GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

Probabilistic models for protein conformational changes

Dissertation
for the award of the degree
"Doctor rerum naturalium" (Dr. rer. nat.)
of the Georg-August-Universität Göttingen

in PhD Program of Computer Science (PCS)
of Georg-August University School of Science (GAUSS)

submitted by
Chuong Thach, Nguyen
from Nghe An, Vietnam

Göttingen 2020

Thesis Committee
Prof. Dr. Michael Habeck
University Hospital Jena
Prof. Dr. Stephan Waack
Georg-August-Universität Göttingen

Members of the Examination Board
1st Referee: Prof. Dr. Michael Habeck
University Hospital Jena

2nd Referee: Prof. Dr. Stephan Waack
Georg-August-Universität Göttingen

Further Members of the Examination Board
Dr. Johannes Söding
Quantitative and Computational Biology
Max Planck Institute for Biophysical Chemistry

Prof. Dr. Jörg Enderlein
III. Physikalisches Institut – Biophysik
Georg-August-Universität Göttingen

Prof. Dr. Winfried Kurth
Department Ecoinformatics, Biometrics & Forest Growth
Georg-August-Universität Göttingen

Prof. Dr. Daniel Rudolf
Institute for Mathematical Stochastics.
Georg-August-Universität Göttingen

Date of oral examination: May 22nd, 2020

I hereby declare that I have written this thesis independently without any help from others and without the use of documents or aids other than those stated. I have mentioned all used sources and cited them correctly according to established academic citation rules.

Göttingen 2020

Abstract

Proteins are macromolecules that perform multiple functions. They are not rigid molecules, but instead proteins can change their conformation to perform critical tasks driven by binding small ligands, by assembling into large macromolecular complexes or by physiological factors. Characterization of protein conformational change and analyzing transitional pathways along protein conformational states are essentially tasks for computational biology. Here we propose probabilistic models to characterize protein conformational change. The first model disentangles protein structure into rigid bodies, whereas the second model proposes the probabilistic network model for the transitions between conformational states. Our first model is a generative process using Gaussian mixture models to represent rigid domains, which generated the input structures through spatial transformation. To estimate our model parameters, we use two approaches: using deterministic Expectation-Maximization algorithm and stochastic Gibbs sampler. The second model is an elastic way to expand the application spectrum of our model. The model uses anharmonic springs that involve the molecular distances that are allowed to break in a stochastic fashion. The function of the spring potential is inferred from a statistical analysis of a database of large-scale conformational changes in proteins. In addition we deploy our model in a webservice, as well as we deposit a precomputed dataset of rigid domains and a selective dataset of conformational pathway between conformational states. Finally, we employ graph-based algorithms to solve the problem of a model-free base solution. This work is not limited to biological applications, but can also be applied to robotics and computer vision.
This thesis is based on the following publications and manuscripts, respectively:

- Thach Nguyen, Michael Habeck, A probabilistic model for detecting rigid domains in protein structures, Bioinformatics, Volume 32, Issue 17, 1 September 2016, Pages i710–i717, https://doi.org/10.1093/bioinformatics/btw442
- Habeck M, Nguyen T. A probabilistic network model for structural transitions in biomolecules.Proteins. 2018;86:634–643.https://doi.org/10.1002/prot.25490
- Linh Dang, Thach Nguyen, Michael Habeck, and Stephan Waack. A graph-based algorithm for detecting rigid domains in protein structures. Submitted
- Thach Nguyen, Christian Böhm, Michael Habeck, A computational web server for segmenting protein structure into rigid bodies, in preparation.

Acknowledgment

I want to express my deepest and sincere gratitude to my supervisor Prof. Dr. Michael Habeck, for supporting me during my Ph.D. and for giving me invaluable advice, not only also work for life experience here. His example in working and vision outside inspired me in all these years and will guide me in the future.
I would also like to thank the people I collaborated with: Linh Dang and Prof. Stephan Waack. I also thank Dr. Johannes Söding, Prof. Dr. Jörg Enderlein, Prof. Dr. Winfried Kurth and Prof. Dr. Daniel Rudolf for serving on the examination board of this dissertation.
My gratitude goes to the whole IMS and MPI staff as well; in particular, my officemate Nima Vakili, who works together, created a friendly working atmosphere at MPI.
I want to thank Dr. Chad Grabner for useful comments and proofreading. I must also thank the DFG for their financial support my research works. And finally, thank my family: my parents who sacrificed everything for me, my sister that always supported me, my beloved wife, and my daughter.

Contents

Chapter 1

Introduction

1.1 Protein structure and protein dynamics

Proteins are biological polymers, which are the building blocks and fundamental units of all life forms. Fibrous protein such as keratin, collagen create the structure of the cell and the living organ. Proteins perform most of the critical and vital tasks in living cell. Enzymes catalyze the biochemical reactions. The immune system uses antibodies to bind antigens. Hemoglobin and myoglobin store and transport oxygen, rhodopsin senses light, titin, and other motor protein build animal muscle to generate force. Protein is composed of 20 different natural amino acids that are encoded in the primary structure. In the native state, proteins are folded into three-dimensional (3D) structures, each 3D structure called a conformational state. Protein structure plays an essential role in understanding protein function. Several experimental methods are used to determine protein structure.
From the 1950s, the first protein structure myoglobin was elucidated by Xray crystallography (Kendrew *et al.* (1958)). X-ray crystallography is the primary method used to determine protein structure and has been used to solve more than 80 percent of the protein structures. The requirement of growing protein crystals, which may alter the structure, is the main limitation of the technique. In contrast, Nuclear Magnetic Resonant (NMR) can observe protein structure in their native states in solution (Wüthrich (1976), Wüthrich (2001)). NMR can capture protein conformational dynamics, however, using NMR to solve large protein structure and the requirement of a large purified sample are the main challenges of the method. These experimental methods create a protein database (PDB), which is an important data source for biologists. The three-dimensional structures of the

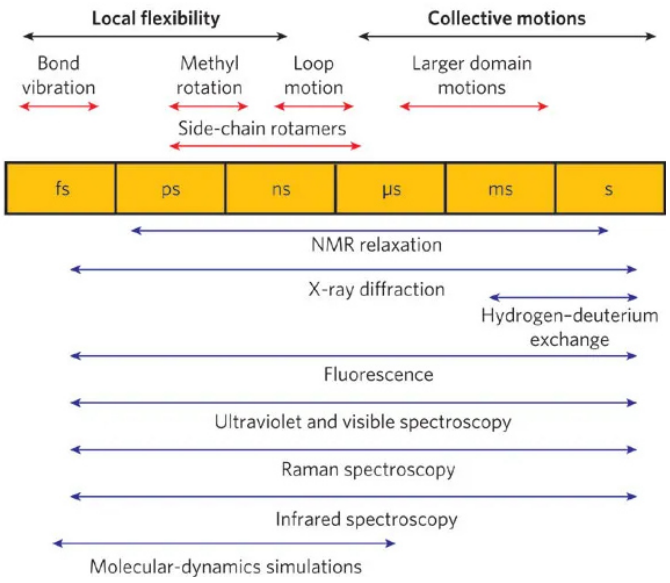

Figure 1.1: Timescale of dynamic processes in proteins and the experimental methods that can detect fluctuations on each timescale (Henzler-Wildman and Kern (2007))

protein, which are captured in "snapshots" of Xray crystal image or EM Map may give the wrong impression that protein structures are rigid. In physiological conditions, proteins are dynamic and flexible. The internal dynamics of protein structures can be classified into three main categories, which vary in amplitude and time scale (Voet and Voet (2010)).

- Atomic fluctuations, e.g., the vibration of individual bonds, which has a time scale from femtosecond ($10^{-15}s$) to tens of picosecond ($10^{-11}s$) and with a tiny spatial amplitude between 0.01 to 1 Å. These fluctuations can be quantified by the crystallography B-factor in Xray data.

- Collective motions are the movement of side-chains; have a longer time scale range from picosecond ($10^{-11}s$) to millisecond. These dynamics have a larger amplitude between 0.1 and less than 5Å.

- Large conformational changes, e.g., enzyme dynamics, referred to as allosteric that involve higher amplitude movements between 0.5 and several tens Å. The time scale varies from nanosecond to $10^{3}s$. In this work, we investigate large conformational changes.

Protein dynamics are important but difficult to study. Among the three categories, the third dynamic is the most difficult to observe in both experimental and computational studies.

NMR is the experimental method that can study these dynamics, but it has limits, as we mentioned before. The recent rise in the development of Cryogenic Electron Microscopy (CryoEM) can determine very large protein molecules in different conformational states (Frank (2002)). It can capture multistate of conformation. FRET (Hanson *et al.* (2007)) observes protein conformational change in their native state. These experimental methods provide information about protein dynamics.
Molecular dynamics (MD simulations) is an alternative way to simulate these fluctuations. MD simulations model the dynamics of proteins by applying physical forces such as electrostatic or hydrophobic and hydrophilic forces. There are several force fields which are well configurated for MD simulations. However, studying large conformational changes beyond a microsecond is still challenging because of computational cost. Figure 1.1 summarizes the entire spectrum of protein dynamics and methods used to study these dynamics.

1.2 Models for conformational change

1.2.1 Protein domains

The structure of proteins is organized into four levels. The primary structure consists of the list of amino acids. The second level, called secondary structure to classify protein structure into structural segments: alpha-helix, beta-sheet, and random coil. The third and fourth levels are tertiary structure, and quaternary structure refers to the whole structure of the polypeptide chain of a protein engaged in more complex interaction. In the tertiary structure, proteins consist of domains. Although protein domain has been a general concept in biology, there are several definitions of domain(Ponting and Russell (2002)). Protein domains can be defined on the sequence level using sequence comparison methods and correspond to evolutionary conserved parts of proteins. One of the first protein domains in the sequence database Pfam (Sonnhammer *et al.* (1997)) originally classified proteins into families. It can also elucidate the sequential protein domain using multiple alignments by the hidden Markov model. Protein domains can also be defined on the structural level. Under the structural definition, domains are compact, globular units of proteins that exist and fold independently. This is more robust because structure is more conserved than sequence. The SCOP (Structural Classification of Proteins database (Murzin *et al.* (1995))) and CATH (Class, Architecture, Topology, and Homologous (Orengo *et al.* (1997))) are two protein classifications that classified protein structure into several

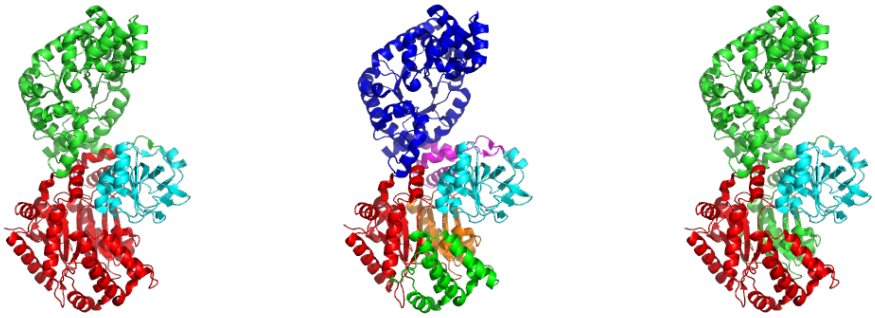

Figure 1.2: Comparison of Pyruvate phosphate dikinase (PPDK) domain from SCOP, CATH, and our structural domains. Left: SCOP domain notation SCOPID: 8037995, 8037992, 8037990. Middle: CATH domain notation, CATH Code: 3.30.1490.20. Our rigid domains for two conformations PDBID: 1KC7(A) and 2R82(A)

categories mostly by their folding type. The classifications are based on the secondary structure, such as mainly alpha-helix, mainly beta-sheet, the mixture of alpha-helix and beta-sheet, and categorize by folding type in semi automatical way with less consideration for domain identification. Besides, domains from SCOP and CATH are diverse. In some situations, for example, PPDK in Figure 1.2 below, domains from SCOP and CATH are different. Our domain is identified if there is a significant change in internal distance of two conformations. Therefore our model requires a minimum of two protein conformations, which have a large conformational change. The domain identification relies on analyzing the different distance matrix (Nichols *et al.* (1995)). The difference distance matrix is independent of the heterogeneous coordinate. To extend for multiple structures, we use difference distance matrix (DDM) which is defined as

$$\Delta_{nn'} = \frac{1}{M} \sum_{m} |d^m_{nn'} - \bar{d}_{nn'}| \,.$$

Where $d^m_{nn'}$ is the distance between atom n and n' in conformation m, $\bar{d}_{nn'}$ is the average distance over M conformations. Rigid domains correspond to patches in which $\Delta_{nn'}$ is close to zero because the internal structure does not change.
In Figure 1.2, we compare our domains with SCOP and CATH in large protein PPDK. Our domains in this example are similar to domains from SCOP. SCOP and CATH often identify domains as globular subsets from a single structure. However, in another example, Adenylate Kinase, Figure 1.3 shows our structure domain disagree with domain from SCOP and CATH. CATH and SCOP usually consider small proteins as single-domain proteins. Identifying dynamic domains is essential for understanding the biological

Figure 1.3: Comparison of Adenylate Kinase (AdK) domain from SCOP, CATH, and our structural domains. Left and Middle: Our domain notion PDBID: 4AKE(A) and 1AKE(A). Right: SCOP and CATH domain, SCOP ID: 8058333, CATH Code: 3.40.50.300

process, such as studying the macromolecule interaction, active site identification. One example of a protein domain is the conformational change of chaperonin GroEL/GroES in Figure 1.4. The chaperonin supports the proper folding of nascent polypeptide. Figures from left to right shows GroEL with GroES and GroEL without GroES (holo and apo form). Those allosteric conformational changes happen with GroES binding. Each GroEL chain is composed of three domains, the Apical, Intermedia, and Equatorial domain (Xu *et al.* (1997)). These domain structures can be viewed as rigid blocks when we consider their internal structure.

1.2.2 Computational methods and structural database

One of the first attempts was made by Gerstein and coworkers to discover the protein conformational change and classify them into hinge and shear motion (Gerstein *et al.* (1994)). Their works developed into algorithms and databases; one of them is the Rigidfinder (Abyzov *et al.* (2010)), which defined the rigidity criterion for small segmentations, using dynamic programming to expand the rigidity condition to discover the rigid domain. However, the program uses the cutoff threshold, which is not easily estimated in general. The other algorithms are StoneHinge(Keating *et al.* (2009)), FlexOracle (Flores and Gerstein (2007)) which use the energy minimization to determine the cutting point of segmentation. Their works published as Molecular Motion Database (MolmovDB) contain many trajectories of conformation changes (Gerstein and Krebs (1998)). Dyndom (Hayward and Berendsen (1998)) focuses on determining the protein structure domain

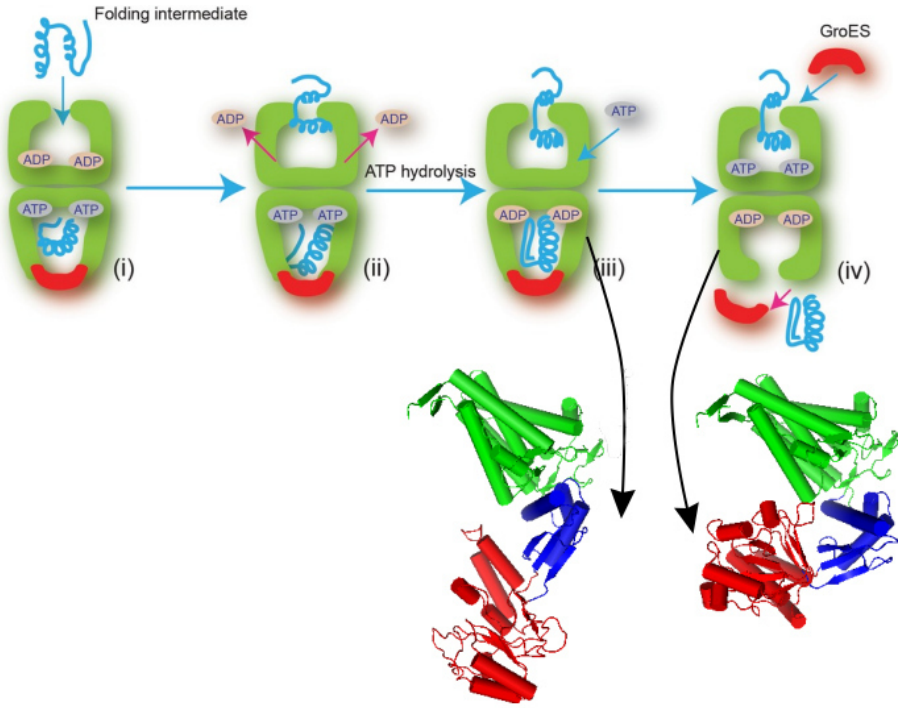

Figure 1.4: Allostery in GroEL/GroES from Xu *et al.* (1997). Bottom, the structure of two conformations GroeEL color by domain. Red: Apical, Blue: Intermediate, Green: Equatorial

and the hinge axis. This method uses K-mean to cluster the rotational vector and use the quality assessment to tune the number of clusters. Over the years, Dyndom database has assembled computational results from its user uploads. The dataset consists of pairwise structures of proteins in different conformational states, which gives us reliable data to test our algorithm. However, Dyndom has high redundancy, and the algorithm supports only a pair of structures are the limitation of this method. Other works for identifying the rigid regions in proteins such as Hingefind(Wriggers and Schulten (1997)), identifies and characterizes domain movements. Hingfind partitions two input structures into subsets and determines the best-fit Root Mean Square Deviation (RMSD) use Kabsch's algorithm (Kabsch (1976)). Recently Spectrus (Ponzoni *et al.* (2015)) used the spectral cluster to analyze the variance distance matrices in order to determine rigid parts of protein structures. The more ambitious problem is disentangling rigid bodies from a single structure by using normal mode analysis for a single structure firstly proposed in Hinsen (1998) and using elastic network model HingeProt (Emekli *et al.* (2008))). However, it is unclear when the mode spectrum spreads into more than two modes.
Those works give an overview of determining protein rigid and hinge position. However, none of these works proposes a model for protein conformational changes, which are essential for quality assessment as well as to determine the fitness of new data to the model. To study large conformational change, in my thesis, I proposed two probabilistic models that cover the whole spectrum of protein dynamics. The first model is the rigid domain model provides a coarse grained view of large scale dynamics in terms of rigid body movements. The second model is the adaptive network model in principle allows for full flexibility but tries to preserve the initial structure as much as possible.

1.2.3 Probabilistic model and Bayesian Inference

Our approach uses probabilistic models for conformational changes in protein. The input data of our model is experimental three-dimensional structures X. We model using probability distributions a set of parameter θ. The model use probability to quantify the uncertainty of our assumption. We use Bayesian statistics to infer the model parameters from the data. We employ Bayes rule for data X and model parameter θ:

$$p(\theta|X) = \frac{p(\theta) \times p(X|\theta)}{p(X)} \tag{1.1}$$

In this equation, the posterior distribution $p(\theta|X)$ is the distribution of parameters given the data. $p(X|\theta)$ is the likelihood which quantifies the fitness of data given model parameters. The prior distribution $p(\theta)$ is a distribution of the parameters without the data. The constant $p(X)$ is the marginal likelihood or "model evidence".
From 18th century Bayesian statistic is a controversial view of probability. Classical or frequentist view probability as repeatable events. Baysian statistic quantifies the uncertainty of the event or degree of our belief (Bishop (2006)). Under the classical view, the model parameter is fixed; therefore, the likelihood $p(X|\theta)$ is the function of data X. In contrast, there is only a single dataset under the Bayesian, and the likelihood is the function of parameter θ. The convenience of the Bayesian method is that we can cooperate with prior distributions. In practice, the prior distribution is selected in the conjugate form of the likelihood for mathematical convenience.
To infer the model parameters θ, we use two algorithms. The first approach uses Expectation Maximization (EM) (Dempster *et al.* (1977), Hirsch and Habeck (2008)) to estimate the model parameter in a deterministic way analytically. To find out the maximum of likelihood $p(X|\theta)$ with given data X, EM perform expectation step (E step) and maximization step (M step) iteratively.
The second approach employs Gibbs sampling (Geman and Geman (1984), Nguyen and Habeck (2016)) to sample the model parameters stochastically. The Bayesian infer the model parameter θ use data X, likelihood $p(X|\theta)$ by including a prior $p(\theta)$ over the parameters. To estimate a set of model parameter $\theta : (\theta_1, \theta_2 ... \theta_n)$, Gibbs sampler generates a Markov chain by sampling each parameter given the other parameters. An overview of the Gibbs sampler is presented in Algorithm 1.

Algorithm 1 Gibbs sampler

Require: Data X, number of Gibbs sampling iterations n_{iter}
  Initialize $\theta : (\theta_1, \theta_2 ... \theta_n)$
  for $i = 1, \ldots, n_{\text{iter}}$ **do**
    Sample θ_1 given (X, θ_2, θ_3 ..)
    Sample θ_2 given (X, θ_1, θ_3 ..) ...
  end for
  return θ

1.2.4 Gaussian Mixture Model for protein conformational change

In our models, the protein structures are the ensembles of M three dimensional structures with length N: X is an array size (MxNx3). The input coordinates are selected by the coordinate of carbon alpha (Cα). To avoid the heterogeneous structure, we superimpose all structures with one reference using Kabsch algorithm (Kabsch (1976)). Our first model is a generative model to generate the input structures from the target structures by the structural transformation.

$$X_{mn} \simeq R_{mk}Y_{kn} + t_{mk} \quad \text{if} \quad z_n = k\,. \tag{1.2}$$

The rotation matrix R_{mk} size (3x3) and the translation vector t_{mk} transform positions of the kth domain Y_k to the mth structure X_m with latent parameter $z_n = k$. In the update version, we reduce the number of parameter by a single target structure Y. Model in equation (1.2) above is only approximately. We model it by the Gaussian distribution with mean $R_{mk}Y_{kn} + t_{mk}$ and standard deviation σ_k:

$$p(X_{mn}|Y_{kn}, R_{mk}, t_{mk}, \sigma_k, Z_{nk} = 1) = \mathcal{N}(R_{mk}Y_{kn} + t_{mk}, \sigma_k^2) \tag{1.3}$$

where $\mathcal{N}(\mu, \sigma^2)$ indicates three-dimensional spherical Gaussian distribution has the mean μ and the standard deviation σ.
The single Gaussian distribution can not characterize complicated data, which are spread into different modes. To expand the capability of a Gaussian distribution, we use the Gaussian Mixture Model (GMM), which is a combination of Gaussian distributions.

$$GMM(X_{mn}|\theta) = \sum_{k=1}^{K} w_k \mathcal{N}(X_{mn}|\mu_k, \sigma_k^2) \tag{1.4}$$

Where θ is the set of model parameter, w_k is the weight of the k-th distribution which satisfies $\sum_{k=1}^{K} w_k = 1$. There are two challenges when using GMM. First, before using GMM, we have to specify the number of its component K. Second, both two methods EM and Gibbs sampler we use to estimate the model parameters depends on the initial step. We address the solution for these problems in Chapter 2.

