exemplar in both partitionings did not occur.

Next, we looked at objects which were not chosen as exemplars in either of the AP runs. These objects were initially assigned to two alternative groups, and to appoint them to P_+ or P_- , we checked their cumulative affinity to their suggested clusters. If the total affinity of an object to its exemplar in E_- (and followers) was greater than the alternative total affinity (associated with E_+), the object

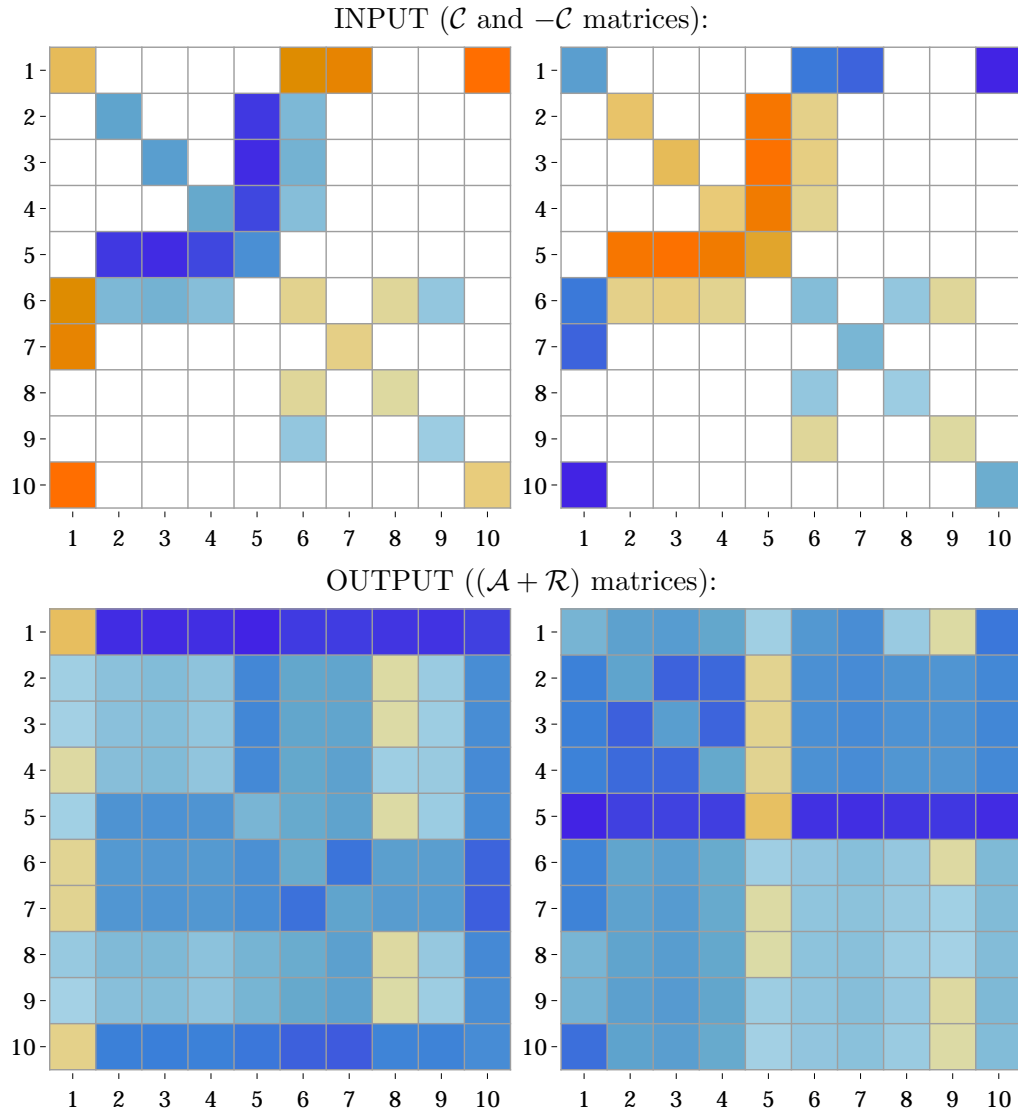


Figure B.1: **Input and output of the AP method.** Matrices $(\mathcal{A} + \mathcal{R})$ produced for affinity matrices \mathcal{C} and $-\mathcal{C}$ contain the information required for a final clustering. Positive elements in matrices are denoted with warm colors, whereas blue squares indicate negative values. The AP algorithm found two exemplars for \mathcal{C} (nodes 1 and 8), and two for $-\mathcal{C}$ (nodes 5 and 9). These were the nodes for which there were positive values in the $(\mathcal{A} + \mathcal{R})$ matrices.

was added to the P_- group (and *vice versa*). The result of this procedure was a rudimentary partitioning.

We observed that the diagonal-setting step had a huge impact on the clustering, and that a fairly good SCORE can be achieved by manipulating the diagonal. Values of the diagonal elements can either be assigned based on values in their

corresponding rows, or can be set uniformly based on the values in the whole matrix. To generate a set of high-scoring initial partitionings for the genetic clustering algorithm, we carried out clusterings with the following list of methods to setting up the diagonal:

- mean over affinities in a row;
- mean over non-zero affinities in a row;
- uniform: minimal positive affinity within the whole matrix;
- uniform: maximal positive affinity within the whole matrix;
- uniform: mean over positive affinities within the whole matrix.

Finally, we arrived at five affinity matrices, resulting in five clusterings, which were then mutated to create a set of solutions to initialize the genetic clustering procedure.

B.4 Parameters of the genetic clustering algorithm

Within the framework of the AP algorithm, there were cases in which even the smallest variations in the values on the diagonal of the affinity matrix led to significantly different clusterings. However, in most cases the obtained results had fairly high SCOREs. We could not determine whether this resulted from the properties of the graphs at hand, or simply from a drawback of the AP method. We realized that we can use the AP procedure to generate sets of diverse, but fairly high-scoring results. Thus, we decided to employ the genetic clustering algorithm, which is known to rely on the quality of the initial set of results.

The genetic clustering algorithm, aided by the initialization procedure based on the AP algorithm, returned the optimal partitionings for a wide range of its parameters. When testing the efficacy of the genetic procedure, we scanned four parameters: the size of the population after each selection, the number of pair used for the crossover stage, the number of specimens used for the mutation stage, and the number of steps used in the stopping criterion. The default values of these parameters were: 200, 50, 20, and 10, respectively (as detailed in the main article). Keeping the ratio between these parameters fixed, a decrease of their values made the method faster, but there was a greater chance of arriving at a sub-optimal solution. An increase of their values led to the optimal solution with an even higher probability, but at the cost of a longer running time. The default values were chosen such that in all cases the genetic algorithm would yield the optimal partitionings.

B.5 Complete results for the CCC(I)I, NCC(I)I, CC1CC(I)I molecules

Tables: B.1, B.2, B.3, B.4, B.5, B.6 collect results for the additional models discussed in the main article. Note that in all cases the partitionings found by the

genetic clustering algorithm were the optimal ones.

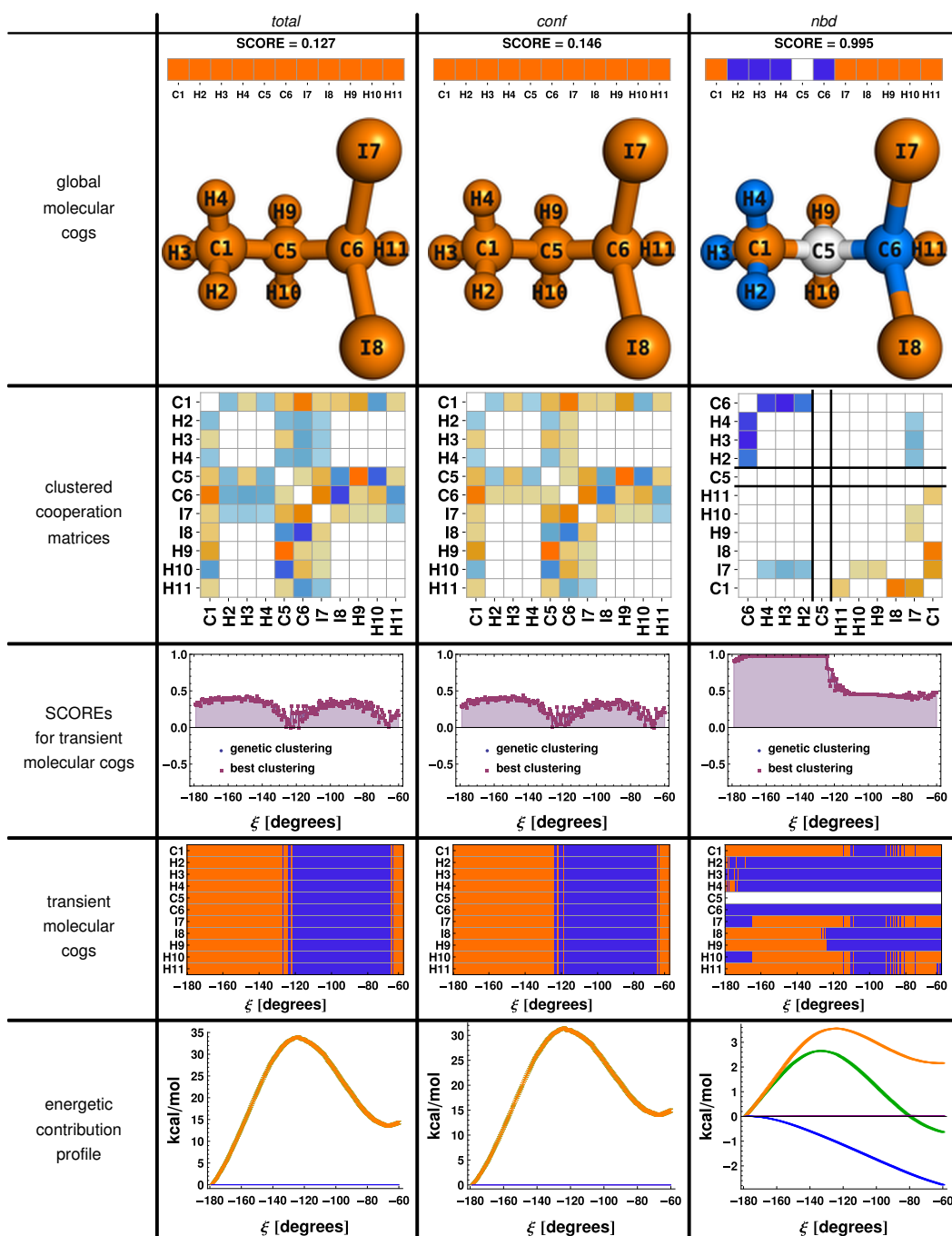


Table B.1: Results summary for molecule CCC(I)I, for interactions: *total*, *conf* and *nbd*.