The mixture model is not limited only to Gaussian distributions; in general, it can construct the model from other distributions. For example, in the second model, we introduce a two-component mixture of generalized lognormal distributions. The flexibility of the mixture model gives us a powerful tool to characterize complicated data.

1.2.5 Model for structural transitions

The rigid body model we introduce above delineates protein ensembles into conserved parts. To describe the full conformational pathway from limit experimental conformations, we can use MD Simulations (Karplus (2002), Karplus and Kuriyan (2005)). However, as we mentioned before high computational cost is the main drawback of this method. Alternative method is using Gaussian Network Model (GNM) (Tirion (1996), Haliloglu *et al.* (1997)). GNM models the macromolecule as an elastic spring network to study the conformational change. First we study the interatomic distances r in each conformational state. The GNM assume that the difference distance follow a Gaussian distribution. Because the distance as well as distance ratio is non-negative, we can expand the Gaussian distribution into the generalized lognormal distribution (GLN):

$$\mathrm{GLN}(r;\rho,\lambda,\beta) = \frac{\beta}{2\Gamma(1/\beta)\,\lambda\, r}\,\exp\{-|\ln(r/\rho)|^{\beta}/\lambda^{\beta}\} \tag{1.5}$$

GLN has three positive parameters ρ, λ, and β. β controls the shape of the distribution, $\log\rho$ is the mean, median and mode. The scale parameter λ determine the variance of the distribution $\mathrm{var}(\ln r) = \frac{\Gamma(3/\beta)}{\Gamma(1/\beta)}\,\lambda^2$. Using Bayesian inference, we can infer the parameter ρ, λ, and β from data $\{r_i\} = \{r_1, \ldots, r_n\}$. Using the Jeffrey prior, the posterior distribution is:

$$\Pr(\rho,\lambda,\beta\,|\,\{r_i\}) \propto \frac{\beta^{n-1}}{\rho\,\lambda^{n+1}\,\Gamma(1/\beta)^n}\exp\left\{-\sum_{i=1}^{n}|\ln r_i - \ln\rho|^{\beta}/\lambda^{\beta}\right\}. \tag{1.6}$$

The network model for the structural transition is describe detail in Chapter 5.

1.3 Synopsis

This thesis consists of several chapters that propose different models of protein conformational change. We begin with the first probabilistic model for segmenting protein conformational change into rigid bodies (Chapter 2). Chapter 3 is our computational web server, as well as a published dataset in the real application. In Chapter 5, we introduce our probabilistic network model for structure transition in biomolecules. Chapter 4, we introduce a new approach for model-free based on graph algorithms collaborate with Linh Dang and Prof Stephan Waack. We discuss our results, and finally, we conclude this thesis in Chapter 6, summarize our achievement, and give an outlook for our future research.

Chapter 2

Probabilistic model for detecting rigid domains in protein structures

This chapter is the first research result from my Ph.D. study. Here we present a primitive probabilistic model for segmenting ensemble protein into a set of small rigid bodies. This chapter was published in Bioinformatics journal 2016. Cited as: Nguyen and Habeck 2016.

Own contribution:

- Concept and implementation of the algorithm and the code.
- Construct parser for the Dyndom dataset for test.
- All figures, tables.
- Manuscript in parts.

Bioinformatics, 32, 2016, i710–i717
doi: 10.1093/bioinformatics/btw442
ECCB 2016

OXFORD

A probabilistic model for detecting rigid domains in protein structures

Thach Nguyen[1] and Michael Habeck[1,2,*]

[1]Felix Bernstein Institute for Mathematical Statistics in the Biosciences, University of Göttingen and [2]Max Planck Institute for Biophysical Chemistry, Göttingen 37077, Germany

*To whom correspondence should be addressed

Abstract

Motivation: Large-scale conformational changes in proteins are implicated in many important biological functions. These structural transitions can often be rationalized in terms of relative movements of rigid domains. There is a need for objective and automated methods that identify rigid domains in sets of protein structures showing alternative conformational states.

Results: We present a probabilistic model for detecting rigid-body movements in protein structures. Our model aims to approximate alternative conformational states by a few structural parts that are rigidly transformed under the action of a rotation and a translation. By using Bayesian inference and Markov chain Monte Carlo sampling, we estimate all parameters of the model, including a segmentation of the protein into rigid domains, the structures of the domains themselves, and the rigid transformations that generate the observed structures. We find that our Gibbs sampling algorithm can also estimate the optimal number of rigid domains with high efficiency and accuracy. We assess the power of our method on several thousand entries of the DynDom database and discuss applications to various complex biomolecular systems.

Availability and Implementation: The Python source code for protein ensemble analysis is available at: https://github.com/thachnguyen/motion_detection

Contact: mhabeck@gwdg.de

1 Introduction

The function of many biomolecular machines involves internal structural dynamics. Under physiological conditions, multiple conformational states are typically explored, some of which might facilitate structural transitions that are relevant for the biomolecule's function. There is a hierarchy of conformational changes in proteins, ranging from smaller internal adaptations to large-scale global rearrangements of entire domains (Henzler-Wildman *et al.*, 2007). Large-scale conformational changes are often implicated in interactions with other molecules. Therefore, to gain a deeper understanding of many cellular processes, it is crucial to detect and rationalize these structural transitions.

Many large-scale conformational changes in proteins can be described as rigid-body movements (Gerstein *et al.*, 1994). Several computational methods for detecting rigid domains in protein structures have been proposed. DynDom (Hayward *et al.*, 1997; Hayward and Berendsen, 1998) is a method for automated segmentation into rigid domains based on an analysis of pairs of alternative structures. A 3D version of the original 1D version has been developed (Poornam *et al.*, 2009). A database with automated as well as user-curated segmentations provides a rich resource for studying conformational changes in proteins (Lee *et al.*, 2003). MolMovDB also provides a large collection of conformational changes and morphs between alternative conformational states (Flores *et al.*, 2006). Spectrus (Ponzoni *et al.*, 2015) is a recent method for detecting rigid domains, and has been shown to be highly efficient and accurate in challenging analyses involving large assemblies.

Most existing methods for finding rigid domains rely on an analysis of the difference distance matrix or the matrix of distance fluctuations observed in alternative protein structures (e.g. (Abyzov et al., 2010) or (Ponzoni *et al.*, 2015)). A shortcoming of these methods is that they often lack a statistical framework for parameter inference, which makes it difficult, if not impossible to assess parameter uncertainties or to compare alternative models quantitatively. Moreover, most methods depend on algorithmic parameters, whose impact is not always intuitively clear, and for which it is difficult to find parameter settings that work for a diverse range of structures.

Here we introduce a probabilistic model to detect rigid domains in protein structures showing multiple conformational states. The model explicitly implements the notion that movements in protein structures can be rationalized in terms of rigid-body motions. We develop an efficient Markov chain Monte Carlo algorithm to estimate the model parameters within a Bayesian framework (Jaynes,

2003). The algorithm estimates the three-dimensional structures of the rigid domains as well as their location. For a particular choice of the prior probability over the segmentation variables, we obtain a Gaussian mixture model for protein ensembles (Hirsch and Habeck, 2008). We also demonstrate that our sampling algorithm can be used to detect the number of rigid domains in a data-driven fashion, circumventing the need to choose the number of rigid domains beforehand. We test our algorithm by running it in an automated mode on more than 3000 entries from the DynDom database and observe high agreement for most examples. Finally, we present an in-depth analysis of various examples of conformational changes in large biomolecular assemblies.

2 Methods

We assume that we are given M experimental protein structures represented by N atom positions X_{mn}. The $N \times 3$ matrix X_m stores all atom positions that are used for the detection of rigid domains. Typically, we use Cα positions to represent a structure, in which case N equals the size of the protein.

2.1 Gaussian mixture model for the detection of rigid motions in biomolecules

Our goal is to find K rigid domains ($1 \leq K \leq N$) into which the structures can be decomposed. The segmentation of the structures into rigid domains is encoded as a binary $N \times K$ matrix $Z \in \{0,1\}^{N \times K}$ satisfying $\sum_k Z_{nk} = 1$, i.e. the nth position can be assigned to exactly one domain k only, indicated by $Z_{nk} = 1$. Alternatively we can represent the segmentation using an integer-valued N-dimensional vector $z \in \{1, \ldots, K\}^N$, where z_n indicates the index of the domain to which the nth position has been assigned (i.e. $Z_{nz_n} = 1$).

The structure of each of the K rigid domains is represented by a $N \times 3$ matrix Y_k. We assume that the structure ensemble X is generated by rigid transformations of the domains

$$X_{mn} \simeq R_{mk} Y_{kn} + t_{mk} \quad \text{if} \quad z_n = k. \tag{1}$$

Rigid transformations involve a global rotation and translation of the domain. The transformed domains are patched together to build the full structure X_m (see Fig. 1). That is, the rotation matrix R_{mk} and the translation vector t_{mk} map positions of the kth domain Y_k onto the mth structure X_m whenever $z_n = k$.

Model (1) holds only approximately. We account for deviations due to experimental errors or shortcomings of the model by assuming a Gaussian error model, where each domain has its own error parameter σ_k:

$$p(X_{mn} | Y_k, R_{mk}, t_{mk}, \sigma_k, Z_{nk} = 1) = \mathcal{N}(R_{mk} Y_{kn} + t_{mk}, \sigma_k^2) \tag{2}$$

where $\mathcal{N}(\mu, \sigma^2)$ indicates three-dimensional spherical Gaussian distribution centered at μ with a standard deviation σ.

The complete likelihood of the entire structure ensemble is

$$p(X | Y, R, t, \sigma, Z) = \prod_k (2\pi\sigma_k^2)^{-3MN_k/2} e^{-\sum_n z_{nk} \Delta_{nk}^2 / 2\sigma_k^2} \tag{3}$$

where we introduced

$$\Delta_{nk}^2 = \sum_{m=1}^{M} ||X_{mn} - R_{mk} Y_{kn} - t_{mk}||^2, \quad N_k = \sum_n Z_{nk} \tag{4}$$

and denote sets of parameters of the same kind collectively by $R = \{R_{mk} \,|\, m = 1, \ldots, M, k = 1, \ldots, K\}$, $Y = \{Y_k \,|\, k = 1, \ldots, K\}$, etc.

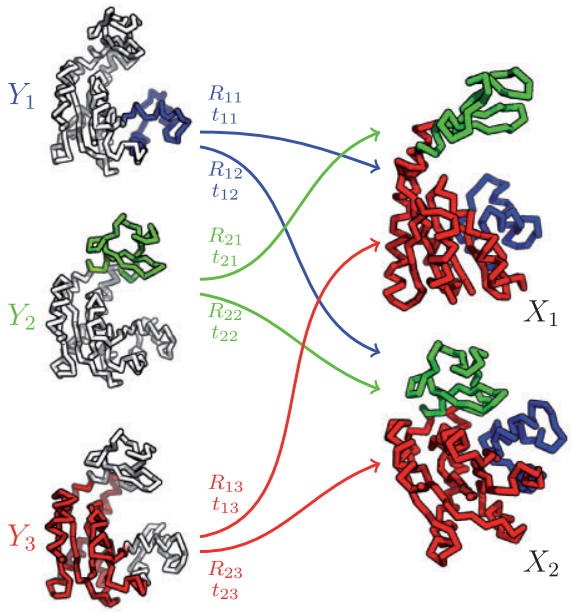

Fig. 1. Generative model for detecting rigid domains in proteins. Conformational changes in Adenylate kinase (AdK) can be described as the rigid movement of three domains (for details see Section 3.1): Y_1 (NMP), Y_2 (LID) and Y_3 (CORE) highlighted in blue, green and red, respectively. Structural regions shown in white correspond to positions that are not part of the rigid domain. The experimental structures X_1 (PDB code 4ake(A)) and X_2 (PDB code 1ake(B)) show the open and the closed state of AdK. These structures can be generated by rigidly transforming the domains Y_k by applying a rotation matrix R_{mk} and adding a translation vector t_{mk}

Bayesian inference allows us to incorporate prior information about meaningful segmentations. The simplest assumption is that z_n are independent variables that follow a categorical distribution with event probabilities $w_k \in [0,1]$, $\sum_k w_k = 1$. That is, Nw_k is the size of the kth rigid domain. For this, prior we have

$$p(Z|w) = \prod_n \left[\sum_k Z_{nk} = 1 \right] \prod_k w_k^{Z_{nk}} \tag{5}$$

where $[\cdot]$ is the Iverson bracket (i.e. $[A] = 1$ if statement A is true, $[A] = 0$ if statement A is false). For prior (5), it is possible to sum over all possible segmentations analytically. We recover the mixture model for protein ensembles, which we proposed previously (Hirsch and Habeck, 2008)

$$p(X|Y, R, t, \sigma, w) = \prod_{n=1}^{N} \sum_{k=1}^{K} w_k \prod_{m=1}^{M} \frac{e^{-\Delta_{mnk}^2 / 2\sigma_k^2}}{(2\pi\sigma_k^2)^{3/2}} \tag{6}$$

where $\Delta_{mnk} = ||X_{mn} - R_{mk} Y_{kn} - t_{mk}||$.

The assumption of independent segmentation variables is not realistic for protein structures, for which, we expect that rigid domains span multiple successive positions. Prior (5) ignores the fact that the atom positions come in a meaningful order, and would equally apply to structure ensembles in which the atom positions were scrambled. The information about the order of atom positions can be encoded in an alternative prior over the segmentation labels z_n

$$p(z_{n+1}|z_n, w) = w\,[z_{n+1} = z_n] + \frac{1-w}{K-1}[z_{n+1} \neq z_n]\,. \tag{7}$$

Prior (7) imposes sequential correlations between successive segmentation variables: With probability $w \in [0,1]$, the assignment of the next position is identical to the label of the current position.

With probability $1-w$, the label of the next position is randomly chosen among the $K-1$ alternative labels.

2.2 Gibbs sampler for probabilistic segmentation into rigid domains

To find meaningful segmentations of the structure ensemble, we have to estimate all unknown model parameters $\theta = (Y, R, t, \sigma, w)$ where w denotes a probability vector in case of prior (5) and a scalar in case of prior (7). Parameter estimation can be achieved by Markov chain Monte Carlo techniques (Neal, 1993).

We use a Gibbs sampler (Geman and Geman, 1984) for parameter estimation. In case of prior (5), the Gibbs sampler can be viewed as a stochastic version of the previously published expectation maximization algorithm [12]. To facilitate the use of a Gibbs sampler, we choose conjugate prior probabilities for the unknown parameters θ. The translation parameters t_{mk} and the positions of the rigid domains Y_{kn} follow three-dimensional spherical Gaussians:

$$p(t_{mk}|\mu_t, \sigma_t) = \mathcal{N}(\mu_t, \sigma_t^2), \quad p(Y_{kn}|\mu_Y, \sigma_Y) = \mathcal{N}(\mu_Y, \sigma_Y^2). \tag{8}$$

We do not estimate the hyperparameters $\mu_t, \sigma_t, \mu_Y, \sigma_Y$ but fix them to reasonable values: $\mu_t = \mu_Y = 0$, and $\sigma_t = \sigma_Y = 10$ (all in Å). The choices for μ_t and μ_Y are plausible because the structures have been centered. The choices for σ could be refined by using, for example, the radius of gyration, but our tests showed that the results are not sensitive to the exact choice. The prior distribution of the rotation matrices is uniform over $SO(3)$ (Habeck, 2009). The inverse variances (precisions) of the rigid domains are assumed to follow a Gamma distribution:

$$\sigma_k^{-2} \sim \mathcal{G}(a_\sigma, b_\sigma) \tag{9}$$

where $\mathcal{G}(a,b)$ indicates a Gamma distribution with shape parameter $a>0$ and scale $b>0$. Again, we fix the hyperparameters $a_\sigma = b_\sigma = 1/10$, i.e. the expected diversity of the kth domain is $\sigma_k \approx 1$ a priori.

The Gibbs sampler cycles over groups of parameters that are updated conjointly, while the other parameters are kept fixed. To implement the Gibbs sampler, we have to work out the conditional posterior distributions. The translations follow Gaussian distributions

$$t_{mk} \sim \mathcal{N}(\mu_{mk}, \sigma_{mk}^2) \tag{10}$$

where

$$\frac{1}{\sigma_{mk}^2} = \frac{N_k}{\sigma_k^2} + \frac{1}{\sigma_t^2}$$
$$\mu_{mk} = \sigma_{mk}^2 \left(\frac{1}{\sigma_k^2} \sum_n z_{nk} (X_{mn} - R_{mk} Y_{kn}) + \frac{\mu_t}{\sigma_t^2} \right)$$

The conformations of the rigid domains are sampled from Gaussian distributions

$$Y_{kn} \sim \mathcal{N}(\mu_{kn}, \sigma_{kn}^2) \tag{11}$$

where

$$\frac{1}{\sigma_{kn}^2} = \frac{M}{\sigma_k^2} + \frac{1}{\sigma_Y^2}$$
$$\mu_{kn} = \sigma_{kn}^2 \left(\frac{1}{\sigma_k^2} \sum_m R_{mk}^T (X_{mn} - t_{mk}) + \frac{\mu_Y}{\sigma_Y^2} \right)$$

The rotation matrices follow a matrix von Mises–Fisher distribution and are sampled using the algorithm from the following equation (Habeck, 2009):

$$R_{mk} \sim \exp \operatorname{tr}(A_{mk}^T R_{mk}) \tag{12}$$

where

$$A_{mk} = \frac{1}{\sigma_k^2} \sum_n z_{nk} Y_{kn} (X_{mk} - t_{mk})^T.$$

Sampling of the noise levels σ_k is achieved by simulating a Gamma distribution

$$\sigma_k^{-2} \sim \mathcal{G}(a_k, b_k) \tag{13}$$

where

$$a_k = \frac{3N_k}{2} + a_\sigma, \quad b_k = \frac{1}{2} \sum_n z_{nk} \Delta_{nk}^2 + b_\sigma.$$

To update the segmentation parameters z_n, we first collect the contributions from the data and introduce the (unnormalized) likelihood factors contributing to the probability of atom n being assigned to the kth domain

$$L_{nk} = \sigma_k^{-3M} e^{-\Delta_{nk}^2 / 2\sigma_k^2}.$$

The conditional posterior of the assignment variables depends on the choice of the prior. For the first prior (Eq. (5)) with independent segmentation variables, we have

$$z_n \sim \prod_k p_{nk}^{[z_n = k]}, \quad p_{nk} = \frac{w_k L_{nk}}{\sum_{k'} w_{k'} L_{nk'}}. \tag{14}$$

For the second prior (Eq. (7)) with sequential coupling between the segmentation variables, we have

$$z_n \sim \prod_k q_{nk}^{[z_n = k]} \tag{15}$$

where the probabilities

$$q_{nk} = \frac{L_{nk} \left((1-w)[k \neq z_{n-1}] + (K-1) w [k = z_{n-1}] \right)}{\sum_{k'} L_{nk'} \left((1-w)[k' \neq z_{n-1}] + (K-1) w [k' = z_{n-1}] \right)}$$

depend on the previous assignment except for the first position ($q_{1k} = L_{1k} / \sum_{k'} L_{1k'}$).

In case of prior (5), w are the component weights of the mixture model (6). We assume a conjugate Dirichlet prior with a single concentration parameter α, and set $\alpha = 1/K$ if not stated otherwise. The conditional posterior probability follows a Dirichlet distribution:

$$w_k \sim \mathcal{D}(\alpha + N_k) \propto \left[\sum_k w_k = 1 \right] \prod_k w_k^{N_k + \alpha - 1}. \tag{16}$$

For the second prior (7), w, the probability that two successive atoms belong to the same rigid domain, is a scalar. We assume a Beta prior, $\mathcal{B}(\alpha, 1-\alpha)$, with parameters chosen such that $\langle w \rangle = \alpha$ is close to one, reflecting the fact that in protein structures, we typically have, only few rigid domains stretching over large segments. We let $\alpha = 0.95$ in our tests based on the second prior. The conditional posterior distribution is again a Beta distribution:

$$w \sim \mathcal{B}(Q + \alpha, N - Q - \alpha) \propto w^{Q + \alpha - 1} (1-w)^{N - Q - \alpha - 1} \tag{17}$$

where $Q = \sum_{n>1} [z_n = z_{n-1}]$ is the number of times successive positions that are assigned to the same rigid segment.

2.3 Initialization of the Gibbs sampler

Similar to expectation maximization, the Gibbs sampler is only guaranteed to converge locally. Therefore, its success depends

Algorithm 1. Gibbs sampler for probabilistic segmentation of protein structures into rigid domains.

Require: Ensemble of M protein structures X_m of length N, the number of Gibbs sampling iterations n_{iter}
Initialize z, Y, R, t, σ, and w
for $i = 1, \ldots, n_{\text{iter}}$ **do**
  Sample R given z, Y, t, w, σ using Eq. (12)
  Sample Y given z, R, t, w, σ using Eq. (11)
  Sample σ given z, Y, R, t, w using Eq. (13)
  Sample w given z, Y, R, t, σ using Eq. (16) or (17)
  Sample z given Y, R, t, w, σ using Eq. (14) of (15)
  Sample t given z, Y, R, w, σ using Eq. (10)
end for
return z, Y, R, t, σ, w

strongly on the initial values for the model parameters. We initialize the model parameters as follows. We compute the average Gram matrix, $\sum_m X_m X_m^T/M$, across all members of the ensemble and use a metric embedding algorithm [7] to compute an initial estimate of Y_k based on the first three eigenvectors with largest eigenvalues.

The initial segmentation for given K is found by using spectral clustering (Uw *et al.*, 2001; Von Luxburg, 2007). To obtain an atom-by-atom similarity matrix that can be used for spectral clustering, we transform the matrix of distance fluctuations by using an exponential transform similar to (Ponzoni *et al.*, 2015).

An overview of the Gibbs sampler is presented in Algorithm 1.

2.4 Comparison of segmentations

There are two major difficulties that complicate the comparison of alternative segmentations. First, the segmentation labels can switch without changing the structure of the segmentation. Second, two segmentations can involve a different number of rigid domains.

Our first approach to compare two segmentations z and z' based on K and K' rigid domains first aims to establish a correspondence between the segmentation labels by using a linear assignment algorithm. To do so, we compute a $K \times K'$ matrix of overlaps between the labels

$$O_{kk'} = \sum_n [z_n = k]\,[z'_n = k']\,.$$

We then run the Hungarian method (Kuhn, 1955) on the cost matrix $-O_{kk'}$ to find a list of corresponding segmentation labels (k, k') that maximize the overlap of both segmentations. The average overlap of both segmentations is

$$o(z, z') = \frac{1}{N} \sum_{(k,k')} O_{kk'} \tag{18}$$

where $o(z, z') \in [0, 1]$.

A simple alternative to the linear assignment approach is to compare alternative segmentations by using a label forgetting representation based on a binary $N \times N$ matrix $B(z)$ (Adametz and Roth, 2011; McCullagh and Yang, 2008)

$$B_{nn'}(z) = [z_n = z_{n'}]\,. \tag{19}$$

It is possible to recover the segmentation from B by calculating the connected components of the undirected graph whose adjacency matrix is B. If z and z' encode two segmentations (possibly obtained for a different number of components K and K'), we use the following metric to assess their dissimilarity:

$$d(z, z') = \frac{2}{N(N-1)} \sum_{n<n'} |B_{nn'}(z) - B_{nn'}(z')| \tag{20}$$

where N is the number of atom positions (the length of the protein in case of Cα atoms). Segmentation error (20) is the average Hamming distance of the upper diagonal of B and B'. The expected segmentation error for two random segmentations is 1/2. The two comparison metrics, $o(z, z')$ and $d(z, z')$, are highly anti-correlated: a high overlap $o(z, z')$ typically entails a small segmentation error $d(z, z')$.

2.5 Practical issues

We use Cα positions to represent protein conformational states but other choices would also be possible. The segmentation model has been implemented in Python and is based on numpy, scipy and the CSB library (Kalev *et al.*, 2012). In cases where the protein sequences are not identical due to mutations, or because atom records are missing in the PDB file, we use Clustal W (Larkin *et al.*, 2007) to align the sequences. The analysis is then restricted to amino acids for which structural information is available for all conformational states (gap-less columns in the alignment). We typically run the Gibbs sampler (Algorithm 1) multiple times and select the simulation achieving highest posterior probability.

3 Results and discussion

3.1 Segmentation into rigid domains by Gibbs sampling

To illustrate our segmentation algorithm, we first analyze Adenylate kinase (AdK) for which many experimental and theoretical results about conformational changes are available. AdK catalyzes the interconversion of adenine nucleotides and is composed of three structural domains. Two smaller domains, NMP (residues 30–60) and LID (residues 115–160), are inserted into the largest domain (CORE). AdK binds ATP and AMP and converts them into two ADP molecules. The binding is facilitated by a closure of the LID and NMP binding domains.

To identify the rigid domains in AdK, we use an open conformation (PDB code 4ake) and a closed structure (PDB code 1ake). We set $K = 3$ and run the Gibbs sampler for both priors using $n_{\text{iter}} = 1000$. Figure 2(A–C) shows the resulting segmentation obtained with both priors. For comparison, we also show the difference distance matrix $|d^{1\text{ake}}_{nn'} - d^{4\text{ake}}_{nn'}|$. Both segmentations highly agree with each other. The domain boundaries found with prior (5) are CORE: 1–31, 72–117, 160–214, NMP: 31–71 and LID: 118-159. The domain boundaries found with prior (7) are CORE: 1–29, 73–116, 168–214, NMP: 30–71, LID: 117–167. The LID domain is slightly larger with the second prior due to the enforcement of sequential correlations.

The Gibbs sampler converges rapidly as indicated by the evolution of the log likelihood (Fig. 2D and E). Within 50 Gibbs sampling iterations, the posterior mode has been found. Based on this finding, we set $n_{\text{iter}} = 500$ in the remainder of the paper. Figure 2F shows the posterior histogram of the sequential coupling probability of the second prior (7). The coupling probability w scatters about an average value of 0.98.

3.2 Estimating the optimal number of rigid domains

In general, the number of rigid domains is unknown. Strategies to estimate the optimal number of domains in a data-driven fashion such as the Bayesian information criterion (BIC) (Schwarz, 1978) or cross-validation (Stone, 1974) are difficult to apply in our context.

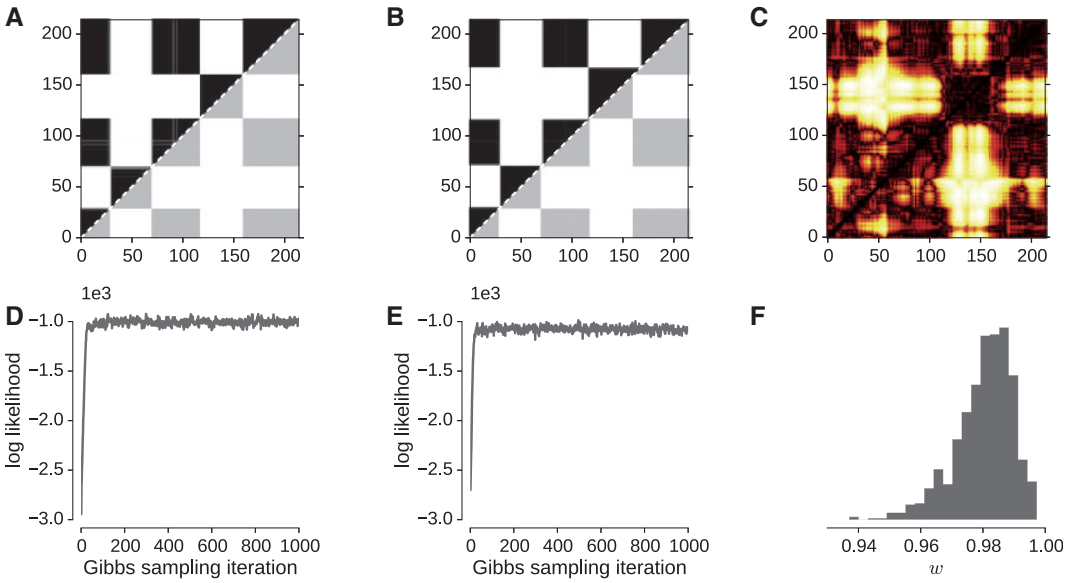

Fig. 2. Segmentation analysis of Adenylate kinase. (A) Segmentation based on prior (5) using the label forgetting representation (upper diagonal matrix). The segmentation found in the literature is shown in gray below the diagonal. (B) Label forgetting representation of the segmentation found with prior (7). (C) Difference distance matrix between open and closed conformation of AdK (shown is the exponentially transformed absolute deviation between the distances to improve the visibility). (D and E) Evolution of the log likelihood during Gibbs sampling for both priors (D: first prior (Eq. (5)), E: second prior (Eq. 7)). (F) Estimated probability w that two successive domain assignments z_n, z_{n+1} are identical

BIC assumes that the model parameters are independent, which is certainly not true for our segmentation model. Moreover, BIC is based on an approximation that is only valid if the number of data is much larger than the number of model parameters. This assumption is also violated in our application.

Cross-validation assumes that the data can be partitioned into more or less independent sets of data points that can then serve as training data and test data. This is not easily possible for protein structures, which show strong correlations between atomic positions. Figure 3A shows the results of 10-fold cross-validation for AdK. Both training and test error are almost indistinguishable and continue to decay as we increase the number of domains. Although the largest improvement in the test error is for $K = 3$, it is not clear how to choose the optimal K based on a cross-validation analysis.

Many statistical methods for choosing the number of clusters have been proposed. The gap statistic (Tibshirani *et al.*, 2001) for example, estimates the number of clusters based on within-cluster dispersion. The silhouette score (Rousseeuw, 1987) compares the distances between each cluster member and intra-cluster distances. Practical tests show that both methods work poorly for our model. Figure 3B shows the average silhouette score as a function of K. We observe a small preference for the correct number of rigid domains. However, $K = 2$ scores almost equally high, whereas $K = 4$ is preferred less. Inspection of the AdK structures and of the different distance matrix (Fig. 2C) shows that rather the opposite behavior would be desirable.

Bayesian inference provides a natural and powerful framework for model comparison (Jaynes, 2003; MacKay, 2003). Model selection is based on marginal likelihoods or *model evidences*. For the rigid domain detection problem, the relevant marginal likelihood is $p(X|K)$, i.e. the probability of the experimental protein ensemble assuming K rigid domains obtained by summing and integrating over all segmentation and model parameters.

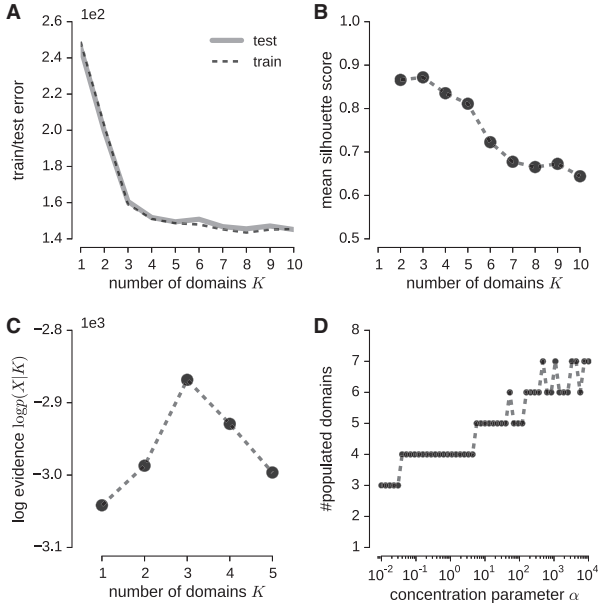

Fig. 3. Estimation of the number of rigid domains for AdK. (A) Results of 10-fold cross-validation based on random partitions of atoms into 10 disjoint sets. The training and test errors are defined as the negative logarithm of the likelihood. The value for the training set is further divided by $10 - 1 = 9$ to account for differences in the number of data points used to evaluate both errors. (B) Analysis based on the silhouette score. The silhouette score is defined for every atom position. Shown is the average score over all atoms versus the number of rigid domains K (for $K = 1$ it is not possible to define the silhouette score). (C) Log evidence estimated from five parallel tempering simulations at $K = 1, \ldots, 5$. (D) Result of prior parallel tempering in which the concentration parameter α serves as a temperature-like parameter. Shown is the number of populated domains for which $N_k = \sum_n Z_{nk} > 0$

Unfortunately, it is highly challenging to compute $p(X|K)$. One possibility is to use advanced sampling algorithms such as annealed importance sampling (Neal, 2001) or parallel tempering (Geyer, 1991; Swendsen and Wang, 1986) to estimate $p(X|K)$. But the choice of the temperature schedule is highly non-trivial for mixture models. Moreover, even if we could find a good schedule, running parallel tempering simulations on an entire database of protein conformational changes is not practical. Nonetheless, we have estimated $p(X|K)$ for K ranging between 1 and 5 for AdK using five parallel tempering simulations (Fig. 3C). These simulations are time consuming and involve several rounds of optimizing the temperature schedule until good swapping rates are obtained. The evidence clearly prefers $K=3$, consistent with the segmentation found in the literature.

A more efficient way of selecting the optimal number of domains is the 'overfitting' approach by van Havre *et al.* (2015). The idea is to work with a large number of components K, preferably much larger than the actual number of components that the data require, and to control the complexity of the model by the hyperparameter α, the concentration parameter of the component weights w_k. For large α, we enforce that all domains have more or less the same length N/K, because we know from Eq. (16) that

$$\langle w_k \rangle = \frac{N_k + \alpha}{N + K\alpha} \overset{\alpha\to\infty}{\to} \frac{1}{K}\,.$$

As we let $\alpha \to 0$, the weights are allowed to deviate from the average value $1/K$. For small α, the model will start to switch off unused components by letting $w_k \to 0$.

van Havre *et al.* (2015) propose to use 'prior parallel tempering' (PPT) with α serving as a temperature-like parameter. Here, we explored the use of PPT for segmenting AdK using a model with a maximum of $K=10$ rigid domains. The concentration parameter α varies between $\alpha = 10^4$ and $\alpha = 10^{-2}$. Figure 3D shows the results of PPT. At small α, the model only populates three domains, consistent with the literature. For larger α, it populates up to seven domains. Therefore, PPT seems to be capable of estimating the optimal number of rigid domains without the need to evaluate marginal likelihoods. However, since PPT still requires substantial computational resources, we will now investigate a shortcut to achieve automated segmentations of similar quality.

3.3 Gibbs sampling detects the optimal number of rigid domains

Instead of using PPT with annealing of the prior distribution over the component weights, we found that simply running a Gibbs sampler with a large number of rigid domains achieves a similar effect of populating only a small fraction of all K domains.

Let us illustrate this again for Adenylate kinase. We set $K=10$ and run the Gibbs sampler for both choices of the prior probability. Figure 4A and B shows that with both priors, Eqs (5) and (7), the Gibbs sampler converges to the correct segmentation. Let us contrast the performance of the Gibbs samplers with the result obtained by expectation maximization (Hirsch and Habeck, 2008). Expectation maximization fails to find a parsimonious segmentation (Fig. 4C). The Gibbs samplers can switch off domains by not populating them ($N_k=0$), whereas expectation maximization tends to use almost all domains resulting in a scattered segmentation.

The evolution of the segmentation error is shown in Figure 4D. Both Gibbs samplers achieve a similar segmentation accuracy compared with the reference found in the literature, whereas the segmentation found by expectation maximization is considerably worse.

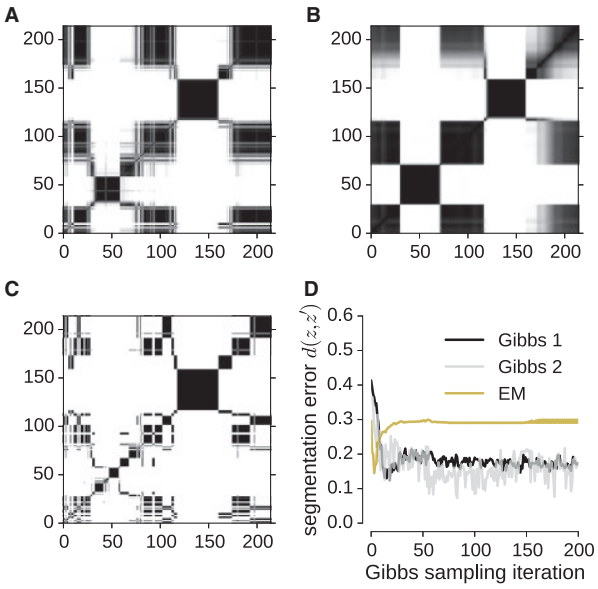

Fig. 4. Segmentation analysis of Adenylate kinase with $K=10$ rigid domains. (A and B) Label forgetting representation of the segmentation averaged over posterior samples generated with the Gibbs sampler using prior (5) and (7), respectively. (C) Segmentation found with expectation maximization. (D) Evolution of the segmentation error $d(z,z')$ [Eq. (20)] during Gibbs sampling for both priors ("Gibbs 1" and "Gibbs 2") and expectation maximization ('EM')

3.4 Large-scale benchmark

To assess the quality of our probabilistic segmentation algorithms in a systematic way, we ran both Gibbs samplers on more than 3000 examples from the DynDom database (Hayward *et al.*, 1997). Each entry comprises a pair of protein structures showing a varying degree of conformational heterogeneity. The entries were downloaded and processed automatically using a Python script. For each entry, we ran 500 iterations of Gibbs sampling based on the prior probability (5) and (7) and a maximum of 10 components ($K=10$). Before starting the Gibbs samplers, the structures were centered and superimposed onto their average structure. The last 50 segmentations generated with the Gibbs sampler were used to assess the quality of the segmentation in terms of the overlap (18) and the segmentation error (20).

Figure 5 shows histograms of the segmentation overlap and of the segmentation error averaged over the last 50 segmentations sampled with the Gibbs samplers. We observe a good agreement between the DynDom segmentation and the Bayesian segmentation for most cases. The median segmentation overlap is 90% and 87% for both priors (Eqs (5) and (7)), respectively. The median segmentation errors are 0.13 and 0.15 for both priors. Expectation maximization, in comparison, achieves a significantly lower segmentation accuracy with 43% median overlap and 0.39 median segmentation error. The second prior performs slightly worse than the first prior, which might be due to sequential couplings that are too strong in some cases, as it was already indicated by the tests on AdK.

There are a few cases where we observe a large disagreement between the segmentation found by DynDom and our rigid domain decomposition. Some of these discrepancies can be explained by the fact that DynDom tends to prefer a rather small number of rigid domains even for large protein structures. For example, the worst overlap between the segmentation based on the first prior (5) and DynDom is achieved for Apo RB69 DNA Polymerase (PDB codes

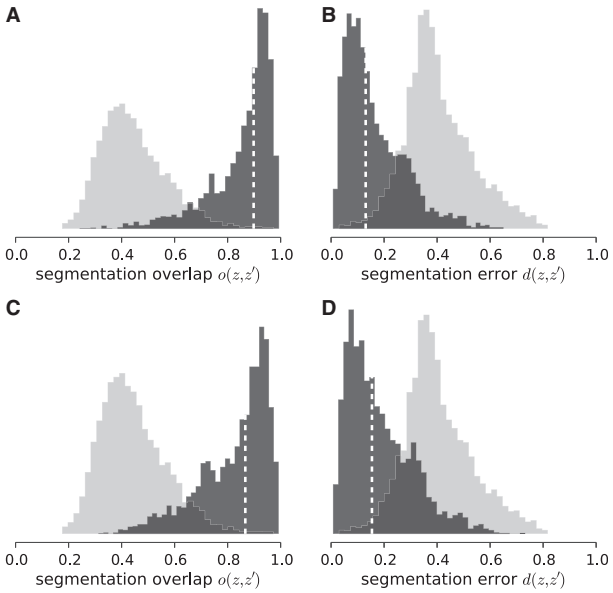

Fig. 5. Large-scale benchmark on $>$ 3000 entries from the DynDom database. (A and C) Segmentation overlap for Gibbs sampling with prior (5) and (7), respectively. (B and D) Segmentation errors for Gibbs sampling with prior 1 and 2. Median values are indicated as dashed vertical lines. The light gray histograms are the results obtained with expectation maximization

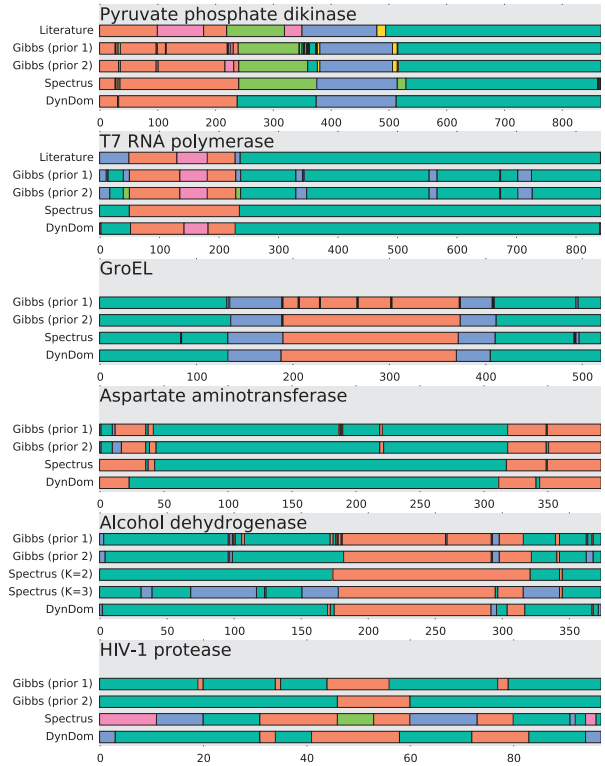

Fig. 6. Analysis of several proteins involved in large-scale conformational changes. Large assemblies: Pyruvate phosphate dikinase, T7 RNA polymerase and GroEL. Medium-sized proteins: Aspartate aminotransferase and alcohol dehydrogenase. Small protein: HIV-1 protease. Shown are the domains found by various methods and in the literature, where corresponding domains are shown in the same color

1ih7(A) and 1ig9(A)). DynDom finds three rigid domains for this pair of structures, whereas our Gibbs sampler selects a segmentation based on six or seven rigid domains, consistent with the Spectrus analysis. A similar case is multidrug transporter AcrB (PDB codes 4dx7(A), 4dx7(B)) showing the largest disagreement between the DynDom analysis and the Gibbs sampler using the second prior. DynDom finds only two rigid domains, whereas the Gibbs sampler prefers seven or eight domains, which is again supported by Spectrus.

3.5 Applications to large-scale structural transitions

Finally, we evaluate our algorithm on proteins of variable size, ranging from small proteins to large assemblies. Figure 6 and Table 1 summarize the segmentation analysis for each example.

Pyruvate phosphate dikinase (PPDK) is a large enzyme that is composed of four domains and contains two remotely located reaction centers (Lim *et al.*, 2007). We applied our Gibbs samplers to two structures of PPDK and compared the results to the annotation found in the literature (Lim *et al.*, 2007) as well as to the segmentation found by DynDom and Spectrus. The Spectrus score peaks at $K = 3$, which is very similar to our segmentations. DynDom identifies only two rigid domains and fails to identify the additional domains. Our second large-scale example is T7 RNA polymerase (T7 RNAP), which is involved in the initiation and elongation of RNA transcripts. The segmentation estimated by the Gibbs samplers is consistent with the annotation in the literature (Theis *et al.*, 2004) and the DynDom analysis. Spectrus identifies only two domains in the T7 RNAP structures. Our third example of a large assembly is the chaperonin GroEL, which provides a suitable environment for protein folding and prevents aggregation. For this example, all methods agree quite well.

We also analyzed two medium-sized proteins. For Aspartate aminotransferase (AST), again all methods highly agree with each other. Alcohol dehydrogenase (AhD) is an enzyme that decomposes alcohol into the aldehyde. This chemical action is performed by interactions between the catalytic domain of one subunit and the coenzyme-binding domain. Our Gibbs samplers detect two large domains in AhD, whose location is similar to the DynDom segmentation. The Spectrus clustering score peaks at $K = 3$ for the AhD structures, introducing one additional domain in comparison with our and DynDom's result. Spectrus' segmentation at $K = 2$ scores less well, but is more consistent with the segmentation found by DynDom and the Gibbs samplers.