B.5. COMPLETE RESULTS FOR THE CCC(I)I, NCC(I)I, CCLCC(I)I MOLECULES93

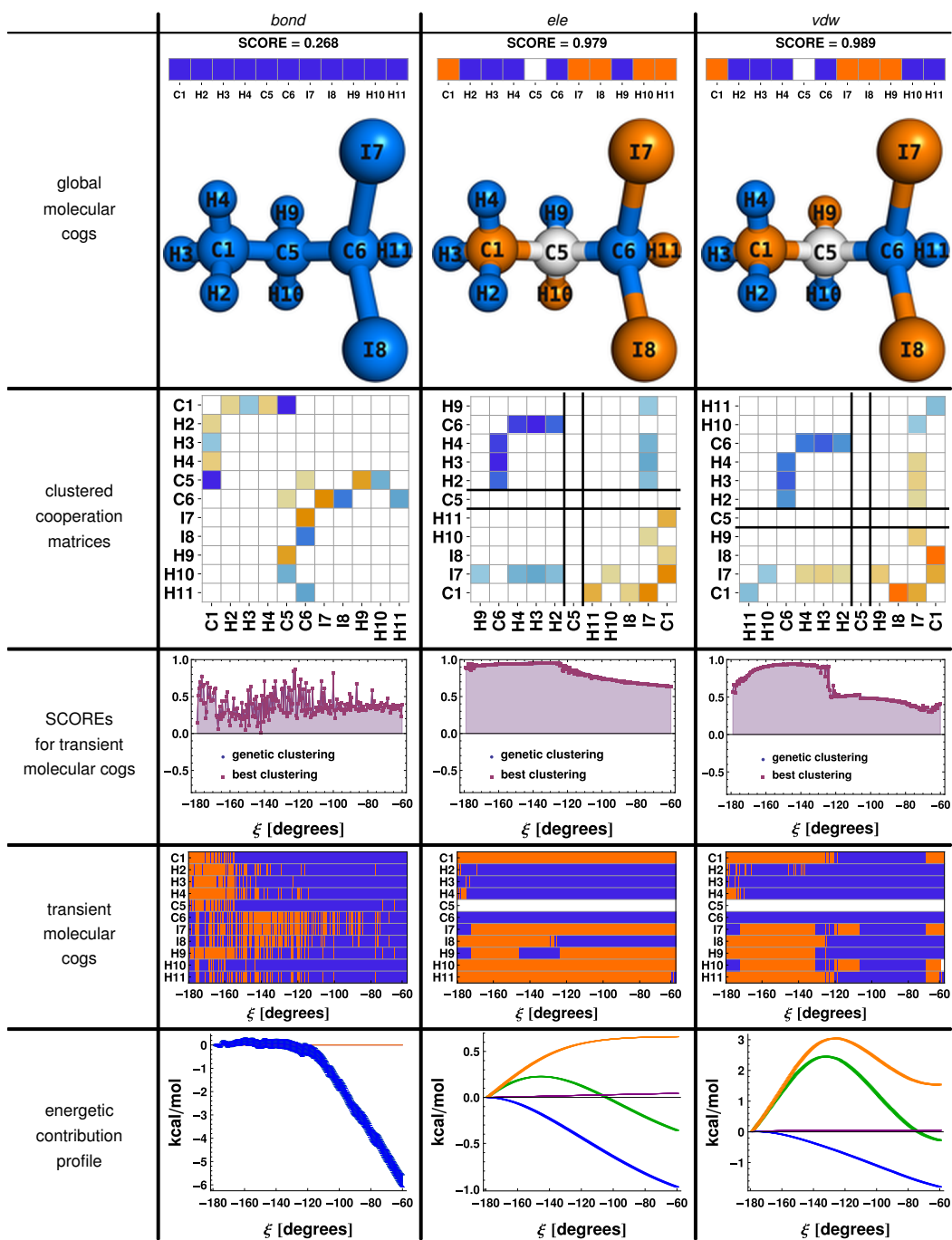


Table B.2: Results summary for molecule CCC(I)I, for interactions: *bond*, *ele* and *vdw*.