Table 1. Proteins involved in large-scale conformational transitions

Protein name	PDB Code	Chain ID	Size N	K
Large assemblies				
PPDK	1kc7	A		
	2r82	A	872	4
T7 RNA polymerase	1qln	A	842	5
	1msw	D		
GroEL	1aon	A	524	3
	1aon	H		
Medium-sized proteins				
Aspartate aminotransferase (AST)	9aat	A	401	3
	1ama	A		
Alcohol dehydrogenase (AdH)	1adg	A	374	3
	2ohx	A		
Small proteins				
HIV-1 protease	3hvp	A	99	2
	4hvp	A		

Finally, we analyzed a small protein (HIV-1 protease). For this example, Spectrus introduced a large number of domains: the quality score peaks at $K = 5$. We observed a similar behavior for other small proteins, which might indicate that the Spectrus clustering score might deteriorate for small systems. Our result is consistent with the DynDom analysis and detects two major domains.

4 Conclusions

We introduce a generative model for large-scale conformational changes in proteins. A Bayesian framework allows us to estimate the model parameters efficiently, which makes our domain detection approach suited for large-scale applications. The results are consistent with segmentations found by alternative approaches for the detection of rigid domains such as DynDom (Hayward *et al.*, 1997; Hayward and Berendsen, 1998) and Spectrus (Ponzoni *et al.*, 2015). An important parameter of any segmentation approach is the number of rigid domains. In our previous implementation based on expectation maximization (Hirsch and Habeck, 2008), we had to set this parameter manually. Here we show that it is possible to estimate the number of rigid domains in a data-driven fashion by using Gibbs sampling.

Our approach has various practical and conceptual advantages over existing approaches. First, it is based on an intuitive model, which implements the idea that structural transitions can be decomposed into the translation and rotation of rigid domains relative to each other. Our model is a *generative model* and could, therefore, also be used to sample new or intermediate conformational states. This opens new avenues for applications in structural refinement or modeling of protein structures based on the experimental data. Second, we provide a clean and objective framework for parameter inference, which separates model parameters from algorithmic parameters. In addition to parameter estimates, we also provide an assessment of parameter uncertainties. Third, it is possible to incorporate prior knowledge, for example, about the location of hinges. Finally, symmetric oligomers constitute an important class of oligomeric proteins. The current model does not take internal symmetries into account but models either individual chains or the entire assembly. In a future extension of our model, we will support the decomposition of symmetric assemblies by taking into account the symmetry group.

Funding

This work was supported by Deutsche Forschungsgemeinschaft (DFG) grant SFB860 TP B9.

Conflict of Interest: none declared.

References

Abyzov,A. *et al.* (2010) RigidFinder: a fast and sensitive method to detect rigid blocks in large macromolecular complexes. *Proteins*, **78**, 309–324.

Adametz,D. and Roth,V. (2011) Bayesian partitioning of large-scale distance data. *Nips* **2011**, 1368–1376.

Flores, S. *et al.* (2006) The Database of Macromolecular Motions: new features added at the decade mark. *Nucleic Acids Res.*, **34**, 296–301.

Geman, S. and Geman, D. (1984) Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.*, **PAMI-6**, 721–741.

Gerstein, M. *et al.* (1994) Structural mechanisms for domain movements in proteins. *Biochemistry*, **33**, 6739–6749.

Geyer, C.J. (1991) Markov chain Monte Carlo maximum likelihood. In: *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface*, pp. 156–163.

Gower, J. (1966) Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika*, **53**, 325–338.

Habeck, M. (2009) Generation of three-dimensional random rotations in fitting and matching problems. *Comput. Stat.*, **24**, 719–731.

Hayward, S. and Berendsen, H.J. (1998) Systematic analysis of domain motions in proteins from conformational change: new results on citrate synthase and T4 lysozyme. *Proteins*. **30**, 144–154.

Hayward, S. *et al.* (1997) Model-free methods of analyzing domain motions in proteins from simulation: a comparison of normal mode analysis and molecular dynamics simulation of lysozyme. *Proteins: Struct. Funct. Genet.*, **27**, 425–437.

Henzler-Wildman, K.A. *et al.* (2007) A hierarchy of timescales in protein dynamics is linked to enzyme catalysis. *Nature*, **450**, 913–916.

Hirsch, M. and Habeck, M. (2008) Mixture models for protein structure ensembles. *Bioinformatics*, **24**, 2184–2192.

Jaynes, E.T. (2003) *Probability Theory: The Logic of Science*. Cambridge University Press, Cambridge UK.

Kalev, I. *et al.* (2012) CSB: a Python framework for structural bioinformatics. *Bioinformatics*, **28**, 2996–2997.

Kuhn, H.W. (1955) The hungarian method for the assignment problem. *Naval Res. Logistics Quarterly*, **2**, 83–97.

Larkin, M.A. *et al.* (2007) Clustal W and Clustal X version 2.0. *Bioinformatics*, **23**, 2947–2948.

Lee, R.A. *et al.* (2003) The DynDom database of protein domain motions. *Bioinformatics*, **19**, 1290–1291.

Lim, K. *et al.* (2007) Swiveling domain mechanism in pyruvate phosphate dikinase. *Biochemistry*, **46**, 14845–14853.

MacKay, D.J.C. (2003) *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, Cambridge UK.

McCullagh, P. and Yang, J. (2008) How many clusters? *Bayesian Anal.*, **1**, 101–120.

Neal, R.M. (1993) Probabilistic inference using Markov chain Monte Carlo methods. *Technical report CRG-TR-93-1*. Department of Computer Science, University of Toronto.

Neal, R.M. (2001) Annealed importance sampling. *Stat. Comput.*, **11**, 125–139.

Ponzoni, L. *et al.* (2015) SPECTRUS: a dimensionality reduction approach for identifying dynamical domains in protein complexes from limited structural datasets. *Structure (London, England: 1993)*, **23**, 1516–1525.

Poornam, G.P. *et al.* (2009) A method for the analysis of domain movements in large biomolecular complexes. *Proteins: Struct. Funct. Bioinform.*, **76**, 201–212.

Rousseeuw, P.J. (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, **20**, 53–65.

Schwarz, G. (1978) Estimating the dimension of a model. *Ann. Stat.*, **6**, 461–464.

Stone,M. (1974) Cross-validatory choice and assessment of statistical predictions. *J. R. Stat. Soc. Ser. B (Methodol.)*, **36**, 111–147.

Swendsen, R.H. and Wang, J.S. (1986) Replica Monte Carlo simulation of spin glasses. *Phys. Rev. Lett.*, **57**, 2607–2609.

Theis, K. *et al.* (2004) Topological and conformational analysis of the initiation and elongation complex of t7 RNA polymerase suggests a new twist. *Biochemistry*, **43**, 12709–12715.

Tibshirani, R. *et al.* (2001) Estimating the number of clusters in a data set via the gap statistic. J. R. Stat. Soc. B, **63**, 411–423.

Uw, S. *et al.* (2001) On spectral clustering: analysis and an algorithm. *Adv. Neural Inf. Process. Syst.*, **14**, 849–856.

van Havre, Z. *et al.* (2015) Overfitting Bayesian mixture models with an unknown number of components. *PLoS One*, **10**, e0131739.

Von Luxburg, U. (2007) A tutorial on spectral clustering. *Stat. Comput.*, **17**, 395–416.

Chapter 3

PRISM Server: a webserver of Probabilistic RIgid Segmentation Model for segmenting protein structure

This chapter is the implementation of previous work to the webservice. Here we present a modified probabilistic model for segmenting ensemble protein into a set of small rigid bodies, a precomputed dataset of segmented protein was introduced. This manuscript is in preparation.

Own contribution:

- Concept and implementation of the algorithm and the code.
- Construct the Dyndom dataset for test
- All figures, tables
- Manuscript

Chapter summary

Motivation: We present PRISM server, a computational web and a dataset of protein structures segmented into rigid bodies. Currently two algorithms the Expectation Maximization (EM) and stochastic Gibbs sampler are implemented.
Availability: : The web server is freely available at http://prism.stochastik.math.uni-goettingen.de.

Introduction

Proteins are elastic life machines that perform critical tasks at the molecular level of living cells. Those protein structures are observed in different conformational states. To support the understanding of protein structure, we implemented a computational web by our previous work (Nguyen and Habeck (2016), Nguyen and Habeck (2020) in preparation). We believe the webserver and precomputed dataset of protein rigid bodies will support us in understanding the biological process from underlying structural data.

System and Methods

Computational webserver workflow

PRISM Server is a user-friendly webserver for computational tasks segmenting protein into rigid bodies. We use both Biopython parser (Cock *et al.* (2009)) and CSB utilizing function (Kalev *et al.* (2012)) and to retrieve and preprocess the protein structures. The entire ensembles aligned by Clustal Omega (Sievers *et al.* (2011)), an average estimation is used to process missing structure. The computational work use one of two approaches the Gibbs sampler and the expectation maximization run on server side.

The result is exported into an interactive JSMol session, which is supported by most web browsers, user can modify the view and store it into the local machine, the result example is shown in Figure 3.1. The server is built with the modern Django web framework, using JSON and SQLlite database, the applications are written in Python.

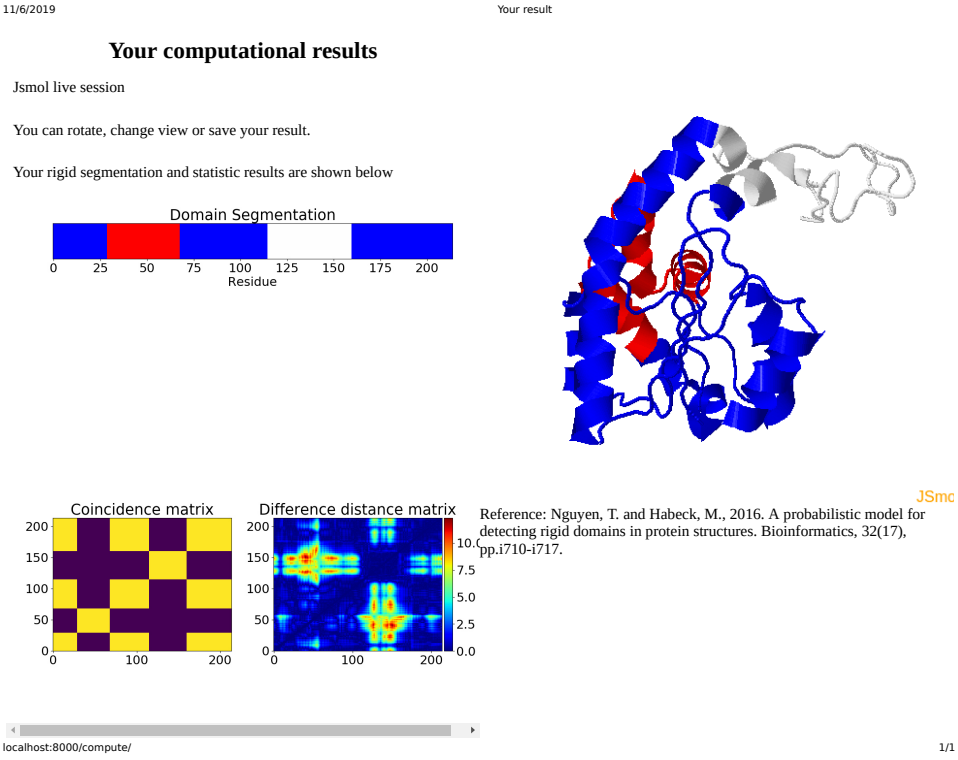

Figure 3.1: Detail about the result interface, the top left panel shown the segmentation by color, the lower left showed the average difference distance matrices of entire ensembles and the segmentation encoding matrix

Precomputed dataset

A precomputed data set is deposited at the same address. We acquire and cluster protein structure by using 95% sequential similarity cutoff threshold, the protein identification list from Blastclust (Altschul *et al.* (1997)). To reduce the redundancy of protein ensembles, we cluster the protein structure by running the DBSCAN algorithms (Ester *et al.* (1996)) using RMSD as a distance metric, and our refined result contains 2660 entries.

Conclusion and future work

We develop the PRISM webserver and dataset for segmenting protein ensembles into rigid bodies. Using our algorithms, users can extract the flex region, hinges, and rigid region

of proteins. We plan to add the database connection to another accessible bioinformatics database, enabling the automated updating of our database. Integrate our structural model with the sequential information, as well as constructing the transitional pathway from different conformational states.

Acknowledgments and Funding

This work has been supported by Deutsche Forschungsgemeinschaft (DFG) SFB 860/TP B09.
Conflict of Interest: none declared.

Bibliography

Abyzov, A., Bjornson, R., Felipe, M., and Gerstein, M. (2010). *PROTEINS: Structure, Function, and Bioinformatics*, **78**(2), 309–324.

Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997). *Nucleic acids research*, **25**(17), 3389–3402.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). *Journal of statistical mechanics: theory and experiment*, **2008**(10), P10008.

Cock, P. J., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B., *et al.* (2009). *Bioinformatics*, **25**(11), 1422–1423.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). *J. R. Stat. Soc. B*, **39**, 1–38.

Emekli, U., Schneidman-Duhovny, D., Wolfson, H. J., Nussinov, R., and Haliloglu, T. (2008). *Proteins: Structure, Function, and Bioinformatics*, **70**(4), 1219–1227.

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

Flores, S. C. and Gerstein, M. B. (2007). *BMC bioinformatics*, **8**(1), 215.

Frank, J. (2002). *Annu Rev Biophys Biomol Struct*, **31**, 303–319.

Geman, S. and Geman, D. (1984). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-6**(6), 721–741.

Gerstein, M. and Krebs, W. (1998). *Nucleic acids research*, **26**(18), 4280–4290.

Gerstein, M., Lesk, A. M., and Chothia, C. (1994). *Biochemistry*, **33**(22), 6739–6749.

Haliloglu, T., Bahar, I., and Erman, B. (1997). *Physical review letters*, **79**(16), 3090.

Hanson, J. A., Duderstadt, K., Watkins, L. P., Bhattacharyya, S., Brokaw, J., Chu, J. W., and Yang, H. (2007). *Proc Natl Acad Sci USA*, **104**, 18055–18060.

Hayward, S. and Berendsen, H. J. (1998). *Proteins Structure Function and Genetics*, **30**(2), 144–154.

Henzler-Wildman, K. and Kern, D. (2007). *Nature*, **450**, 964–972.

Hinsen, K. (1998). *Proteins: Structure, Function, and Bioinformatics*, **33**(3), 417–429.

Hirsch, M. and Habeck, M. (2008). *Bioinformatics*, **24**, 2184–2192.

Hrabe, T., Li, Z., Sedova, M., Rotkiewicz, P., Jaroszewski, L., and Godzik, A. (2015). *Nucleic acids research*, **44**(D1), D423–D428.

Kabsch, W. (1976). *Acta Cryst.*, **A32**, 922–923.

Kalev, I., Mechelke, M., Kopec, K. O., Holder, T., Carstens, S., and Habeck, M. (2012). *Bioinformatics*, **28**(22), 2996–2997.

Karplus, M. (2002). Molecular dynamics simulations of biomolecules.

Karplus, M. and Kuriyan, J. (2005). *Proceedings of the National Academy of Sciences*, **102**(19), 6679–6685.

Keating, K. S., Flores, S. C., Gerstein, M. B., and Kuhn, L. A. (2009). *Protein Science*, **18**(2), 359–371.

Kendrew, J. C., Bodo, G., Dintzis, H. M., Parrish, R., Wyckoff, H., and Phillips, D. C. (1958). *Nature*, **181**(4610), 662–666.

Monzon, A. M., Juritz, E., Fornasari, M. S., and Parisi, G. (2013). *Bioinformatics*, **29**(19), 2512–2514.

Murzin, A. G., Brenner, S. E., Hubbard, T., and Chothia, C. (1995). *J. Mol. Biol.*, **247**, 536–540.

Nguyen, T. and Habeck, M. (2016). *Bioinformatics*, **32**(17), i710–i717.

Nguyen, T. and Habeck, M. (2020). **in preparation**.

Nichols, W. L., Rose, G. D., Ten Eyck, L. F., and Zimm, B. H. (1995). *Proteins: Structure, Function, and Bioinformatics*, **23**(1), 38–48.

Orengo, C. A., Michie, A. D., Jones, S., Jones, D. T., Swindells, M. B., and Thornton, J. M. (1997). *Structure*, **5**(8), 1093–1109.

Ponting, C. P. and Russell, R. R. (2002). *Annual review of biophysics and biomolecular structure*, **31**(1), 45–71.

Ponzoni, L., Polles, G., Carnevale, V., and Micheletti, C. (2015). *Structure*, **23**(8), 1516–1525.

Sievers, F., Wilm, A., Dineen, D., Gibson, T. J., Karplus, K., Li, W., Lopez, R., McWilliam, H., Remmert, M., Söding, J., *et al.* (2011). *Molecular systems biology*, **7**(1), 539.

Sonnhammer, E. L., Eddy, S. R., and Durbin, R. (1997). *Proteins: Structure, Function, and Bioinformatics*, **28**(3), 405–420.

Tauchert, M. J., Fourmann, J.-B., Lührmann, R., and Ficner, R. (2017). *Elife*, **6**.

Tirion, M. M. (1996). *Physical review letters*, **77**(9), 1905.

Traag, V. A., Van Dooren, P., and Nesterov, Y. (2011). *Physical Review E*, **84**(1), 016114.

Voet, D. and Voet, J. G. (2010). *Biochemistry*. John Wiley & Sons.

Von Luxburg, U. (2007). *Statistics and computing*, **17**(4), 395–416.

Wriggers, W. and Schulten, K. (1997). *Proteins Structure Function and Genetics*, **29**(1), 1–14.

Wüthrich, K. (1976). *NMR in biological research: peptides and proteins*. North-Holland Amsterdam.

Wüthrich, K. (2001). *Nature Structural & Molecular Biology*, **8**(11), 923.

Xu, Z., Horwich, A. L., and Sigler, P. B. (1997). *Nature*, **388**(6644), 741–750.

Chapter 4

A graph-based algorithm for detecting rigid domains in protein structures

We develop a new domain segmentation algorithm that is capable of analyzing the entire structure database efficiently. This method does not require the choice of protein-dependent tuning parameters, such as the number of rigid domains. Graph clustering algorithms allow us to reduce the graph and run the Viterbi algorithm on the associated line graph.
Cited as: Dang et al. 2020. (submitted)

Own contribution:

- Support the implementation of the algorithm.
- Construct the Dyndom dataset and code for test.
- Figure 3.
- Manuscript in parts.

Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

OXFORD

Subject Section

A graph-based algorithm for detecting rigid domains in protein structures

Linh Dang [1,*], Thach Nguyen [2], Michael Habeck [2,3,4] and Stephan Waack [1]

[1]Institute of Computer Science, University of Göttingen, [2]Felix Bernstein Institute for Mathematical Statistics in the Biosciences, University of Göttingen, [3]Max Planck Institute for Biophysical Chemistry, Göttingen 37077, Germany, [4]Microscopic Image Analysis Group, University Hospital Jena, Germany.

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Conformational transitions are implicated in the biological function of many proteins. Structural changes in proteins can be described approximately as the relative movement of rigid domains against each other. Despite previous efforts, there is a need to develop new domain segmentation algorithms that are capable of analysing the entire structure database efficiently and do not require the choice of protein-dependent tuning parameters such as the number of rigid domains.
Results: We develop a graph-based method for detecting rigid domains in proteins. Structural information from multiple conformational states is represented by a graph whose nodes correspond to amino acids. Graph clustering algorithms allow us to reduce the graph and run the Viterbi algorithm on the associated line graph to obtain a segmentation of the input structures into rigid domains. In contrast to many alternative methods, our approach does not require knowledge about the number of rigid domains. Moreover, we identified default values for the algorithmic parameters that are suitable for a large number of conformational ensembles. We test our algorithm on examples from the DynDom database and illustrate our method on various challenging systems whose structural transitions have been studied extensively.
Availability: The Python source code is available at https://github.com/dtklinh/Protein-Rigid-Domains-Estimation
Contact: ldang1@gwdg.de
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Proteins are molecular machines that are involved in a large variety of biological processes. Protein function is often driven by large-scale structural transitions (Henzler-Wildman and Kern, 2007). Experimental methods for biomolecular structure determination such as X-ray crystallography, NMR and cryo-electron microscopy have been used to determine thousands of atomic structures of proteins in different conformational states. A powerful approach to understand structural transitions in proteins is to decompose structures of different states into rigid domains and classify protein movements by hinge and shear motions of these structural domains (Gerstein *et al.*, 1994).

Given the large number of available protein structures, we need computational methods that identify structurally conserved domains in a set of alternative structures in an automated fashion with minimal user intervention. For example, one could use the software to study molecular dynamics trajectories at the level of rigid domains to gain an understanding of large-scale movements, or identify important active sites located at the interface between rigid domains.

A number of computational methods for detecting rigid domains in protein structures have been developed. Dyndom (Hayward and Berendsen, 1998) identifies rigid domains by clustering a set of rotation vectors. Hingefind (Wriggers and Schulten, 1997) focuses on the detection of hinge residues, which are detected via differences in bending angles. RigidFinder (Abyzov *et al.*, 2010) finds rigid domains via a dynamic programming algorithm that optimizes the rigidity of structural segments extracted from two conformational states. These methods are limited to two input structures and require the selection of a cutoff parameter (Abyzov *et al.*, 2010), which can impact the results quite strongly. Spectrus

(Ponzoni *et al.*, 2015) applies spectral clustering to distance fluctuations and supports multiple input structures. However, the number of clusters relies on a quality score, which sometimes gives ambiguous results. Probabilistic approaches (Hirsch and Habeck, 2008; Nguyen and Habeck, 2016) segment protein structures into rigid domains as part of a generative probabilistic model. The model parameters, including the segmentation, are inferred with expectation maximization or Gibbs sampling. However, choosing the initial parameters as well as the number of rigid segments is still a critical issue, because both algorithms explore parameter space only locally, and can therefore require many restarts from different initial conditions.

A more ambitious goal is to predict rigid domains from a single structure by, for example, molecular dynamic simulation or an elastic network model that can both be used to generate a set of alternative conformational states. HingeProt (Emekli *et al.*, 2008) and Domain Finder (Hinsen, 1998) use an elastic network model to predict hinge residues by analyzing the correlation between selected pairs of eigenvectors of the correlation matrix. However, in general it is unclear which modes contribute most strongly to the movement, in particular if a conformational change involves multiple modes. FlexOracle (Flores and Gerstein, 2007) finds hinge positions by identifying split points with minimal energetic impact.

Despite the rich literature on methods for rigid-domain detection in protein structures, there is still a need for algorithms that are robust, reliable, able to handle high-throughput data and yet do not require extensive parameter tunning. Here, we introduce a graph-based method that infers a binary labeling that encodes if pairs of amino acids belong to identical or different rigid domains. Our algorithm proceeds in two stages: First, we construct a protein graph based on spatial proximity, which we cluster using the Louvain algorithm to obtain a coarse-grained graph of reduced size. Second, edges in the reduced graph are labeled by applying a line graph transformation along with the general Viterbi algorithm. We benchmark our algorithm on 487 entries of the DynDom database and find a high agreement with the reference segmentation. In addition, we also present a detailed analysis of various proteins that show a large variety of conformational transitions and compare our results to other methods.

2 Methods

We organize the Methods section as the following. First of all, we present a toy example and a motivation behind our method. Secondly, we describe crucial parts and techniques we use to build up the algorithm. And last but not least we present the final algorithm.

2.1 Toy example and a motivation

The crucial shortcoming of several existing methods such as (Ponzoni *et al.*, 2015; Hayward *et al.*, 1997; Nguyen and Habeck, 2016) comes from the requirement of prior knowledge of how many rigid domains in the target protein which is not always available (illustrated by Figure 1A). To overcome that, we present another approach (Figure 1B) which we binarily label edges instead of vertices. If a graph belongs to exponential models (the value of a function defined on a graph is the product of its functions defined on vertices and edges), its label could be inferred by general Viterbi algorithm (Dong *et al.*, 2014) which is heuristically able to find the most probable labels. However, we are only interested in edges' labels. Thus, we utilize the trick of line graph transformation illustrated in Figure 2 (Evans and Lambiotte, 2010). Instead of directly calculating the labels of edges in the protein graph, we calculate the vertices' label of its corresponding line graph whose vertices are edges of the protein graph by the mean of general Viterbi algorithm (Dong *et al.*, 2014).

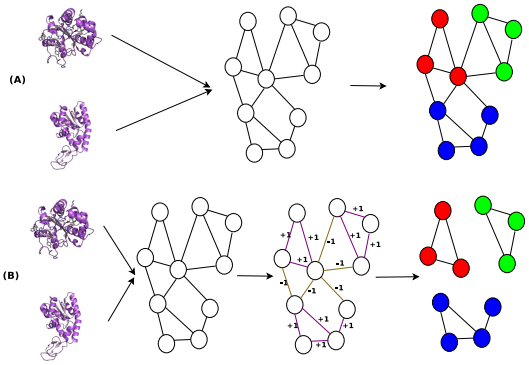

Fig. 1. Illustration the workflows of two different approaches. Panel(A) From left to right: protein conformations; a protein graph; given the number of rigid domains (number of colors), the algorithm produces the segmentation. Panel(B) From left to right: protein conformations; a protein graph; an algorithm estimating edges' labels (+1/-1: two incident vertices belong to the same domain/different domains); final segmentation by removing the negative edges.

2.2 Notations

Our algorithm aims to infer a rigid-domain segmentation from $M > 1$ conformational states of a protein π. Each conformational state is encoded by a $N \times 3$ matrix $X \in \mathbb{R}^{N\times 3}$ whose rows are the 3D coordinates of representative atoms (typically Cα atoms), i.e. $X_{m,n}$ is the position of the n-th atom in the m-th conformation. Every conformational state gives rise to a symmetric $N \times N$ distance matrix $D^{(m)}$:

$$D^{(m)}_{k,l} := \|X_{m,k} - X_{m,l}\| \qquad (k,l = 1,2,\ldots,N), \tag{1}$$

where $\|\cdot\|$ denotes the Euclidian norm.

We encode the conformational variability across all M structures X_m through a *protein graph*

$$\mathcal{PG}_\pi = (\mathcal{A}_\pi, \mathcal{E}_\pi) \tag{2}$$

whose vertices $\mathcal{A}_\pi$ are the atoms $\{1,2,\ldots,N\}$. An edge between atoms k,l is introduced if and only if

$$\max_{m=1,2,\ldots,M} D^{(m)}_{k,l} \le \delta\,. \tag{3}$$

We ran tests with $\delta = 7.5$, 10.5 and 13.5 Å. To decide if a subset $\mathcal{A} \subseteq \mathcal{A}_\pi$ is rigid, we use the criterion

$$\mathrm{RMSD}(\mathcal{A}) := \max_{m,m'=1,2,\ldots,M} \mathrm{RMSD}_{\mathcal{A}}(X_m, X_{m'}) < 3.5\,\text{Å} \tag{4}$$

where $\mathrm{RMSD}_{\mathcal{A}}(X_m, X_{m'})$ is the root mean square deviation (RMSD) (Kabsch, 1976) between conformations X_m and $X_{m'}$ reduced to atoms in $\mathcal{A}$.

2.3 Coarse graining of the protein graph

Rigid domains form densely connected subsets of nodes in the protein graph.

Due to the quadratic growth of the vertices in line graph, it is computationally infeasible to work directly on the protein graph. To reduce the size of the protein graph, we run the Louvain algorithm (Blondel *et al.*, 2008; Traag *et al.*, 2011; Traag, 2015) that partitions the nodes $\mathcal{A}_\pi$ into

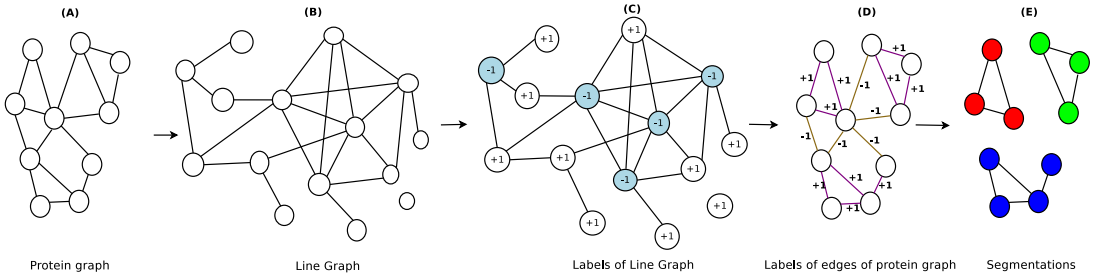

Fig. 2. Our graph-based workflow of a toy example. (A) Protein graph derived from protein states. (B) Line graph derived from the protein graph. (C) Label of line graph estimated via general Viterbi algorithm. (D) Corresponding edges' labels in the protein graph. (E) The final segmentation is achieved by removing the negative edges in the protein graph.

communities. The parameters of the Louvain algorithm are chosen such that the communities

- are small enough to include, with a few exceptions, amino acids that are part of the same rigid domain (i.e. criterion (4) is met for every community);
- are large enough to make the MAP prediction (10) feasible.

If $\mathcal{C}_\pi$ is a partition found by the Louvain algorithm, the *coarse-grained graph*

$$\mathcal{CG}_\pi = (\mathcal{C}_\pi, \mathcal{V}_\pi) \tag{5}$$

links two communities c_1 and c_2 by an undirected edge $(c_1, c_2) \in \mathcal{V}_\pi$ if at least one pair of amino acids $a_1 \in c_1, a_2 \in c_2$ is linked in the protein graph: $(a_1, a_2) \in \mathcal{E}_\pi$. Obviously, the coarse-grained graph $\mathcal{CG}_\pi$.

The mean variance of all distances between two communities c_1 and c_2 defined by

$$\text{mean-var}_D(c_1, c_2) := \frac{1}{|c_1||c_2|(M-1)} \sum_{a_1 \in c_1} \sum_{a_2 \in c_2} \sum_{m=1}^{M} \left(D^{(m)}_{a_1,a_2} - \frac{1}{M} \sum_{m'=1}^{M} D^{(m')}_{a_1,a_2} \right)^2 \tag{6}$$

is a key quantity of our method.

2.4 Outlier detection

Our assumption is that the bigger the mean variance between two vertices in the coarse-grained graph is, the more likely they belong to different domains. However, it is not obvious how to define that mapping which is independently valid across proteins. To overcome that issue, we hypothesize that the values of mean variance among vertices which belong to same domain follow a certain distribution and the ones from different domains are the outliers. In this specific study, the outliers are always the ones whose values are bigger than the rest. However, the set of outliers could be relaxed by enlarging it toward the highest value of data points in the assumed distribution. The outliers detection method described in the following plays an important role to define functions on vertices and edges in the line graph.

We use 11 predicates

$$\text{outlier}_{\mu,D}(x_i) \in \{-1, +1\} \, (\mu = 1, \ldots, 11, \; i = 1, \ldots, k)$$

to detect outliers in a set $D = \{x_1, \ldots, x_k\}$ of k real-valued data points where

$$\text{outlier}_{\mu,D}(x_i) = -1$$

indicates that x_i is an outlier in D according to the μ-th criterion, whereas $\text{outlier}_{\mu,D}(x_i) = +1$ indicates the opposite.

The first outlier predicate is a robust variant of the Z-score (Iglewicz and Hoaglin, 1993) defined by

$$\tilde{Z}_i := 0.6745 \, \frac{x_i - \text{MED}}{\text{MAD}} \tag{7}$$

where MED is the sample median and MAD the sample median absolute deviation of D. A shortcoming of the standard Z-score is that it is based on the sample mean and variance of D which are not robust against outliers. Following Iglewicz and Hoaglin (1993) we define

$$\text{outlier}_{1,D}(x_i) = -1 \quad :\Longleftrightarrow \quad \left|\tilde{Z}_i\right| > 3.5. \tag{8}$$

For $\mu = 2, 3, \ldots, 11$, we set

$$\text{outlier}_{\mu,D}(x_i) = -1 \; :\Longleftrightarrow \; x_i \text{ is in the top } 5(\mu - 1)\% \text{ of } D. \tag{9}$$

Thus, outlier models with larger index μ will detect a growing set of outliers.

2.5 A short introduction into CRFs

Let us consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{N})$ whose nodes we call *sites* and $\mathcal{V} = \{1, 2, \ldots, m\}$ without loss of generality. Sites are labeled by elements of the finite set $\mathcal{B}$. Words of length ℓ over the finite alphabet $\mathcal{O}$ are called *observations*. $\mathcal{N}$ is the set of edges in the site graph $\mathcal{G}$. The neighborhood $\mathcal{N}_i \subseteq \mathcal{V}$ of site $i \in \mathcal{V}$ consists of all sites $j \in \mathcal{V}, j \neq i$ that are linked to i by an edge in $\mathcal{N}$. Obviously $i \notin \mathcal{N}_i$. For every label sequence $\boldsymbol{y} \in \mathcal{B}^m$ and subset $I \subseteq \mathcal{V}$, $\boldsymbol{y}_I$ denotes the partial labeling of sites in I: $\boldsymbol{y}_I := \{(i, y_i) \,|\, i \in I\}$.

A pair $(\boldsymbol{X}, \boldsymbol{Y})$ composed of a random observation $\boldsymbol{X} \in \mathcal{O}^\ell$ and a random label sequence $\boldsymbol{Y} \in \mathcal{B}^m$ realizes a feature-based exponential model if the conditional probability $\mathrm{p}(\boldsymbol{y} \mid \boldsymbol{x})$ of all pairs $(\boldsymbol{x}, \boldsymbol{y})$ is

$$\mathrm{p}(\boldsymbol{y} \mid \boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \exp\left(\sum_{\gamma=1}^{c} \sum_{|I|=\gamma} \Psi^{(\gamma)}(\boldsymbol{y}_I, \boldsymbol{x}) \right),$$

where

$$Z(\boldsymbol{x}) := \sum_{\boldsymbol{y}' \in \mathcal{B}^m} \exp\left(\sum_{\gamma=1}^{c} \sum_{|I|=\gamma} \Psi^{(\gamma)}(\boldsymbol{y}'_I, \boldsymbol{x}) \right).$$

$\sum_{|I|=\gamma}$ denotes a sum over all cliques I of size γ in $\mathcal{G}$; c is the maximum clique size. For every clique size $\gamma \leq c$, the function $\Psi^{(\gamma)}(\boldsymbol{y}_I, \boldsymbol{x})$ is the feature of cliques of size γ. Under very weak assumptions feature-based

exponential models coincide with the class of *conditional random fields* where at every site i the label is conditionally independent of the labels outside $\mathcal{N}_i$ given the observation and the labels of $\mathcal{N}_i$.

The labeling problem is solved by computing a labeling sequence

$$\boldsymbol{y}^* := \operatorname*{argmax}_{\boldsymbol{y}\in\mathcal{B}^m} \mathrm{p}\left(\boldsymbol{y} \mid \boldsymbol{x}\right) \tag{10}$$

that achieves maximum posterior probability (MAP prediction). In general, MAP prediction is **NP**-hard. This is even true if only site and edge features are considered as in our segmentation model (Section 2.6). In particular, the variant of the Viterbi algorithm devised by Dong *et al.* (2014) has an exponential running time. Only if the underlying site graph is small enough, it can be used within a feasible time bound. We therefore introduce a coarse-graining step to reduce the size of the protein graph. The coarse-grained protein graph serves as a site graph for our CRF-based rigid domain prediction.

2.6 CRF-based prediction of rigid domains

Our CRF-based prediction algorithm for rigid domains is best understood as a recursive procedure RDP-CRF where each recursive call refers to a specific feature-based exponential model. Every RDP-CRF call receives a site graph $\mathcal{G} = (\mathcal{V}, \mathcal{N})$ such that $\mathrm{RMSD}\left(\mathcal{V}\right) \geq 3.5\text{Å}$ and an outlier model index μ as inputs and tries to expand a list $\mathfrak{RD}$ of pairwise disjoint subsets of amino acids in $\mathcal{A}$. Initially, $\mathfrak{RD}$ is empty.

The *initial site graph* is the *line graph*

$$\text{line-graph}\big(\mathcal{CG}_\pi\big) = \big(\mathcal{V}_\pi, \mathcal{N}_\pi\big) \tag{11}$$

of the coarse-grained graph $\mathcal{CG}_\pi$ defined in (5). The edges of the coarse-grained graph $\mathcal{CG}_\pi$ become the nodes of the line graph $\mathcal{V}_\pi$ and will be called *sites* in the following. Two different sites $v_1 = (c_{11}, c_{12})$ and $v_2 = (c_{21}, c_{22})$ are linked by $(v_1, v_2) \in \mathcal{N}_\pi$ if and only if $c_{11} = c_{21}$ and $(c_{12}, c_{22}) \notin \mathcal{V}_\pi$. The initial outlier model index is set to $\mu = 1$. Thus, the initial call of RDP-CRF is $\text{RDP-CRF}\big(\text{line-graph}\big(\mathcal{CG}_\pi\big), 1\big)$. After termination, $\mathfrak{RD}$ forms a preliminary segmentation into rigid domains. A postprocessing step (Subsection 2.7) produces our final prediction.

For every recursive call of RDP-CRF, the input site graph

$$\mathcal{G} = \big(\mathcal{V}, \mathcal{N}\big) \tag{12}$$

equals $\text{line-graph}\big(\mathcal{CG}\big)$ where

$$\mathcal{CG} = \big(\mathcal{C}, \mathcal{V}\big) \tag{13}$$

is the connected subgraph of the coarse-grained graph $\mathcal{CG}_\pi$, where $\text{line-graph}\big(\mathcal{CG}\big)$ is obtained from $\mathcal{CG}$ in the same way as $\text{line-graph}\big(\mathcal{CG}_\pi\big)$ is calculated from $\mathcal{CG}_\pi$. Consequently, the input site graph $\mathcal{G}$ is a subgraph of the initial site graph $\text{line-graph}\big(\mathcal{CG}_\pi\big)$ defined in Eq. (11). Using Eq. (6), we define site observations and edge observations for every site in $\mathcal{V}_\pi$ and every edge in $\mathcal{N}_\pi$ (see Eq. (5)):

- *Site observations:* For every site $v = (c_1, c_2) \in \mathcal{V}_\pi$, let

$$x_v := \text{mean-var}_D\big(c_1, c_2\big) \tag{14}$$

  be the observation at site v.
- *Edge observations:* For every edge $e = (v_1, v_2) \in \mathcal{N}_\pi$, where $v_1 = (c_1, c)$ and $v_2 = (c, c_2)$ and (c_1, c_2) is *not* an edge of the coarse-grained graph $\mathcal{CG}_\pi$, let

$$x_e := \text{mean-var}_D\big(c_1, c_2\big) \tag{15}$$

  be the observation at edge e.

Site and edge observations do not change during a recursive call of RDP-CRF.

Let $\mathcal{G} = \big(\mathcal{V}, \mathcal{N}\big) = \text{line-graph}\big(\mathcal{CG}\big)$ be any input site graph where $\mathcal{CG} = \big(\mathcal{C}, \mathcal{V}\big)$ is a connected subgraph of the coarse-grained graph $\mathcal{CG}_\pi$ (see Eqs. (12) and (13)), and let μ be any outlier model index (see Eqs. (8) and (9)). Let us discuss what happens when calling $\text{RDP-CRF}\big(\mathcal{G}, \mu\big)$.

Step 1: The feature-based exponential model associated with this call is computed as follows. Based on Eqs. (14) and (15) two sets of real-valued data points $D_{\mathcal{G}}^{\text{sites}}$ and $D_{\mathcal{G}}^{\text{edges}}$ are extracted: $D_{\mathcal{G}}^{\text{sites}} = \{x_v : v \in \mathcal{V}\}$ and $D_{\mathcal{G}}^{\text{edges}} = \{x_e : e \in \mathcal{N}\}$. We introduce for every edge $e \in \mathcal{N}$ and every node $v \in \mathcal{V}$ $\xi_e := \text{outlier}_{\mu, D_{\mathcal{G}}^{\text{edges}}}\big(x_e\big)$ and $\xi_v := \text{outlier}_{\mu, D_{\mathcal{G}}^{\text{sites}}}\big(x_v\big)$. We call ξ_v the *local label* of site v.

On the other hand, the *global label* of site v is its predicted label.

The following features determine the conditions under which equality and inequality of local labels ξ_v and global labels y_v ($v \in \mathcal{N}$) are rewarded or penalized. For every site $v \in \mathcal{V}$ and every label $y_v \in \{+1, -1\}$, the site feature $\Psi_\mu^{(1)}\left(y_v, \boldsymbol{x}\right)$ is defined by

$$\Psi_\mu^{(1)}\left(y_v, \boldsymbol{x}\right) := y_v \xi_v \,. \tag{16}$$

Equality of local and global labels is rewarded, differences are penalized. For every edge $e = \big(v_1, v_2\big) \in \mathcal{N}$, every $y_{v_1} \in \{+1, -1\}$ and every $y_{v_2} \in \{+1, -1\}$, the feature $\Psi_\mu^{(2)}\left(y_{v_1}, y_{v_2}, \boldsymbol{x}\right)$ is defined by Eqs. (17–19), where a distinction regarding the possible values of ξ_e, ξ_{v_1} and ξ_{v_2} is made.

Case "Two values among $\xi_e, \xi_{v_1}, \xi_{v_2}$ are equal to -1." Then equality of the local labels and the labels at sites v_1 and v_2 is rewarded, whereas inequality is penalized:

$$\Psi_\mu^{(2)}\left(y_{v_1}, y_{v_2}, \boldsymbol{x}\right) := \begin{cases} +1 & \text{if } y_{v_1}\xi_{v_1} + y_{v_2}\xi_{v_2} = 2; \\ -1 & \text{otherwise.} \end{cases} \tag{17}$$

Case "$\xi_{v_1} = \xi_{v_2} = +1$" seems to indicate that the three endpoints of v_1 and v_2 (which are in fact incident edges of the protein graph) belong to the same rigid component. But the node shared by the two edges may be part of a hinge region between two rigid components. This is likely to occur if $\xi_e = -1$, in which case we have to decide to which component the hinge node belongs. This decision is based on a comparison of x_{v_1} and x_{v_2} where no decision can be made, if $x_{v_1} = x_{v_2}$:

$$\Psi_\mu^{(2)}\left(y_{v_1}, y_{v_2}, \boldsymbol{x}\right) := \begin{cases} +1 & \text{if } y_{v_1} = -1, y_{v_2} = +1, \xi_e = -1 \text{ and } x_{v_1} > x_{v_2}; \\ +1 & \text{if } y_{v_1} = +1, y_{v_2} = -1, \xi_e = -1 \text{ and } x_{v_1} < x_{v_2}; \\ +1 & \text{if } y_{v_1} = y_{v_2} = +1 \text{ and } \xi_e = +1; \\ 0 & \text{if } y_{v_1} y_{v_2} = -1, \xi_e = -1 \text{ and } x_{v_1} = x_{v_2}; \\ -1 & \text{otherwise.} \end{cases} \tag{18}$$

For any other combination of ξ_{v_1}, ξ_{v_2} and ξ_e we set

$$\Psi_\mu^{(2)}\left(y_{v_1}, y_{v_2}, \boldsymbol{x}\right) \equiv 0. \tag{19}$$

Step 2: The Viterbi algorithm by Dong *et al.* (2014) is run on the feature-based exponential model that we computed in Step 1.

Step 3:

- If no site of $\mathcal{G}$ is labeled -1 by the Viterbi algorithm, and if $\mu < 11$, we call $\text{RDP-CRF}\big(\mathcal{G}, \mu + 1\big)$, and the control flow returns to the calling procedure;

- If no site of $\mathcal{G}$ is labeled -1 by the Viterbi algorithm, and if $\mu = 11$, we add $\bigcup_{c\in\mathcal{C}} c$ to $\mathfrak{RD}$, and the control flow returns to the calling procedure;
- All sites of $\mathcal{G}$ labeled -1 by the Viterbi algorithm are removed from the edge set of $\mathcal{CG} = (\mathcal{C}, \mathcal{V})$ resulting in $\mathcal{CG}' = (\mathcal{C}, \mathcal{V}')$;
- If $\mathcal{CG}'$ is still connected, we call $\text{RDP-CRF}\big(\text{line-graph}(\mathcal{CG}'), 1\big)$ and the control flow returns to the calling procedure.
- If $\mathcal{CG}' = (\mathcal{C}, \mathcal{V}')$ decomposes into d connected components $\mathcal{CG}'_\delta = (\mathcal{C}_\delta, \mathcal{V}'_\delta)$ where $\delta = 1, 2, \ldots, d$, then, for all $\delta = 1, 2, \ldots, d$,

– $\bigcup_{c\in\mathcal{C}_\delta} c$ is added to $\mathfrak{RD}$, if

$$\text{RMSD}\big(\bigcup_{c\in\mathcal{C}_\delta} c\big) < 3.5\text{Å}$$

– $\text{RDP-CRF}\big(\text{line-graph}(\mathcal{CG}'_\delta), 1\big)$ is called, if

$$\text{RMSD}\big(\bigcup_{c\in\mathcal{C}_\delta} c\big) \geq 3.5\text{Å}$$

Step 4: The control flow returns to the calling procedure.