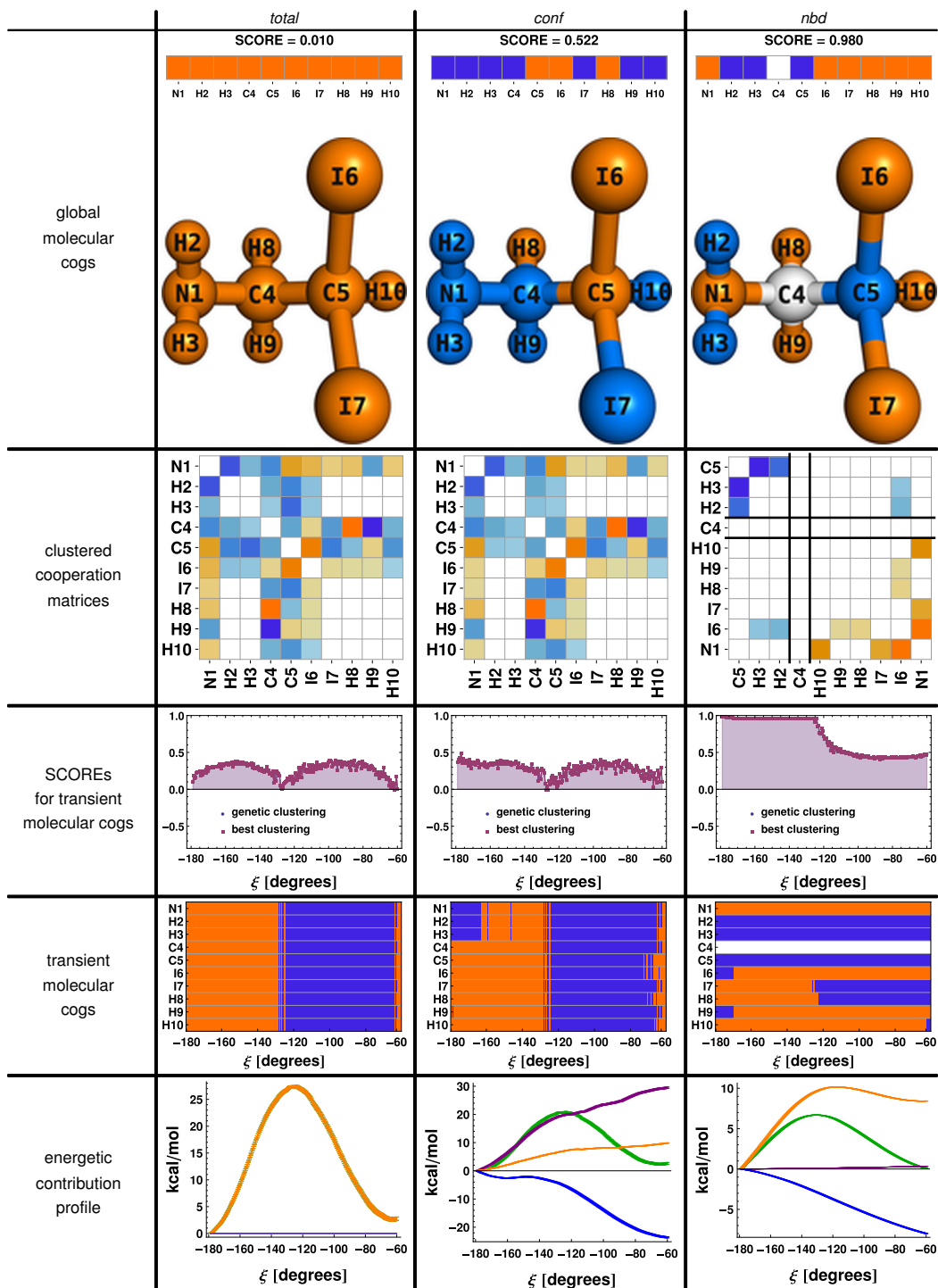


Table B.3: Results summary for molecule NCC(I)I, for interactions: *total*, *conf* and *nb*.

B.5. COMPLETE RESULTS FOR THE $CCC(I)I$, $NCC(I)I$, $CCLCC(I)I$ MOLECULES95

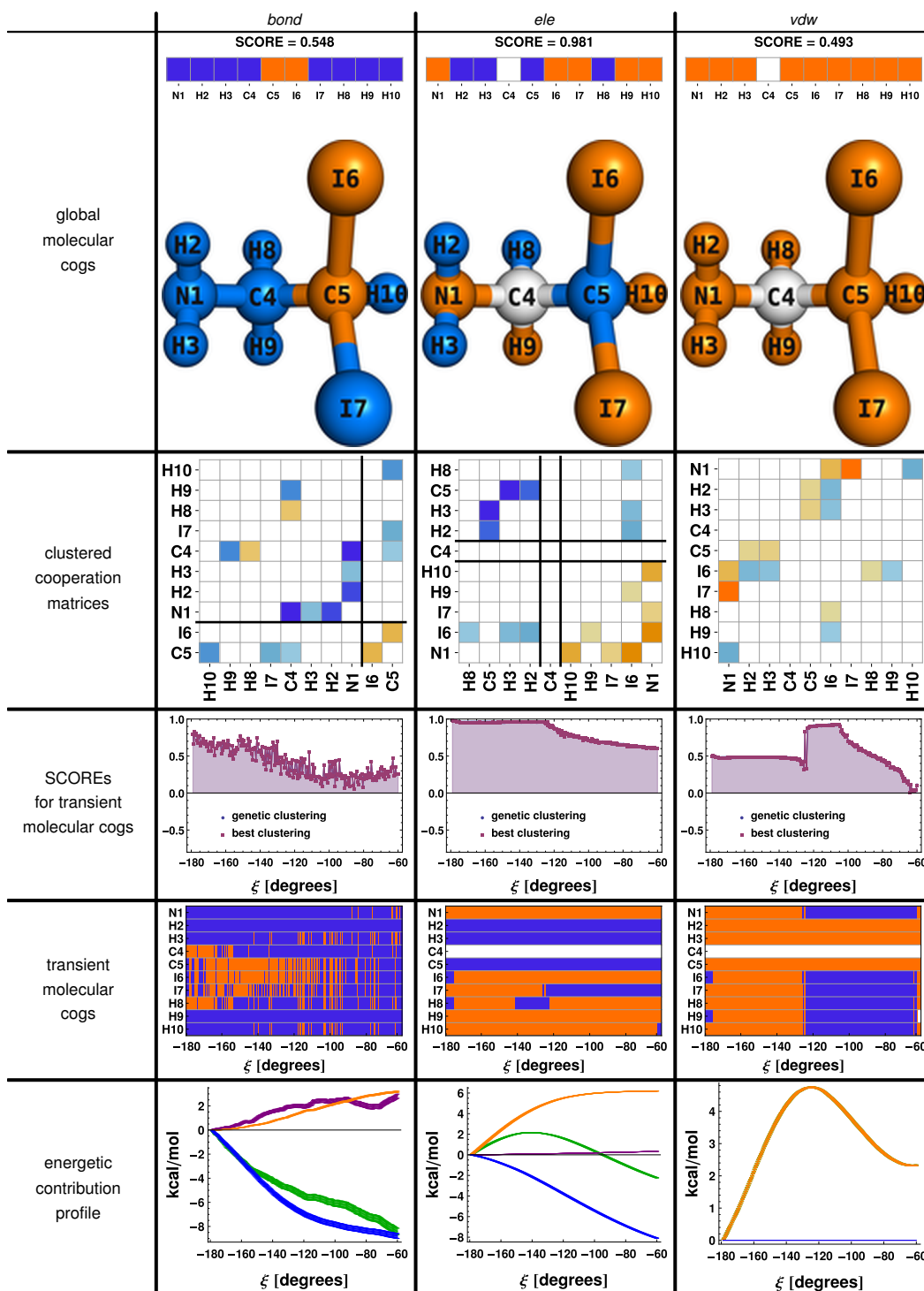


Table B.4: Results summary for molecule $NCC(I)I$, for interactions: *bond*, *ele* and *vdw*.

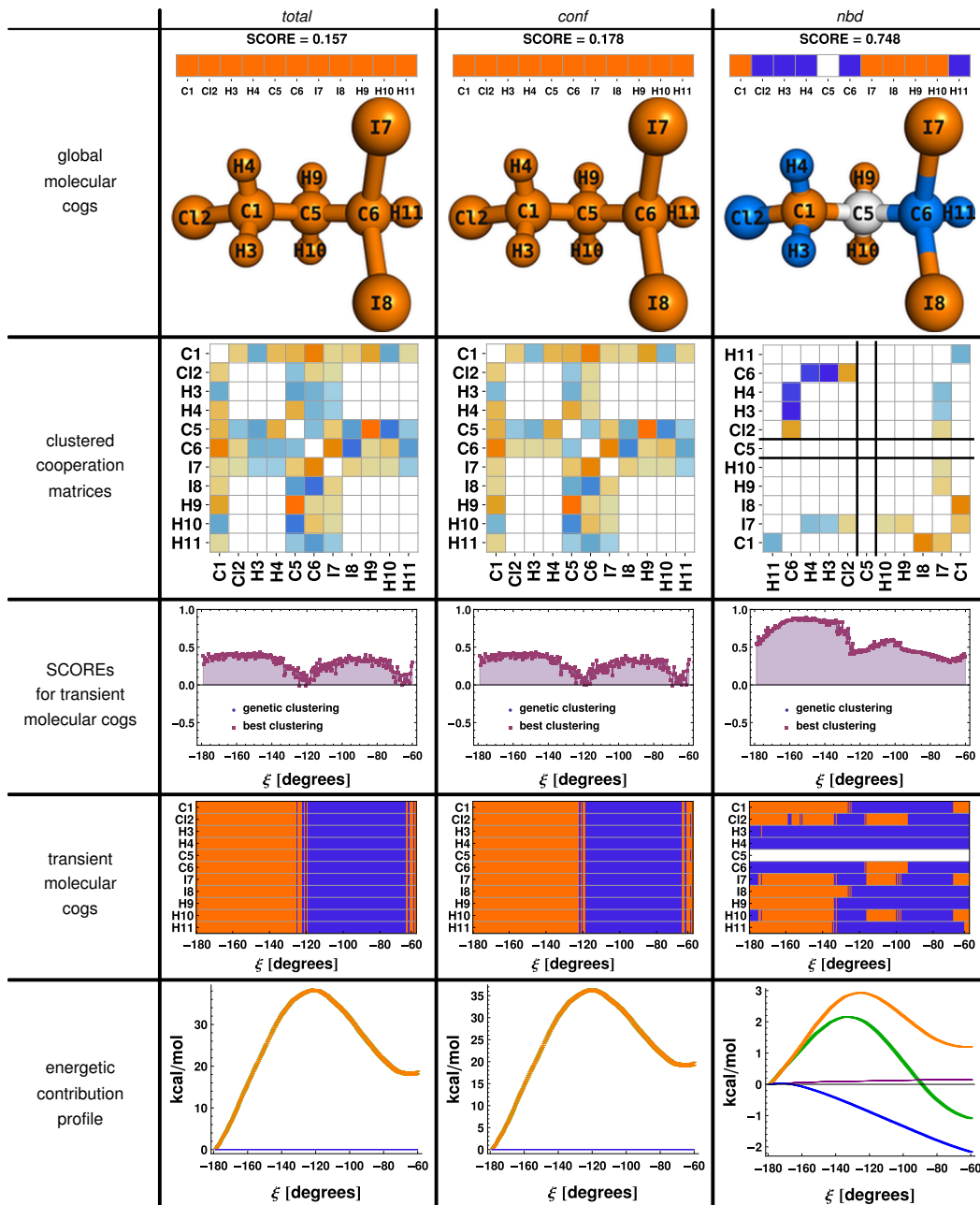


Table B.5: Results summary for molecule CC1CC(I)I , for interactions: *total*, *conf* and *nbd*.