After termination of the initial call $\text{RDP-CRF}\big(\text{line-graph}(\mathcal{CG}_\pi), 1\big)$, it may happen that list $\mathfrak{RD}$ is too fragmented to encode a reasonable segmentation. More precisely, some of rigid domains are partitioned by list elements. The reason is that the outlier detection $\text{outlier}_{\mu, D^{\text{sites}}_{\mathcal{G}}}$ used in Eqs. (17–19) is sometimes too permissive, in particular if $\mu > 1$ and $D^{\text{sites}}_{\mathcal{G}}$ is small. Therefore, we run the merging algorithm described in the next subsection as postprocessing step on list $\mathfrak{RD}$.

2.7 Finalizing the rigid domain segmentation

Let $\mathfrak{pRD}$ be the list of all unordered pairs $\{\mathcal{D}, \mathcal{D}'\}$ of different elements $\mathcal{D}$ and $\mathcal{D}'$ in $\mathfrak{RD}$. If $\mathfrak{RD}$ changes, $\mathfrak{pRD}$ changes accordingly.

Merging Algorithm

$$val \leftarrow \max_{\mathcal{D},\mathcal{D}'\in\mathfrak{pRD}} \frac{\text{RMSD}\big(\mathcal{D}\big)+\text{RMSD}\big(\mathcal{D}'\big)}{\text{RMSD}\big(\mathcal{D}\cup\mathcal{D}'\big)}$$

while $val > 1$ **do**

$$\{\mathcal{D}_*, \mathcal{D}'_*\} \leftarrow \operatorname*{argmax}_{\{\mathcal{D},\mathcal{D}'\}\in\mathfrak{pRD}} \frac{\text{RMSD}\big(\mathcal{D}\big)+\text{RMSD}\big(\mathcal{D}'\big)}{\text{RMSD}\big(\mathcal{D}\cup\mathcal{D}'\big)}$$

$$\mathcal{D}_* \leftarrow \mathcal{D}_* \cup \mathcal{D}'_*$$

Remove $\mathcal{D}'_*$ from $\mathfrak{RD}$

$$val \leftarrow \max_{\{\mathcal{D},\mathcal{D}'\}\in\mathfrak{pRD}} \frac{\text{RMSD}\big(\mathcal{D}\big)+\text{RMSD}\big(\mathcal{D}'\big)}{\text{RMSD}\big(\mathcal{D}\cup\mathcal{D}'\big)}$$

After termination of the Merging Algorithm $\mathfrak{RD}$ is returned as our rigid domain prediction.

3 Results

Our graph-based method aims to segment protein conformations into rigid domains without knowing the number of domains. To achieve this goal, the algorithm assigns a binary label to spatially close pairs of amino acids indicating if both amino acids belong to the same or different rigid domains. Our algorithm proceeds in multiple steps. First, a graph is built from protein conformations where nodes correspond to amino acids. Second, a community detection algorithm reduces the protein graph by merging amino acid vertices that are members of the same community. This step amplifies the signal and reduces the computational complexity of the subsequent steps. Third, the reduced graph is converted to a line graph whose vertices are the edges of the reduced graph. An edge is added to the line graph whenever the original pairs of edges share a vertex and the other two vertices are not linked in the original graph. Fourth, the general Viterbi algorithm computes the most probable binary labeling of the line graph based on a scoring function. By back tracing the line graph labeling we obtain a labeling of the reduced graph resulting in a segmentation of the protein conformations into rigid domains. In the final step, we check if the RMSD of each rigid domain exceeds a threshold, in which case we recursively run the algorithm on the subgraph corresponding to that domain. To validate our algorithm, we first segment conformations of Adelynate Kinase (ADK). We then perform a benchmark on 487 proteins from the DynDom database. Finally, we compare our method with other domain segmentation algorithms on a number of test cases ranging from medium to large scale conformational changes.

3.1 Rigid segmentation of Adenylate Kinase

We first run our algorithm for rigid domain segmentation on Adenylate Kinase (ADK) for which multiple experimental structures showing different conformations are available (Müller *et al.*, 1996). ADK catalyzes the interconversion of adenine nucleotides and is composed of three rigid domains. By closing the NMP-binding domain and the LID domain onto the CORE domain, ADK binds ATP and AMP which are converted to two ADP molecules. The PDB codes of ADK open and closed conformations are 4ake & 1ake respectively. ADK is composed of 214 amino acids which constitute the vertices of the initial protein graph.

To build the protein graph from those two states, we used $\delta = 7.5$ Å as cutoff.

Figure 3 illustrates the workflow of our algorithm and intermediate results for ADK using default values for the algorithm parameters. Panel 3A shows ADK's protein graph in which each vertex is an amino acid; the construction of edges linking spatially close amino acids is described in Methods. Amino acids are grouped by running the Louvain domain detection algorithm (Traag *et al.*, 2011) and merged into vertices of a coarse-grained graph. In the case of ADK, the protein graph comprising 214 vertices is transformed to a coarse-grained graph composed of 20 vertices (Figure 3B). In the next step, we construct the line graph of the coarse-grained graph (Figure 3C). We then run the general Viterbi algorithm (Dong *et al.*, 2014) on a scoring function defined on the line graph. This results in a binary labeling of the line graph (Figure 3D) or, equivalently, a labeling of the coarse-grained graph. Based on this labeling our method splits the coarse-grained graph into three disconnected subgraphs (Figure 3E). Finally, we map the unconnected subgraphs back to the protein graph to obtain a segmentation of ADK into three rigid domains (Figure 3F). Our segmentation agrees strongly with the domain boundaries defined in the literature (Whitford *et al.*, 2007), which we color-coded in Figure 3G for visual comparison. Our segmentation deviates from the literature annotation only in the hinge regions. This discrepancy is due to the ambiguous membership of amino acids in the hinge region which tend to be merged with amino acids from different domains in the coarse-graining step.

Unlike DynDom, our method could work with multiple conformational states. To study this feature, we ran our algorithm again but on 100 ADK conformations generated by morphing between the open and closed state (Habeck and Nguyen, 2018). The algorithm produces a similar results as before.

One of the advantage of our method is that it allows users to integrate a prior knowledge to improve the segmentation. For example, in the above default parameters setting, our method misclassified fifteen amino acids of NMP-binding and LID domain to the core domain. Yet with some prior knowledge about the rigid domains, we could integrate this information to enhance the estimation. For example, suppose we are given the ADK's segmentation calculated from Spectrus (Ponzoni *et al.*, 2015) with $K = 4$ (the number of rigid domains). We could integrate this prior knowledge (named the prior label) to our model as following. The weights of edges

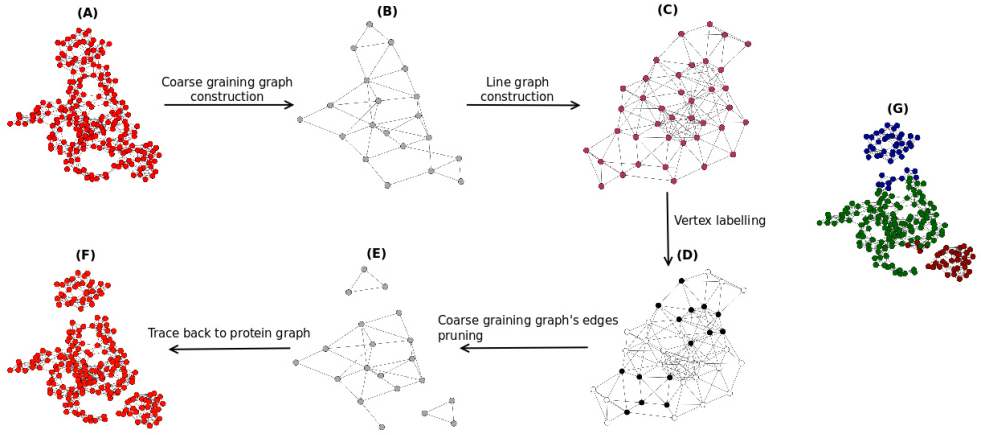

Fig. 3. Graph-based segmentation of ADK into rigid domains. (A) Protein graph constructed from 100 ADK conformations. (B) Reduced graph obtained by coarse graining the protein graph. (C) Line graph of the reduced graph. (D) Line graph with binary vertex labels (black: -1, white: +1) obtained with the general Viterbi algorithm. (E) The injective relation between edges of the reduced graph and vertices of the line graph allows us to also label the edges of the reduced graph. Edges having negative labels are removed resulting in three disconnected subgraphs. (F) Segmented protein graph derived from disconnected subgraphs in the reduced graph. (G) ADK graph with domain annotation from literature encoded by colors.

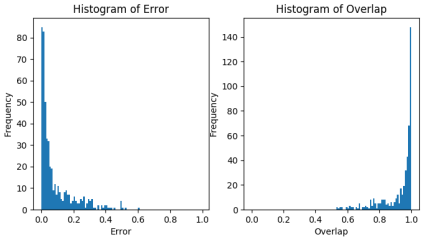

Fig. 4. Histogram of the error and the overlap evaluated on 487 proteins in the DynDom database.

in the protein graph whose vertices belong to different domains according to the prior label will be shrunk by the factor of $\alpha < 1$. In this study, we choose $\alpha = 0.75$. This setting helps the coarse-graining process to reduce the error of inconsistency (mentioned in the Discussion section) and thus improve the performance. For the evaluation, we ran our graph-based method on the new setting and figured out that there were only five amino acids misclassified from LID domain to the core domain. Thus even with an imperfect prior knowledge, this could significantly improve the result.

3.2 Rigid segmentation benchmark

We benchmarked our method on the DynDom database (Lee *et al.*, 2003) reduced to those pairs of proteins whose overall RMSD exceeds 5 Å. Moreover, we removed domains that span less than ten amino acids. To evaluate our method, we use the segmentation error and overlap defined by Nguyen and Habeck (2016). The overlap counts the number of matches between two segmentations after solving a low-dimensional linear assignment problem that maximizes the agreement between the two labelings. The error assesses how often two segmentations disagree on whether a pair of amino acids belongs to the same domain. Although both metrics differ in the details, they are highly anti-correlated.

Figure 4 shows histograms of the error and overlap between our and DynDom's segmentation evaluated on 487 proteins based on an edge cutoff value of 7.5 Å. The median error is 0.038 and the median overlap 0.972. The error and overlap histograms are highly skewed to small and large values, respectively. For approximately 30% of the examples, our method reaches a near perfect agreement with the annotation provided by DynDom (overlap ≥ 0.99). In only a few cases our method fails to produce a reasonable segmentation due to errors in the coarse-graining step and/or an indistinguishable signal derived from the mean variance.

Despite of the disagreements between our method and DynDom, our segmentation sometimes seems to be more reasonable. We investigate the open and closed state of human importin subunit beta-1 (PDB code 3lww, chain A & C) as an example. According to Dyndom, this protein has three rigid domains (Figure 5A) whose RMSDs are 6.843, 4.321, and 2.106(Å) respectively. We notice that the first domain of DynDom (darkgreen) is small, fragmented and has relative big RMSD. The second domain (darkred) has a significant portion intertwined with the third domain (darkblue). On the other hand, our segmentation suggests two separate domains whose RMSDs are 2.228 and 1.003(Å) (Figure 5B), which are much smaller than those in DynDom. In short, this example shows that the segmentation suggested by our model is more reasonable than one from DynDom.

To study the impact of the edge cutoff used in the definition of the protein graph, we ran experiments with varying cutoff values. Table 1 reports the mean and median of the overlap and error obtained with different edge cutoff values. The overlap seems to be largely unaffected by the specific choice of the cutoff, whereas the error drops slightly with larger cutoffs. Two possible explanations come to our mind. First, a larger cutoff results in protein graphs with more connections between amino acid vertices. Denser graphs seem to be more suitable to coarse graining with the Louvain method (see Figure S1 and the Discussion for a demonstration of this claim). Second, also the coarse-grained graph will be denser with larger cutoff values, which seems to improve the scoring of the line graph. However, because denser graphs result in larger line graphs, we need to restrict the cutoff to smaller values to tame the computational costs of the Viterbi algorithm.

3.3 Analysis of various structural transitions

We ran our method on various proteins studied in Nguyen and Habeck (2016) showing different types and scales of conformational changes. Table 2 provides the protein name, size and PDB code; Figure 6 shows

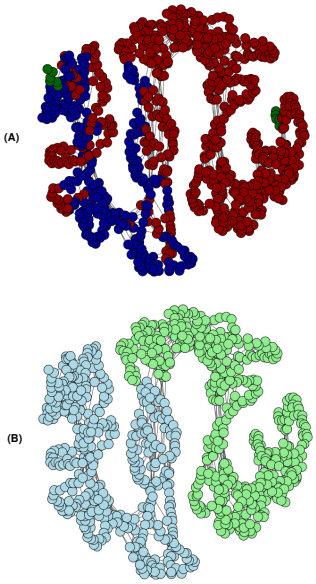

Fig. 5. Protein graph of human importin subunit beta-1 protein. (A) Segmentation suggested by DynDom: three rigid domains colored by dark green, red and blue. (B) Segmentation estimated by our method: two rigid domains colored by light green and blue.

Table 1. Performance of the graph-based algorithm for different edge cutoffs evaluated on the DynDom benchmark.

Metric / Cutoff	Median overlap	Mean overlap	Median error	Mean error
7.5 Å	0.972	0.924	0.038	0.086
10.5 Å	0.977	0.924	0.034	0.083
13.5 Å	0.972	0.926	0.033	0.081

a summary of the segmentation analysis. First, we study and compare the performance of our algorithm (graph-based method) to other methods by analyzing protein complexes that undergo large-scale conformational changes.

Pyruvate phosphate dikinase (PPDK) is a large biomolecular complex that catalyzes the reversible conversion of PEP, AMP, and $\mathrm{P_i}$ to pyruvate and ATP (Lim *et al.*, 2007). We apply our graph-based method to two PPDK structures and compare the segmentation to the annotation found in the literature (Lim *et al.*, 2007) and by other methods such as Spectrus, DynDom as well as Nguyen and Habeck (2016) . Our segmentation agrees strongly with the segmentation provided by DynDom, but fails to detect the additional domain reported in the literature and by Nguyen and Habeck (2016). Typically, our method produces a fewer number of domains than reported in the literature, because we only take changes in a few structural snapshots into account and no additional experimental information. For $K = 3$, Spectrus agrees strongly with the segmentation found by our graph-based approach except for the first domain, which is significantly larger according to Spectrus.

T7 RNA polymerase is involved in the initiation and elongation of RNA transcription. Our segmentation is highly consistent with the results from DynDon, Nguyen and Habeck (2016) and the anotation from the literature (Theis *et al.*, 2004). Spectrus fails to identify the refolding loop inserted in the N-terminal domain.

The chaperonin GroEL (Boisvert *et al.*, 1996) provides a shielded environment to assist protein folding and prevent aggregation. For this example, all methods provide very similar segmentation results.

We also benchmark our method on proteins undergoing medium-scale structural transitions. Aspartate aminotransferase (AST) is an enzyme involved in amino acid metabolism that catalyzes the reversible transfer of an α-amino group between aspartate and glutamate (Karmen *et al.*, 1955). For this example, we find a high agreement between our method and other segmentations. Another example is the enzyme Alcohol dehydrogenase (AhD) that decomposes alcohol into aldehyde. Our graph-based segmentation agrees strongly with the result from DynDom. Spectrus achieves its maximum score for $K = 3$ domains, but introduces an additional domain compared to the other methods. For $K = 2$, the score is lower, but Spectrus' segmentation is more consistent to DynDom and our result.

4 Discussion

Our results demonstrate that segmentation of protein conformations into rigid domains can be achieved with a graph-based algorithm that solves the rigid segmentation problem with an edge-labeling strategy. Let us discuss the key features of the algorithm and the impact of algorithmic parameters. To measure the efficiency of the graph construction and coarse graining, we use a metric that we call inconsistency error. The inconsistency error

Table 2. Proteins in different scale conformational changes involved in the assessment

Protein	PDB code	chain ID	size
PPDK	1kc7	A	872
	2r82	A	
T7 RNA polymerase	1qln	A	842
	1msw	D	
GroEL	1aon	A	524
	1aon	H	
Aspatate aminotransferase	9aat	A	401
	1ama	A	
Alcohol dehydrogenase	1adg	A	374
	2ohx	A	

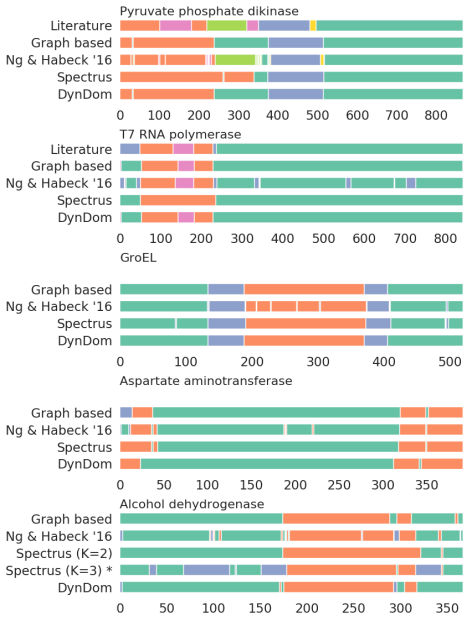

Fig. 6. Analysis of several proteins undergoing conformational changes on a variety of scales. Large-scale conformational changes: pyruvate phosphate dikinase, T7 RNA polymerase, GroEL. Medium-scale conformational changes: Aspartate aminotransferase, Alcohol dehydrogenase. For each protein, the segmentation found by different methods and in the literature are shown. Same color means same domain.

quantifies the heterogeneity of clusters weighted by their size. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph composed of $N = |\mathcal{V}|$ vertices $v_i \in \mathcal{V}$ with labels σ_i and $\mathcal{C} = \{\mathcal{C}_k\}$ a partition of the vertices into clusters $\mathcal{C}_k \subset \mathcal{V}$ obtained by coarse graining. We define the inconsistency error of the coarse graining procedure as $\mathrm{error}(\mathcal{C}|\mathcal{G}) = 2\sum_{\mathcal{C}_k \in C} \frac{|\mathcal{C}_k|}{N} \frac{\sum_{i<j \in \mathcal{C}_k} |\sigma_i \neq \sigma_j|}{|\mathcal{C}_k|\,(|\mathcal{C}_k|-1)}$ which is the average number of labeling mismatches within each cluster weighted by cluster size.

We first study different ways to construct a protein graph from multiple conformations. There are many reasonable options for constructing a protein graph. For example, one possibility is to create an edge if the distance between two vertices is smaller than a cutoff in at least one conformation, and to assign as a weight the number of such conformations. Another possibility (detailed in Methods) is to create an edge if its distance is smaller than the cutoff in *all* conformations, and to weight the edge by the reciprocal exponentiated variance computed over all conformations (such that low-variance edges have a weight close to one and large-variance edges are assigned small weights). Supplementary Figure S1 demonstrates that the second graph construction rule consistently outperforms the first rule based on the inconsistency error. We therefore used the second rule in our benchmark calculations. In addition, we tested different values of the edge cutoff distance and noticed a minor, but not significant improvement of the inconsistency error for larger cutoff values.

We also studied various options for the coarse-graining step. In all tests, we used the Louvain algorithm for fitting Potts models (Traag *et al.*, 2011) for coarse graining. The resolution parameter was adjusted so as to produce about 20 clusters of medium size. Too large clusters risk to

merge amino acids from hinge regions and thus the inconsistency error is expected to increase. Too small clusters will tend to show a smaller inconsistency error at the cost of lowering the significance of the mean variance between two clusters. Large graphs will pose a computational challenge in the Viterbi step, because the number of vertices of the line graph grows quadratically with the number of vertices in the original graph. By using our coarse-graining strategy, we save computational resources and enhance the signal as shown in Supplementary Information (see second section and Supplementary Figure S2).

Moreover, we run our algorithm on Lysozyme (BLAKE *et al.*, 1965), an enzyme contributing to the innate immune system, to investigate if this graph-based algorithm could produce a reasonable segmentation given several actual conformations. In this study, we use 100 conformations of Lysozyme whose PDB codes can be found at the Supplementary. Our algorithm suggests two domains whose $RMSD$ are 1.638 and 4.873 (Å) respectively (Supplementary Figure S3).

To avoid duplication of features involving vertices and edges, we modify the construction of the line graph by discarding an edge if its two end vertices are connected as well. That way, features extracted from edges add new information. Finally, we use a merging routine with heuristic criteria to merge two domains. One may ask if we could skip the labeling step (Viterbi algorithm) and apply the merging routine directly to the clusters found by coarse graining. This simplified version of our algorithm achieves good results on proteins showing a large-scale movement, but fails on more subtle cases. Overall, post-processing via the merging procedure compensates for segmentation errors involving small fragments.

The running time of our algorithm depends on the size of the protein, the density of the protein graph, and the rigidity of the conformational change. Supplementary Figure S4 shows the relationship between protein size and the running time of our graph-based segmentation algorithm. We note that the running time for proteins smaller than 800 amino acids grows slowly in a linear fashion. For the larger proteins, it seems to grow quadratically. There are a few outlier proteins whose running time is significantly longer than for proteins of similar size. In these problematic cases, the signal derived from the mean-variance metric fails to distinguish the labels of vertices and edges in the line graph.

5 Conclusion

We present a new algorithm to characterize structural transitions in proteins. Our graph-based algorithm constructs a graph from a set of protein conformations and detects rigid domains via an edge labeling strategy. A key feature is that the number of rigid domains is determined automatically.

Yet the algorithm allows users to relax the rigid definition of domains, thus resulting to the increase or decrease number of rigid domains. Segmentations produced by our algorithm agree strongly with segmentations found by other methods such as DynDom (Hayward and Berendsen, 1998; Hayward *et al.*, 1997) and Spectrus (Ponzoni *et al.*, 2015) on various medium to large scale structural transitions.