B.5. COMPLETE RESULTS FOR THE $CCC(I)I$, $NCC(I)I$, $CCLCC(I)I$ MOLECULES97

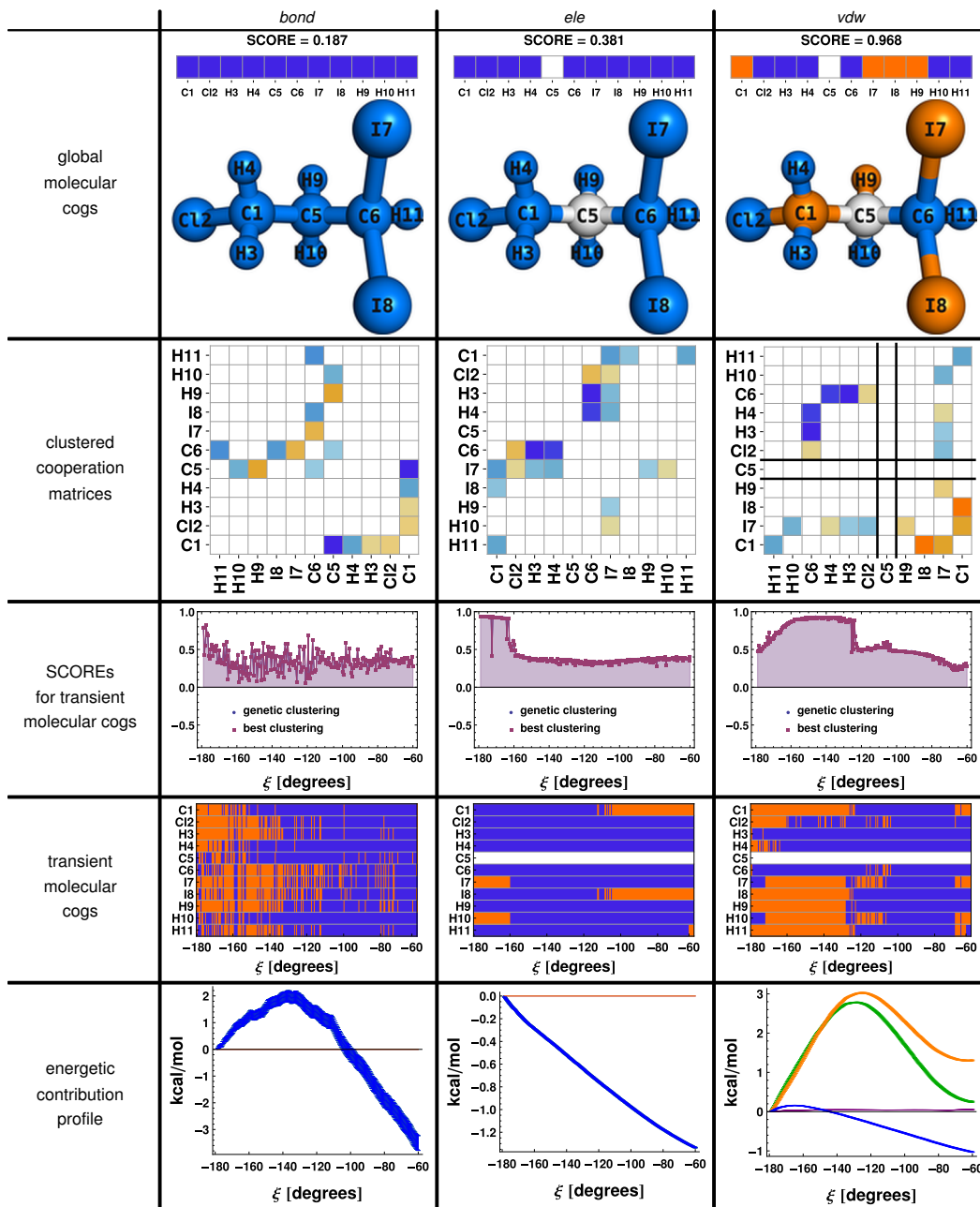


Table B.6: Results summary for molecule $CCLCC(I)I$, for interactions: *bond*, *ele* and *vdw*.

Bibliography

- [1] DM van Aalten et al. “Engineering protein mechanics: inhibition of concerted motions of the cellular retinol binding protein by site-directed mutagenesis.” In: *Protein engineering* 10.1 (1997), pp. 31–37.
- [2] Alexandros Altis et al. “Construction of the free energy landscape of biomolecules via dihedral angle principal component analysis”. In: *The Journal of chemical physics* 128.24 (2008), p. 245102.
- [3] Andrea Amadei, Antonius Linssen, and Herman JC Berendsen. “Essential dynamics of proteins”. In: *Proteins: Structure, Function, and Bioinformatics* 17.4 (1993), pp. 412–425.
- [4] Yelena A. Arnautova, Ruben A. Abagyan, and Maxim Totrov. “Development of a new physics-based internal coordinate mechanics force field and its application to protein loop modeling”. In: *Proteins: Structure, Function, and Bioinformatics* 79.2 (2011), pp. 477–498.
- [5] Ivet Bahar, Ali Rana Atilgan, and Burak Erman. “Direct evaluation of thermal fluctuations in proteins using a single-parameter harmonic potential”. In: *Folding and Design* 2.3 (1997), pp. 173–181.
- [6] G Bao. “Protein mechanics: a new frontier in biomechanics”. In: *Experimental mechanics* 49.1 (2009), pp. 153–164.
- [7] Stefan Bernhard and Frank Noe. “Optimal identification of semi-rigid domains in macromolecules from molecular dynamics simulation”. In: *PloS one* 5.5 (2010), e10491.
- [8] Stefan Boresch and Martin Karplus. “The meaning of component analysis: decomposition of the free energy in terms of specific interactions”. In: *Journal of molecular biology* 254.5 (1995), pp. 801–807.
- [9] Peer Bork. “Shuffled domains in extracellular proteins”. In: *FEBS letters* 286.1 (1991), pp. 47–54.
- [10] G Patrick Brady and Kim A Sharp. “Decomposition of interaction free energies in proteins and other complex systems”. In: *Journal of molecular biology* 254.1 (1995), pp. 77–85.
- [11] ZIMEI Bu and DJ Callaway. “Proteins move! Protein dynamics and long-range allostery in cell signaling”. In: *Adv Protein Chem Struct Biol* 83 (2011), pp. 163–221.

- [12] Edward Carlstein. “The use of subseries values for estimating the variance of a general statistic from a stationary sequence”. In: *The Annals of Statistics* (1986), pp. 1171–1179.
- [13] EA Carter et al. “Constrained reaction coordinate dynamics for the simulation of rare events”. In: *Chemical Physics Letters* 156.5 (1989), pp. 472–477.
- [14] Tianqi Chen and Carlos Guestrin. “Xgboost: A scalable tree boosting system”. In: *arXiv preprint arXiv:1603.02754* (2016).
- [15] Harry Chiang et al. “Molecular mechanics and dynamics characterization of an in silico mutated protein: A stand-alone lab module or support activity for in vivo and in vitro analyses of targeted proteins”. In: *Biochemistry and Molecular Biology Education* 41.6 (2013), pp. 402–408.
- [16] Christophe Chipot. “Frontiers in free-energy calculations of biological systems”. In: *Wiley Interdisciplinary Reviews: Computational Molecular Science* 4.1 (2014), pp. 71–89.
- [17] Christophe Chipot and Andrew Pohorille. *Free energy calculations: theory and applications in chemistry and biology*. Vol. 86. Springer, 2007.
- [18] Rowena Marie Cole. *Clustering with genetic algorithms*. Citeseer, 1998.
- [19] Jeffrey Comer et al. “The Adaptive Biasing Force Method: Everything You Always Wanted To Know but Were Afraid To Ask”. In: *The Journal of Physical Chemistry B* (2014).
- [20] Pawel Daniluk and Bogdan Lesyng. “A novel method to compare protein structures using local descriptors”. In: *BMC bioinformatics* 12.1 (2011), p. 344.
- [21] Pawel Daniluk and Bogdan Lesyng. “Theoretical and Computational Aspects of Protein Structural Alignment”. In: *Computational Methods to Study the Structure and Dynamics of Biomolecules and Biomolecular Processes*. Springer, 2014, pp. 557–598.
- [22] Paweł Daniluk et al. “From experimental, structural probability distributions to the theoretical causality analysis of molecular changes”. In: *Computer Assisted Methods in Engineering and Science* 19.3 (2012), pp. 257–276.
- [23] Ken A Dill. “Additivity principles in biochemistry”. In: *Journal of Biological Chemistry* 272.2 (1997), pp. 701–704.
- [24] Maciej Dziubiński, Paweł Daniluk, and Bogdan Lesyng. “ResiCon: a method for the identification of dynamic domains, hinges and interfacial regions in proteins”. In: *Bioinformatics* 32.1 (2016), pp. 25–34.
- [25] Maciej Dziubiński and Bogdan Lesyng. “Toward the identification of molecular cogs”. In: *Journal of computational chemistry* (2015).
- [26] Bradley Efron. *Bootstrap methods: another look at the jackknife*. Springer, 1992.

- [27] Martin Ester et al. “A density-based algorithm for discovering clusters in large spatial databases with noise.” In: *Kdd*. Vol. 96. 34. 1996, pp. 226–231.
- [28] Bela Farago et al. “Activation of nanoscale allosteric protein domain motion revealed by neutron spin echo spectroscopy”. In: *Biophysical journal* 99.10 (2010), pp. 3473–3482.
- [29] Evelyn Fix and Joseph L Hodges Jr. *Discriminatory analysis-nonparametric discrimination: consistency properties*. Tech. rep. DTIC Document, 1951.
- [30] Daron I Freedberg et al. “Rapid structural fluctuations of the free HIV protease flaps in solution: relationship to crystal structures and comparison with predictions of dynamics calculations”. In: *Protein science* 11.2 (2002), pp. 221–232.
- [31] Brendan J Frey and Delbert Dueck. “Clustering by passing messages between data points”. In: *science* 315.5814 (2007), pp. 972–976.
- [32] Alessandro Genoni, Giulia Morra, and Giorgio Colombo. “Identification of domains in protein structures from the analysis of intramolecular interactions”. In: *The Journal of Physical Chemistry B* 116.10 (2012), pp. 3331–3343.
- [33] Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al. “Similarity search in high dimensions via hashing”. In: *VLDB*. Vol. 99. 6. 1999, pp. 518–529.
- [34] Alexander N Gorban et al. *Principal manifolds for data visualization and dimension reduction*. Vol. 58. Springer, 2008.
- [35] A Gorecki, J Trylska, and B Lesyng. “Causality and correlation analyses of molecular dynamics simulation data”. In: *From Computational Biophysics to Systems Biology (CBSB07), volume NIC Series 36* (), pp. 25–30.
- [36] Adam Gorecki et al. “RedMD – reduced molecular dynamics package”. In: *Journal of computational chemistry* 30.14 (2009), pp. 2364–2373.
- [37] Barry J Grant et al. “Bio3d: an R package for the comparative analysis of protein structures”. In: *Bioinformatics* 22.21 (2006), pp. 2695–2696.
- [38] James C Gumbart, Benoit Roux, and Christophe Chipot. “Standard binding free energies from computer simulations: What is the best strategy?” In: *Journal of chemical theory and computation* 9.1 (2012), pp. 794–802.
- [39] Peter Hall, Joel L Horowitz, and Bing-Yi Jing. “On blocking rules for the bootstrap with dependent data”. In: *Biometrika* 82.3 (1995), pp. 561–574.
- [40] Donald Hamelberg and J Andrew McCammon. “Fast peptidyl cis-trans isomerization within the flexible Gly-rich flaps of HIV-1 protease”. In: *Journal of the American Chemical Society* 127.40 (2005), pp. 13778–13779.
- [41] Jiawei Han and Micheline Kamber. “Data mining: concepts and techniques”. In: *United States of America: Morgan Kaufmann Publishers* (2001).

- [42] Steven Hayward and Herman JC Berendsen. “Systematic analysis of domain motions in proteins from conformational change: New results on citrate synthase and T 4 lysozyme”. In: *Proteins Structure Function and Genetics* 30.2 (1998), pp. 144–154.
- [43] Steven Hayward, Akio Kitao, and Herman JC Berendsen. “Model-free methods of analyzing domain motions in proteins from simulation: a comparison of normal mode analysis and molecular dynamics simulation of lysozyme”. In: *Proteins: Structure, Function, and Bioinformatics* 27.3 (1997), pp. 425–437.
- [44] Konrad Hinsén. “Analysis of domain motions by approximate normal mode calculations”. In: *Proteins Structure Function and Genetics* 33.3 (1998), pp. 417–429.
- [45] Tin Kam Ho. “Random decision forests”. In: *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*. Vol. 1. IEEE. 1995, pp. 278–282.
- [46] Gareth James et al. *An introduction to statistical learning*. Vol. 6. Springer, 2013.
- [47] Wolfgang Kabsch. “A solution for the best rotation to relate two sets of vectors”. In: *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography* 32.5 (1976), pp. 922–923.
- [48] Donata K Kirchner and Peter Guntert. “Objective identification of residue ranges for the superposition of protein structures”. In: *BMC bioinformatics* 12.1 (2011), p. 170.
- [49] John G Kirkwood. “Statistical mechanics of fluid mixtures”. In: *The Journal of Chemical Physics* 3.5 (1935), pp. 300–313.
- [50] Jens-Peter Kreiss and Efsthathios Paparoditis. “Bootstrap methods for dependent data: A review”. In: *Journal of the Korean Statistical Society* 40.4 (2011), pp. 357–378.
- [51] Richard Lavery and Sophie Sacquin-Mora. “Protein mechanics: a route from structure to function”. In: *Journal of biosciences* 32.1 (2007), pp. 891–898.
- [52] Richard A Lee, Moe Razaz, and Steven Hayward. “The DynDom database of protein domain motions”. In: *Bioinformatics* 19.10 (2003), pp. 1290–1291.
- [53] Laurens van der Maaten and Geoffrey Hinton. “Visualizing data using t-SNE”. In: *Journal of Machine Learning Research* 9.Nov (2008), pp. 2579–2605.
- [54] James MacQueen et al. “Some methods for classification and analysis of multivariate observations”. In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. 14. Oakland, CA, USA. 1967, pp. 281–297.
- [55] Gary E Martin, Andrew S Zektzer, et al. *Two Dimensional NMR Methods for Establishing Molecular Connectivity*. VCH, 1988.