Our approach has several advantages over other rigid segmentation methods. First, there is no limitation on the number of protein conformations. In fact, a larger number of conformations should result in a better signal and thereby a superior performance of the algorithm. Second, by using the graph-based model along with a binary labeling of edges, we overcome the need to choose the number of rigid domains, which is necessary for many of the existing methods. Moreover, our method performs well with default parameter settings, which saves the user from parameter tweaking. Finally, our graph-based framework is quite flexible by allowing us to integrate into the scoring function additional information such as the location of hinges.

Acknowledgements

Funding

Michael Habeck and Thach Nguyen haven been supported by Deutsche Forschungsgemeinschaft (DFG), SFB 860 TP B09 .

References

Abyzov, A., Bjornson, R., Felipe, M., and Gerstein, M. (2010). Rigidfinder: a fast and sensitive method to detect rigid blocks in large macromolecular complexes. *PROTEINS: Structure, Function, and Bioinformatics*, **78**(2), 309–324.

BLAKE, C. C. F., KOENIG, D. F., MAIR, G. A., NORTH, A. C. T., PHILLIPS, D. C., and SARMA, V. R. (1965). Structure of hen egg-white lysozyme: A three-dimensional fourier synthesis at 2 å resolution. *Nature*, **206**(4986), 757–761.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, **2008**(10), P10008.

Boisvert, D. C., Wang, J., Otwinowski, Z., Norwich, A. L., and Sigler, P. B. (1996). The 2.4 å crystal structure of the bacterial chaperonin groel complexed with atpgs. *Nature Structural Biology*, **3**(2), 170–177.

Dong, Z., Wang, K., Dang, T. K. L., Gültas, M., Welter, M., Wierschin, T., Stanke, M., and Waack, S. (2014). Crf-based models of protein surfaces improve protein-protein interaction site predictions. *BMC Bioinformatics*, **15**, 277.

Emekli, U., Schneidman-Duhovny, D., Wolfson, H. J., Nussinov, R., and Haliloglu, T. (2008). Hingeprot: automated prediction of hinges in protein structures. *Proteins: Structure, Function, and Bioinformatics*, **70**(4), 1219–1227.

Evans, T. S. and Lambiotte, R. (2010). Line graphs of weighted networks for overlapping communities. *The European Physical Journal B*, **77**(2), 265–272.

Flores, S. C. and Gerstein, M. B. (2007). Flexoracle: predicting flexible hinges by identification of stable domains. *BMC bioinformatics*, **8**(1), 215.

Gerstein, M., Lesk, A. M., and Chothia, C. (1994). Structural mechanisms for domain movements in proteins. *Biochemistry*, **33**(22), 6739–6749.

Habeck, M. and Nguyen, T. (2018). A probabilistic network model for structural transitions in biomolecules. *Proteins*, **86**, 634–643.

Hayward, S. and Berendsen, H. J. (1998). Systematic analysis of domain motions in proteins from conformational change: New results on citrate synthase and t 4 lysozyme. *Proteins Structure Function and Genetics*, **30**(2), 144–154.

Hayward, S., Kitao, A., and Berendsen, H. J. (1997). Model-free methods of analyzing domain motions in proteins from simulation: A comparison of normal mode analysis and molecular dynamics simulation of lysozyme. *Proteins: Structure, Function, and Bioinformatics*, **27**(3), 425–437.

Henzler-Wildman, K. and Kern, D. (2007). Dynamic personalities of proteins. *Nature*, **450**, 964–972.

Hinsen, K. (1998). Analysis of domain motions by approximate normal mode calculations. *Proteins: Structure, Function, and Bioinformatics*, **33**(3), 417–429.

Hirsch, M. and Habeck, M. (2008). Mixture models for protein structure ensembles. *Bioinformatics*, **24**(19), 2184–2192.

Iglewicz, B. and Hoaglin, D. C. (1993). *How to detect and handle outliers*, volume 16. Asq Press.

Kabsch, W. (1976). A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, **32**, 922–923.

Karmen, A., Wroblewski, F., and Ladue, J. S. (1955). Transaminase activity in human blood. *The Journal of clinical investigation*, **34**(1), 126–131. 13221663[pmid].

Lee, R. A., Razaz, M., and Hayward, S. (2003). The DynDom database of protein domain motions. *Bioinformatics*, **19**(10), 1290–1291.

Lim, K., Read, R. J., Chen, C. C. H., Tempczyk, A., Wei, M., Ye, D., Wu, C., Dunaway-Mariano, D., and Herzberg, O. (2007). Swiveling domain mechanism in pyruvate phosphate dikinase,. *Biochemistry*, **46**(51), 14845–14853. PMID: 18052212.

Müller, C. W., Schlauderer, G. J., Reinstein, J., and Schulz, G. E. (1996). Adenylate kinase motions during catalysis: an energetic counterweight balancing substrate binding. *Structure*, **4**, 147–156.

Nguyen, T. and Habeck, M. (2016). A probabilistic model for detecting rigid domains in protein structures. *Bioinformatics*, **32**(17), i710–i717.

Ponzoni, L., Polles, G., Carnevale, V., and Micheletti, C. (2015). Spectrus: A dimensionality reduction approach for identifying dynamical domains in protein complexes from limited structural datasets. *Structure*, **23**(8), 1516–1525.

Theis, K., Gong, P., and Martin, C. T. (2004). Topological and conformational analysis of the initiation and elongation complex of t7 rna polymerase suggests a new twist. *Biochemistry*, **43**(40), 12709–12715. PMID: 15461442.

Traag, V. A. (2015). Faster unfolding of communities: Speeding up the Louvain algorithm. *Physical Review E*, **92**(3), +032801.

Traag, V. A., Van Dooren, P., and Nesterov, Y. (2011). Narrow scope for resolution-limit-free community detection. *Physical Review E*, **84**(1), +016114.

Whitford, P. C., Miyashita, O., Levy, Y., and Onuchic, J. N. (2007). Conformational transitions of adenylate kinase: switching by cracking. *J. Mol. Biol.*, **366**, 1661–1671.

Wriggers, W. and Schulten, K. (1997). Protein domain movements: detection of rigid domains and visualization of hinges in comparisons of atomic coordinates. *Proteins Structure Function and Genetics*, **29**(1), 1–14.

4.1 Support Information

1 Inconsistency error of reduced graph

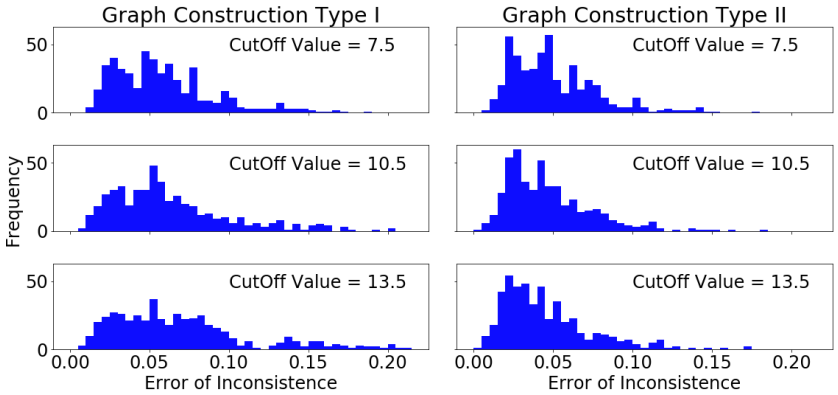

Figure S1: Histogram of inconsistency's error from graph construction type I and II and their various cutoff values respectively.

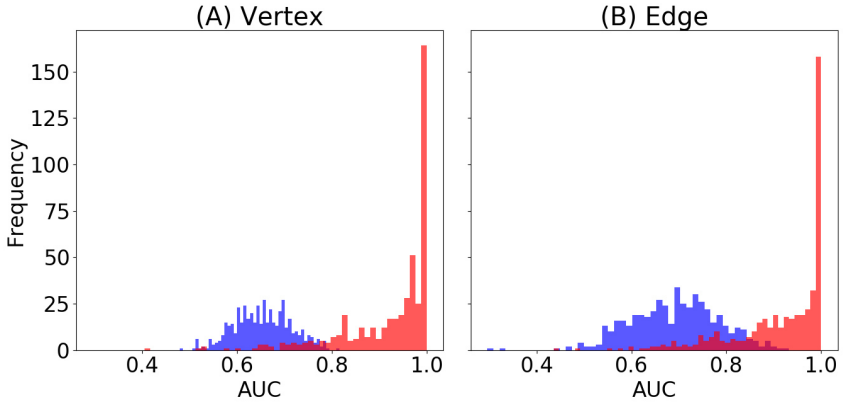

Figure S2: AUC based on the mean-variance of vertices and edges. (A) The blue histogram shows the AUC obtained with the vertex features separating positive from negative vertex labels in the protein graph. The red histogram indicates the performance of the vertex features derived from the coarse-grained graph. (B) AUC for separating positive and negative edge labels based on the edge features derived from the protein graph (blue histogram) and the coarse-grained graph (red histogram).

2 Signal enhancement by coarse graining

As in Methods and Discussion, let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a graph and $\mathcal{LG} = (\mathcal{LV}, \mathcal{LE})$ the line graph derived from $\mathcal{G}$. For each vertex $v^* \equiv (v_0, v_1) \in \mathcal{LV}$ $(v_0, v_1 \in \mathcal{V})$, we evaluate the mean-variance $\text{mean-var}\big(v^*\big) \equiv \text{mean-var}\big(v_0, v_1\big)$ using Eq. (10) in Methods. The label of v^* is $\sigma_{v^*} = +1$ if $\sigma_{v_0} = \sigma_{v_1}$ and -1 otherwise. For each edge $e^* \equiv (v_l, v_m, v_r) \in \mathcal{LE}$ $(v_l, v_m, v_r \in \mathcal{V}; (v_l, v_m), (v_m, v_r) \in \mathcal{E}, (v_l, v_r) \notin \mathcal{E})$, its mean-variance $\text{mean-var}\big(v^*\big) \equiv \text{mean-var}\big(v_l, v_r\big)$ is calculated as above. The label of an edge in the line graph is $\sigma_{e^*} = +1$ if $\sigma_{v_l} = \sigma_{v_r}$, and -1 otherwise.

The performance of the CRF scoring function depends on how well it can separate positive and negative vertices and edges in the line graph. We try to classify vertex and edge labels by thresholding the mean-variance. We assess the classification power of the mean-variance features by using the area under the ROC curve (AUC) before and after coarse graining the protein graph. We evaluate and compare the performance of the mean-variance features for both vertices and edges of the line graph derived from the protein graph and the coarse-grained graph.

Figures S2.A and S2.B indicate that the mean-variance feature of both vertices and edges produces significantly better classification results for the coarse-grained graph compared to the protein graph. Therefore, the coarse-graining step increases the information represented by the mean-variance feature.

3 Lysozyme protein

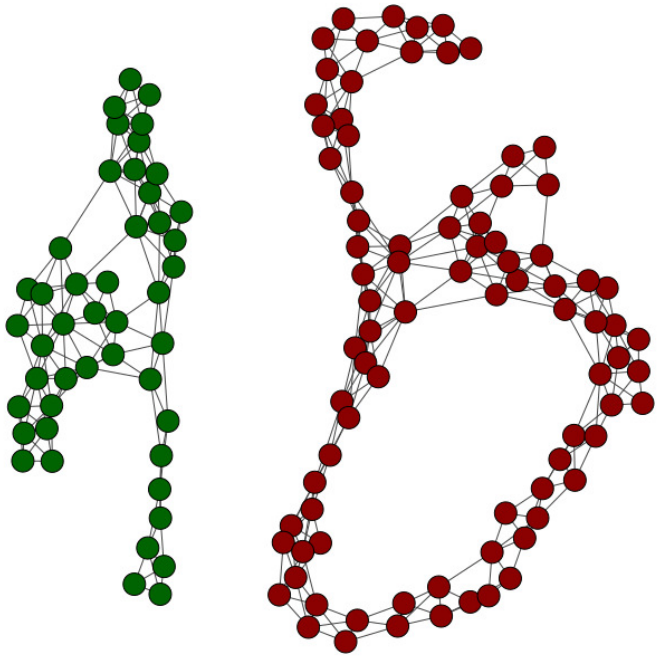

Figure S3: Segmentation of lysozyme protein by our graph-based method. The $RMSD$s of two resulting domains are 1.638 and 4.873 (Å) respectively.

3.1 PDB codes used in the study

2X0G_B 3G43_B 3G43_C 3G43_A 3G43_D 1K90_F 2K61_A 5WSV_A 5WSV_C 5DBR_A 2BE6_A 1S26_F 1S26_E 1S26_D 4OVN_E 1SK6_E 1Y0V_M 1Y0V_L 1Y0V_K 1Y0V_J 1Y0V_I 1Y0V_H 2F3Z_A 1K93_D 3OXQ_A 3OXQ_B 3OXQ_C 3OXQ_D 3BXL_A 2F3Y_A 1XFZ_P 1XFZ_Q 1XFZ_R 1XFZ_S 1XFZ_T 1XFY_P 1XFU_R 1K93_F 1K93_E 1XFZ_O 2DFS_D 2DFS_F 2DFS_B 2DFS_N 2DFS_P 2DFS_R 1XFW_O 1XFU_S 1LVC_D 1XFU_Q 1LVC_F 1XFU_T 1XFU_O 1XFW_T 1XFW_Q 1XFW_P 1XFW_S 1XFW_R 3EWV_A 1XFU_P 2VAY_A 1XFX_O 1XFX_T 2BE6_B 1XFX_S 1XFX_P 1XFX_Q 1XFX_R 4R8G_H 1XFV_O 1OOJ_A 1XFV_T 1LVC_E 1AHR_A 1XFV_P 1XFV_Q 1XFV_R 1XFV_S 1QX7_M 1QX7_B 1QX7_R 4DCK_B 1QX5_D 1XFY_T 1XFY_S 1XFY_R 1XFY_Q 1MUX_A 1XFY_O 1QX5_R 1QX5_T 1QX5_Y 5HIT_A 1QX5_B 1PK0_E 1PK0_D 1PK0_F 1QX5_I 1QX5_J 1QX5_K

4 Running time

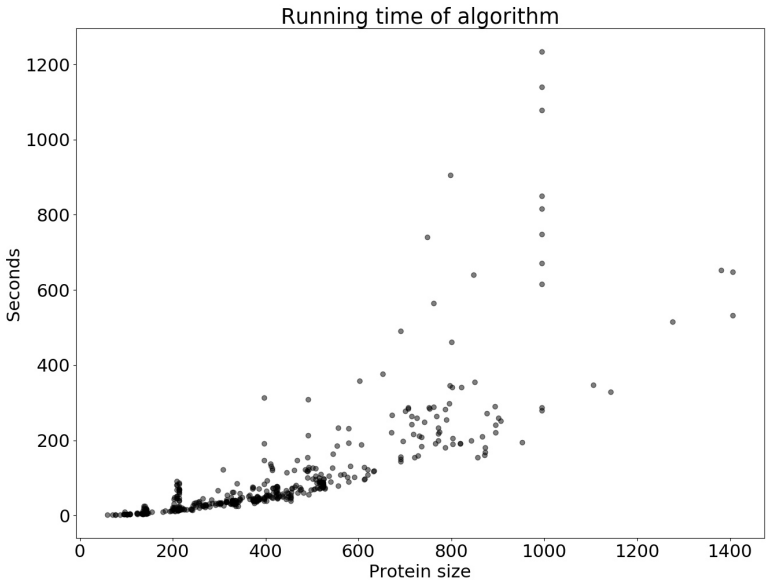

Figure S4: Protein size versus running time (measured in seconds) evaluated for 487 proteins selected from the DynDom database.

Chapter 5

A probabilistic network model for structural transitions in biomolecules

This chapter introduces the probabilistic network model for structural transitions in proteins. Here we present a probabilistic model for structural transitions in protein conformational states. This chapter was published in Protein journal 2018. Cited as: Habeck and Nguyen 2018.

Own contribution:

- Concept and implementation part of the algorithm and the code.
- Construct the Dyndom dataset, the validation data from (Sfriso et al., 2013) for test.
- Figures 3, 4.
- Manuscript in parts and the support information.

Received: 5 September 2017 | Revised: 12 February 2018 | Accepted: 5 March 2018

DOI: 10.1002/prot.25490

RESEARCH ARTICLE

WILEY PROTEINS

A probabilistic network model for structural transitions in biomolecules

Michael Habeck[1,2] | Thach Nguyen[2]

[1]Statistical Inverse Problems in Biophysics, Max Planck Institute for Biophysical Chemistry, Am Fassberg 11, 37077 Göttingen, Germany

[2]Felix Bernstein Institute for Mathematical Statistics in the Biosciences, Georg August University Göttingen, Goldschmidtstrasse 7, 37077 Göttingen, Germany

Correspondence
Michael Habeck, Max Planck Institute for biophysical Chemistry, Am Fassberg 11, 37077 Göttingen, Germany.
Email: mhabeck@gwdg.de

Funding information
Deutsche Forschungsgemeinschaft, Grant/Award Number: SFB860/B09

Abstract

Biological macromolecules often undergo large conformational rearrangements during a functional cycle. To simulate these structural transitions with full atomic detail typically demands extensive computational resources. Moreover, it is unclear how to incorporate, in a principled way, additional experimental information that could guide the structural transition. This article develops a probabilistic model for conformational transitions in biomolecules. The model can be viewed as a network of anharmonic springs that break, if the experimental data support the rupture of bonds. Hamiltonian Monte Carlo in internal coordinates is used to infer structural transitions from experimental data, thereby sampling large conformational transitions without distorting the structure. The model is benchmarked on a large set of conformational transitions. Moreover, we demonstrate the use of the probabilistic network model for integrative modeling of macromolecular complexes based on data from crosslinking followed by mass spectrometry.

KEYWORDS

Bayesian statistics, conformational change, crosslinking/mass spectrometry, Markov chain Monte Carlo, network model, protein structure, structural modeling

1 | INTRODUCTION

Biological function is often linked to conformational changes in the interacting molecules.[1] These structural transitions occur on many different time and length scales, ranging from fast reorientations of side chains to large conformational transitions involving the collective movement of entire domains.[2]

It is challenging to study functional transitions in biomolecules experimentally. X-ray crystallography, the major method to elucidate the structure of biomolecules,[3] provides a rather static picture of a biomolecule in an artificial crystal environment. NMR has the potential to reveal not only the structure but also dynamics, but faces problems when looking at large systems.

Recently, cryo-electron microscopy (cryo-EM) has emerged as a powerful high-resolution method for biomolecular structure determination (for example, Refs.[4,5]). In addition to measuring structural data at atomic or near-atomic resolution, cryo-EM can also provide information about structural dynamics in the form of multistate reconstructions.[6–8] Other methods such as FRET[9] can characterize the native state ensemble explored by large biomolecules.

Although all of these experimental methods can reveal information both on the structure and the dynamics of a biomolecule, we often need to complement experimental data with simulations. Molecular dynamics (MD) can reliably probe and predict conformational dynamics in biomolecules,[10,11] but becomes very costly when studying large-scale rearrangements implicated in cellular processes.

Standard applications of MD simulate the dynamics *ab initio* without any additional data. Alternatively, experimental data can be combined with a computational model to study a conformational transition. There is a long tradition of coarse-grained network models that have been used as a computationally feasible alternative to atomistic molecular dynamics simulations.[12] The Gō model,[13] for example, was among the first models to study protein folding.

The Gaussian network model (GNM) has been proposed to predict the internal dynamics of biological macromolecules[14,15] from a single structural snapshot. It was shown that GNMs can reproduce the local dynamics of proteins as measured by crystallographic B-factors[16] and NMR order parameters.[17] A further improvement was achieved by the anisotropic network model (ANM).[18]

Although elastic network models (ENMs) such as GNM and ANM can to some extent predict the local dynamics, they are less suited to describe large-scale conformational changes in proteins.[12] The reason for this limitation is that ENMs can only capture harmonic motions about a single ground-state but large-scale conformational changes are

anharmonic and can often only be triggered by cracking the bonds of the elastic network.[19] Recently, Kurkcuoglu et al.[20] have proposed an iterative clustering and ENM construction procedure that aims to circumvent the problem and allows for sampling of larger scale structural transitions.

The "crack resistance" of ENMs is less problematic, if experimental data about the alternative conformational state can guide the simulation and eventually trigger the rupture of network bonds. For example, ENMs have been combined with experimental data from cryo-EM or SAXS to sample conformational changes in proteins.[21] But closed-to-open transitions still pose a challenge.[22] Another powerful extension is the deformable elastic network (DEN) by Schröder et al.[23] which was developed mainly to fit biomolecules flexibly into cryo-EM maps. DEN adapts the contact distance in the course of flexible fitting.

Network models have also been used to predict transition paths between two conformational states. The plastic network model proposed by Maragakis and Karplus[24] can find transition paths between two conformational states and predict intermediate structures. A limitation of the plastic network model is that it requires an explicit representation of the target structure. But often, the target state is only characterized implicitly and incompletely by structural data.

All of the above methods have their strength and weaknesses. The most important limitation is that as the models become more complex, we often have to set a number of algorithmic parameters. Therefore, it is desirable to develop a more principled approach that estimates these parameters from the experimental data and known structures.

In this article, we develop a probabilistic network model derived from statistics of protein structures and conformational transitions. The model can be used to find minimally invasive conformational changes that allow for the fitting of experimental data that show an alternative structural state. Our goal is mainly to predict the final structure rather than generating a meaningful transition path. The model is based on a statistical analysis of a large set of conformational changes as well as experimental and theoretical ensembles. Rather than using harmonic springs the analysis suggests the use of anharmonic springs derived from heavy-tailed distance distributions that can be fitted with a generalized lognormal distribution. In addition to using anharmonic springs, the model allows for the rupture of contacts during the conformational transition. The rupture of springs can be simulated efficiently and adaptively as the conformational switch progresses. We demonstrate the efficiency and accuracy of our model on a benchmark of conformational transitions.[25] Finally, we combine our network model with crosslinking restraints to predict alternative conformational states from sparse data observed with crosslinking followed by mass spectrometry.