- [56] Marina Meila. “Comparing clusterings – an information based distance”. In: *Journal of multivariate analysis* 98.5 (2007), pp. 873–895.
- [57] Arun Nair et al. “Massively parallel methods for deep reinforcement learning”. In: *arXiv preprint arXiv:1507.04296* (2015).
- [58] Noel M O’Boyle et al. “Open Babel: An open chemical toolbox”. In: *J Cheminf* 3 (2011), p. 33.
- [59] Wouter K den Otter. “Revisiting the Exact Relation between Potential of Mean Force and Free-Energy Profile”. In: *Journal of Chemical Theory and Computation* 9.9 (2013), pp. 3861–3865.
- [60] Xavier Periole et al. “Structural determinants of the supramolecular organization of G protein-coupled receptors in bilayers”. In: *Journal of the American Chemical Society* 134.26 (2012), pp. 10959–10965.
- [61] Raffaello Potestio, Francesco Pontiggia, and Cristian Micheletti. “Coarse-grained description of protein internal dynamics: an optimal strategy for decomposing proteins in rigid subunits”. In: *Biophysical journal* 96.12 (2009), pp. 4993–5002.
- [62] William H Press. *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press, 2007.
- [63] Jane S Richardson. *The anatomy and taxonomy of protein structure*. Vol. 34. Academic Press, 1981.
- [64] Julia Romanowska, Krzysztof S Nowinski, and Joanna Trylska. “Determining geometrically stable domains in molecular conformation sets”. In: *Journal of Chemical Theory and Computation* 8.8 (2012), pp. 2588–2599.
- [65] Sophie Sacquin-Mora. “Motions and mechanics: investigating conformational transitions in multi-domain proteins with coarse-grain simulations”. In: *Molecular Simulation* 40.1-3 (2014), pp. 229–236.
- [66] S Kashif Sadiq and Gianni De Fabritiis. “Explicit solvent dynamics and energetics of HIV-1 protease flap opening and closing”. In: *Proteins: Structure, Function, and Bioinformatics* 78.14 (2010), pp. 2873–2885.
- [67] Marco Sarich, Frank Noé, and Christof Schütte. “On the approximation quality of Markov state models”. In: *Multiscale Modeling & Simulation* 8.4 (2010), pp. 1154–1177.
- [68] Christian Seifert and Frauke Gräter. “Protein mechanics: How force regulates molecular function”. In: *Biochimica et Biophysica Acta (BBA)-General Subjects* 1830.10 (2013), pp. 4762–4768.
- [69] Anton V Sinitskiy, Marissa G Saunders, and Gregory A Voth. “Optimal number of coarse-grained sites in different components of large biomolecular complexes”. In: *The Journal of Physical Chemistry B* 116.29 (2012), pp. 8363–8374.

- [70] David A Snyder and Gaetano T Montelione. “Clustering algorithms for identifying core atom sets and for assessing the precision of protein structure ensembles”. In: *Proteins: Structure, Function, and Bioinformatics* 59.4 (2005), pp. 673–686.
- [71] R. R. Sokal and C. D. Michener. “A statistical method for evaluating systematic relationships”. In: *University of Kansas Scientific Bulletin* 28 (1958), pp. 1409–1438.
- [72] Yuji Sugita, Akio Kitao, and Yuko Okamoto. “Multidimensional replica-exchange method for free-energy calculations”. In: *The Journal of Chemical Physics* 113.15 (2000), pp. 6042–6051.
- [73] Daniel Taylor, Gavin Cawley, and Steven Hayward. “Classification of domain movements in proteins using dynamic contact graphs”. In: (2013).
- [74] Glenn M Torrie and John P Valleau. “Nonphysical sampling distributions in Monte Carlo free-energy estimation: Umbrella sampling”. In: *Journal of Computational Physics* 23.2 (1977), pp. 187–199.
- [75] Mark E Tuckerman. “Free Energy Calculations: Theory and Applications in Chemistry and Biology”. In: *Journal of the American Chemical Society* 129.35 (2007), pp. 10963–10964.
- [76] Jiri Vondrasek and Alexander Wlodawer. “HIVdb: a database of the structures of human immunodeficiency virus protease”. In: *Proteins: Structure, Function, and Bioinformatics* 49.4 (2002), pp. 429–431.
- [77] Marcus Weber, Wasinee Rungtarityotin, and Alexander Schliep. *Perron cluster analysis and its connection to graph partitioning for noisy data*. Konrad-Zuse-Zentrum für Informationstechnik Berlin, 2004.
- [78] Hyung-June Woo and Benoit Roux. “Calculation of absolute protein–ligand binding free energy from computer simulations”. In: *Proceedings of the national academy of sciences of the united states of america* 102.19 (2005), pp. 6825–6830.
- [79] Willy Wriggers and Klaus Schulten. “Protein domain movements: detection of rigid domains and visualization of hinges in comparisons of atomic coordinates”. In: *Proteins Structure Function and Genetics* 29.1 (1997), pp. 1–14.
- [80] Junjie Wu et al. “External validation measures for K-means clustering: A data distribution perspective”. In: *Expert Systems with Applications* 36.3 (2009), pp. 6050–6061.
- [81] Yuzhen Ye and Adam Godzik. “Flexible structure alignment by chaining aligned fragment pairs allowing twists”. In: *Bioinformatics* 19.suppl 2 (2003), pp. ii246–ii255.
- [82] Semen O Yesylevskyy, Valery N Kharkyanen, and Alexander P Demchenko. “Dynamic protein domains: identification, interdependence, and stability”. In: *Biophysical journal* 91.2 (2006), pp. 670–685.

- [83] Zhiyong Zhang et al. “Defining coarse-grained representations of large biomolecules and biomolecular complexes from elastic network models”. In: *Biophysical journal* 97.8 (2009), pp. 2327–2337.