2 | MATERIALS AND METHODS

2.1 | The generalized lognormal distribution

To construct a probabilistic model for conformational changes in proteins, let us first study the distribution of interatomic distances in protein structures that do not undergo a substantial structural transition. The ANM imposes a harmonic potential between Cα atoms that are in contact.[18] Therefore, the ANM assumes that differences in Cα distances follow a Gaussian distribution. However, because distances are intrinsically positive quantities, it is more adequate to consider differences in the logarithms of the distances (or equivalently the logarithm of the ratio of both distances) rather than differences in the distances themselves.[26] If the log distance ratios are assumed to follow a Gaussian distribution, the model for distance fluctuations is the lognormal distribution.

Here, we will consider a more general family of distance distributions that includes the lognormal distribution as a special case. The generalized lognormal distribution is defined as

$$\mathrm{GLN}(r;\rho,\lambda,\beta)=\frac{\beta}{2\Gamma(1/\beta)\,\lambda\, r}\exp\left\{-|\ln(r/\rho)|^{\beta}/\lambda^{\beta}\right\} \tag{1}$$

where $r \geq 0$ is a non-negative variable such as a distance or a ratio of two distances and Γ is the Gamma function. GLN has three positive parameters ρ, λ, and β. The logarithm of ρ is the mode, mean and median of the distribution of $\ln r$. The shape parameter β controls the fatness of the tails of the distribution. For $\beta \leq 1$, GLN shows increasingly heavy tails. Important special cases are the lognormal distribution ($\beta = 2$) and the log-Laplace distribution ($\beta = 1$). Due to the heavier tails for $\beta < 2$, the generalized lognormal distribution can accommodate occasional outliers and thereby allows for more robust modeling in comparison with the lognormal distribution. The variance of $\ln r$ depends on the scale parameter λ and the shape of the distribution: $\mathrm{var}(\ln r)=\frac{\Gamma(3/\beta)}{\Gamma(1/\beta)}\lambda^2$.

The parameters ρ, λ, and β of the generalized lognormal distribution (Equation 1) can be inferred from n observations $\{r_i\}=\{r_1,\ldots,r_n\}$ in a straightforward way. Assuming a Jeffreys' prior (that is, a uniform prior distribution in log space) for the strictly positive parameters ρ, λ, and β, the full posterior probability is:

$$\Pr(\rho,\lambda,\beta\,|\,\{r_i\}) \propto \frac{\beta^{n-1}}{\rho\,\lambda^{n+1}\,\Gamma(1/\beta)^n}\exp\left\{-\sum_{i=1}^{n}|\ln r_i-\ln\rho|^{\beta}/\lambda^{\beta}\right\}. \tag{2}$$

Because $\ln\rho$ is the median of $\ln r$, $\hat{\rho}=\exp\{\mathrm{median}(\ln r_1,\ldots,\ln r_n)\}$ is a stable estimator for ρ. The sample variance of $\ln r_i$ is an estimator for $\frac{\Gamma(3/\beta)}{\Gamma(1/\beta)}\lambda^2$. Therefore, assuming that we know the shape parameter β, we can estimate the scale by $\hat{\lambda}^2=\frac{\Gamma(1/\beta)}{\Gamma(3/\beta)}\,\mathrm{var}\{\ln r_i\}$. If we plug these two estimators into the posterior distribution (Equation 2), we obtain a probability that depends only on β. This distribution can be maximized numerically to obtain an estimate of the shape parameter $\hat{\beta}$ from which an estimate of the scale parameter $\hat{\lambda}=\sqrt{\mathrm{var}\{\ln r_i\}\,\frac{\Gamma(1/\hat{\beta})}{\Gamma(3/\hat{\beta})}}$ is obtained.

2.2 | Two-component mixture of generalized lognormal distributions

Distance changes observed in biological macromolecules that undergo structural transitions can be roughly divided into two classes: small-amplitude fluctuations due to thermal motions and larger scale transitions that are often involved in the biological function. The distance distribution can therefore be described approximately by a two-component mixture model. One component accounts for the

small-amplitude fluctuations, whereas the other component models large-scale variations in the distances due to the conformational change.

Let us denote the proportion of contacts that are preserved during the conformational change by $w \in [0, 1]$. The two-component mixture model for a distance r is

$$\begin{aligned}\Pr(r|\lambda_1, \lambda_0, \rho_1, \rho_0, \beta_1, \beta_0, w) = w\,\mathrm{GLN}(r; \rho_1, \lambda_1, \beta_1) \\ +(1-w)\,\mathrm{GLN}(r; \rho_0, \lambda_0, \beta_0)\end{aligned} \quad (3)$$

where λ_1 and λ_0 measure the amplitudes of the distance fluctuations. The first component, $\mathrm{GLN}(r; \rho_1, \lambda_1, \beta_1)$, models the distance fluctuations in preserved contacts, whereas the second component describes large-amplitude fluctuations due to the conformational change (that is, $\lambda_1 \ll \lambda_0$).

To learn the mixture model (Equation 3) from observed distances, we use a Gibbs sampler[27] which is commonly used to fit mixture models.[28] The Gibbs sampler uses the identity $p_1 + p_2 = \sum_{z\in\{0,1\}} p_1^z\, p_2^{1-z}$ which is valid for every pair of positive numbers p_1, p_2. Applied to the generalized lognormal mixture (Equation 3) we have

$$\Pr(r|\alpha_0, \alpha_1, w) = \sum_{z\in\{0,1\}} w^z (1-w)^{1-z} \left[\mathrm{GLN}(r|\alpha_1)\right]^z \left[\mathrm{GLN}(r|\alpha_0)\right]^{1-z} \quad (4)$$

where the parameters of the GLN have been collected into $\alpha_i = (\rho_i, \lambda_i, \beta_i)$, $i = 0, 1$. This means that we can simulate the two-component mixture by stochastically switching on ($z = 1$) and off ($z = 0$) network bonds. In a Gibbs sampling approach, the sum in Equation 4 is not calculated analytically but evaluated stochastically by sampling a binary indicator $z_{ij} \in \{0, 1\}$ for every distance r_{ij} that defines a bond between residues i and j.

2.3 | The DynDom database

The DynDom database[29] has been created on the basis of the method by Hayward and Berendsen,[30] which decomposes protein structures into rigid domains. DynDom automatically detects rigid domains by comparing two conformational states of a protein. We analyzed 450 examples from the DynDom database which show a conformational change larger than 5 Å and involve rigid domains larger than 100 amino acids.

2.4 | Inferential structure determination

Inferential structure determination (ISD)[31,32] is a fully probabilistic method for biomolecular structure determination from experimental data D. ISD implements a Bayesian approach that solves a structure determination problem by sampling conformations θ from its posterior probability distribution $\Pr(\theta|D, I)$ (I is the *background information* that encodes all modeling assumptions, θ are the conformational degrees of freedom). According to Bayes's theorem we can decompose the posterior distribution into a product of the likelihood function $L(\theta) = \Pr(D|\theta, I)$, which is the probability of the data given a candidate structure parameterized by θ, and the prior probability $Pr(\theta|I)$:

$$\Pr(\theta|D, I) \propto \Pr(D|\theta, I)\,\Pr(\theta|I). \quad (5)$$

Typically, we choose a Boltzmann distribution for the prior[32]:

$$\Pr(\theta|I) = \frac{1}{Z}\exp\{-E_{\mathrm{phys}}(\theta)\} \quad (6)$$

where $E_{\mathrm{phys}}(\theta)$ is a nonbonded force field, a linearly ramped Lennard-Jones potential,[33] and Z the partition function. Structures are represented with full atomic detail.

When sampling structural transitions, we assume that an atomic resolution structure of at least one conformational state is available. This structure will be encoded using a probabilistic network model. The second conformational state is the target state, which we would like to reach when starting from the first structure. In the morphing scenario, a full-atom structure of the target state is given. However, of higher practical relevance is a situation in which only sparse or low-resolution data about the target state are available.

Let us first consider the morphing scenario. The target state will be modeled by positional restraints. Let y_i denote the Cartesian coordinates of the i-th atom of the target structure, the probability of generating this position from our model θ is a three-dimensional spherical Gaussian

$$\Pr(y_i|\theta, \sigma, I) = \frac{1}{(2\pi\sigma^2)^{3/2}}\exp\left\{-\frac{1}{2\sigma^2}|y_i - x_i(\theta)|^2\right\} \quad (7)$$

where $x_i(\theta)$ is the position of the i-th atom in the model parameterized by the conformational degrees of freedom θ. The standard deviation σ assesses by how much we can deviate from the target structure. The complete likelihood function derived from the target structure is simply the product of probabilities (Equation 7):

$$L_{\mathrm{target}}(\theta) = \prod_i \Pr(y_i|\theta, I) = (2\pi\sigma^2)^{-3N/2}\exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{N}|y_i - x_i(\theta)|^2\right\} \quad (8)$$

The likelihood is effectively based on the RMSD between the model and the target structure. The standard deviation σ is directly related to the RMSD:

$$\langle \mathrm{RMSD}^2\rangle = \left\langle \frac{1}{N}\sum_i |y_i - x_i(\theta)|^2 \right\rangle = 3\sigma^2.$$

By setting σ to a fixed value, the simulation aims for a particular RMSD to the target structure ($\mathrm{RMSD} \approx \sqrt{3}\sigma$). However, it is also possible to estimate σ in the course of the simulation.[34]

As a second application, we also considered crosslinking data as input. The observation of a crosslink between atoms i and j is a binary event $c_{ij} = 1$ that we model by using a logistic function:

$$\Pr(c_{ij} = 1|\theta, \alpha, I) = \frac{1}{1+\exp\{\alpha\,(|x_i(\theta) - x_j(\theta)| - r_c)\}} \quad (9)$$

where r_c is the maximum extension of the crosslinker and α is the steepness of the logistic function. The logistic function has a sigmoidal shape and approaches a maximum crosslinking probability of one for distances smaller than r_c. For distances exceeding the length of the crosslinker, the crosslinking probability gradually drops to zero. The likelihood function resulting from a set of crosslinks is

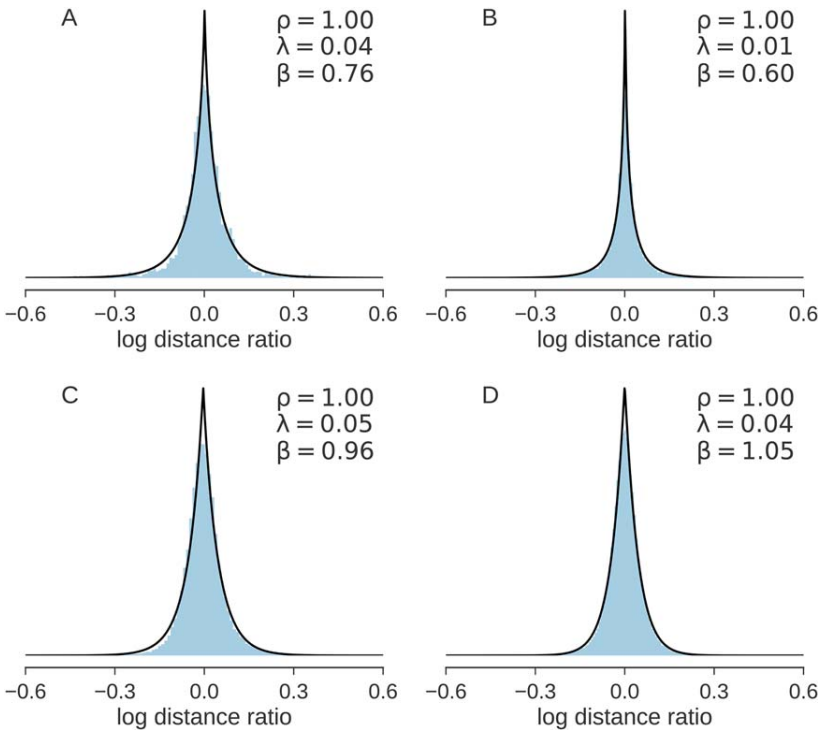

FIGURE 1 Empirical distributions of distance ratios in rigid domains and various structure ensembles (filled histograms) in comparison to fitted generalized lognormal models (black lines; parameter estimates are given in the figure legends). (A) Intradomain distances in GroEL derived by a comparison of the apo and holo conformation. (B) Distribution of log distance ratios in a standard NMR ensemble of adenylate kinase (PDB code 1p4s). (C) Distribution of log distance ratios in an ensemble of ubiquitin obtained with NMR ensemble refinement (PDB code 2k39). (D) Log distance ratios for an MD trajectory of adenylate kinase[38] (the trajectory is part of the MDAnalysis package[39]) [Color figure can be viewed at wileyonlinelibrary.com]

$$\log L_{\text{xlinks}}(\theta) = -\sum_{i,j} \log\left(1+\exp\left\{\alpha\left(|x_i(\theta)-x_j(\theta)|-r_c\right)\right\}\right) \qquad (10)$$

where the sum runs over all pairs of atoms for which a crosslink has been detected.

Structural samples are drawn from the posterior distribution (Equation 5) by using a Markov chain Monte Carlo algorithm.[35] We use Hybrid Monte Carlo, also known as Hamiltonian Monte Carlo[36,37] in internal coordinates, that is, θ are the main-chain and side-chain torsion angles which parameterize the Cartesian coordinates. Sampling of the conformational degrees of freedom is followed by sampling the other model parameters for a fixed structure. Most importantly, these are the network bonds z_{ij} and, in principle, also other unknown quantities such as the standard deviation σ as well as the parameters of the generalized lognormal components α_0, α_1. In the following, however, α_i were set to values suggested by an analysis of a database of conformational transitions.

3 | RESULTS AND DISCUSSION

3.1 | Small-scale distance fluctuations follow a generalized lognormal distribution

To develop a probabilistic model for proteins that undergo large conformational changes, let us first study the Cα distance fluctuations within parts that do not change their internal structure substantially and behave approximately as rigid domains. The DynDom database[29] offers a collection of pairs of protein structures that show two distinct conformational states. DynDom contains >3000 conformational changes of varying magnitude. Here, we consider all pairs of structures that differ by >5 Å Cα RMSD a large-scale conformational change. For all 450 pairs that satisfy this criterion, intradomain distances were computed for both conformational states and compared with each other. As explained in Methods, we used log distance ratios rather than distance differences to compare alternative conformational states.

Figure 1A shows a particular example that is representative of the whole set of pairs of structures involved in a large-scale conformational change. GroEL undergoes a domain closure motion during its functional cycle. The apo and holo conformation of GroEL (PDB codes 1oel and 1aon) exhibit an overall RMSD of ∼12 Å between both conformational states. Figure 1A shows the distribution of the log ratios of Cα distances that are below 10 Å in both conformations. The distribution is centered at zero and approximately symmetric. Its shape can be modeled quite accurately with a generalized lognormal distribution. The estimated parameters for GroEL are $\hat{\rho}=1.0, \hat{\lambda}=0.04$, and $\hat{\beta}=0.76$. Most intradomain distances do not change significantly during the domain closure movement, which is indicated by the central peak at zero. Only a small fraction of intradomain distances changes significantly during the structural transition and contribute to the tails of the distribution.

We find similar distributions of log distance ratios for NMR ensembles and molecular dynamics trajectories. Figures 1B,C show the

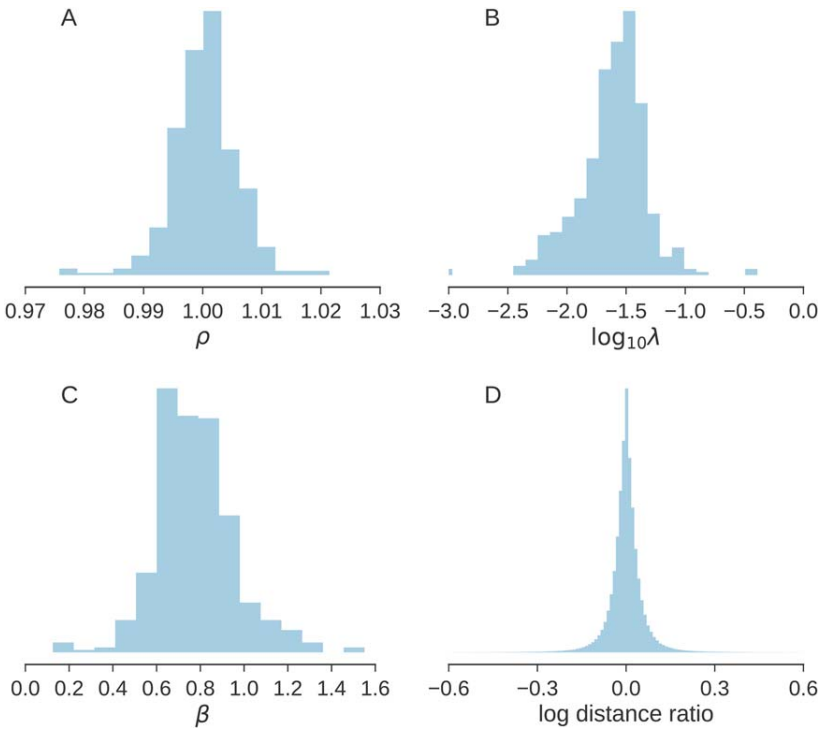

FIGURE 2 Analysis of 450 conformational transitions extracted from the DynDom database. Estimated parameters of the generalized lognormal model: mode ρ (A), logarithm of the scale parameters λ (B), and shape parameters β (C). Panel (D) shows a histogram of the pooled log distance ratios from all 450 examples [Color figure can be viewed at wileyonlinelibrary.com]

distributions of a standard NMR ensemble of adenylate kinase (PDB code 1p4s) and an ensemble of ubiquitin obtained by ensemble refinement (PDB code 2k39). The log distance ratios of both NMR ensembles can also be modeled quite accurately with the generalized lognormal distribution. Again the shape parameter β is smaller than one, indicating that the distribution of log distance ratios has heavier tails than the lognormal and even the log-Laplace distribution. Similar to the structures obtained with ensemble refinement, distance fluctuations in the simulated trajectory (Figure 1D) show a scale parameter β close to one, and can therefore be modeled with a log-Laplace distribution.

We fitted the generalized lognormal model to all log ratios of intradomain distances extracted from 450 large-scale conformational transitions in DynDom. This analysis shows that the parameters of the generalized lognormal distribution vary from case to case (see Figure 2A-C). The average values of all three parameters are $\rho = 1.0$, $\lambda = 0.03$, and $\beta = 0.78$. If we pool the distance ratios extracted from all 450 structural transitions (shown in Figure 2D), we obtain similar estimates: $\rho = 1.0$, $\lambda = 0.02$, and $\beta = 0.63$. To simplify the calculation of restraints, we will use the log-Laplace distribution to describe distance fluctuations (that is, we fix the shape parameters to $\beta = 1$). We then obtain $\rho = 1.$ and $\lambda = 0.04$, as optimal fit of the pooled set of distance fluctuations.

To summarize, NMR ensembles, MD trajectories and rigid domains of protein undergoing large-scale conformational changes all show similar distance fluctuations. To assess these fluctuations, it is more adequate to model the distribution of log distance ratios rather than distance differences. The generalized lognormal model (Equation 1) is flexible enough to capture the distribution of small-amplitude distance fluctuations.

3.2 | Adaptive network model

Large-scale conformational transitions can approximately be described as rigid-body movements.[2] During a domain movement contacts within the rigid domains are mostly preserved, whereas contacts between different domains break. To model the rupture of bonds, we studied the distribution of the log distance ratios between Cα atoms that belong to different rigid domains. The interdomain log distance ratios are typically skewed depending on the direction of the conformational change (closed-to-open or open-to-closed). However, if we pool all interdomain log distance ratios from the 450 large-scale transitions, we observe a zero-centered heavy-tailed distribution that can be approximated with a generalized lognormal distribution (the estimated shape parameter is $\beta = 0.85$).

Based on our previous findings, we chose log-Laplace models to describe intradomain and interdomain distance fluctuations. We fixed the shape parameter $\beta = 1$, even though the database analysis suggested a smaller value, corresponding to a model with heavier tails than the log-Laplace distribution. Our choice was motivated by its simplicity, since it results in a restraining potential that does not require the evaluation of fractional powers (see Equation 1). Moreover, tests with $\beta < 1$ indicated that the exact choice of β does not have a significant impact on the final model.

We can now devise a dynamic network model that allows for the rupture and formation of network bonds by means of a two-component mixture model. If a protein undergoes large conformational changes such as the opening of a domain, some contacts derived from the reference structure will be broken. We do not know which contacts will be lost during the structural transition. For every contact between Cα atoms from residues i and j, we therefore introduce a binary variable $z_{ij} \in \{0, 1\}$ which indicates, if the bond is formed ($z_{ij} = 1$) or broken ($z_{ij} = 0$). The parameter $w \in [0, 1]$ quantifies the probability that a bond remains intact during a conformational change, such that without any additional structural information, z_{ij} follows a Bernoulli distribution

$$\Pr(z_{ij}|w) = w^{z_{ij}} (1-w)^{1-z_{ij}}, \quad z_{ij} \in \{0, 1\}. \tag{11}$$

Our analysis of the 450 transitions from DynDom suggests $w \approx 0.92$.

Given a reference structure θ that defines the network the conditional posterior probability for the formation of bonds is:

$$\Pr(z_{ij}|\theta, w) \propto [w\,\mathrm{GLN}(r_{ij}; r_{ij}(\theta), \lambda_1, 1)]^{z_{ij}} [(1-w)\,\mathrm{GLN}(r_{ij}; r_{ij}(\theta), \lambda_0, 1)]^{1-z_{ij}} \tag{12}$$

where i, j are pairs of amino acids whose Cα distances r_{ij} are smaller than 10 Å, and $r_{ij}(\theta)$ is the distance in the model parameterized by the conformational degrees of freedom θ. The posterior probability (Equation 12) follows by multiplying the prior probability of the bond variables (Equation 11) with the likelihood (Equation 4). Our analysis of the DynDom entries yielded $\lambda_1 \approx 0.04$ and $\lambda_0 \approx 0.2$: the amplitude of distance fluctuations of preserved contacts is five times smaller than the amplitude of distance fluctuations in broken contacts. We sample the bond variables z_{ij} during the simulation of the structural transition by generating binary variables according to probability (Equation 12).

3.3 | Sampling conformational transitions

Given a configuration of the bond variables z_{ij}, the network imposes restraints on the conformational degrees of freedom. The restraining potential resulting from the network is:

$$E_{\mathrm{network}}(\theta) = \sum_{(i,j)} \left(\frac{z_{ij}}{\lambda_1} + \frac{1-z_{ij}}{\lambda_0} \right) |\ln [r_{ij}(\theta)/r_{ij}]|. \tag{13}$$

If a bond is switched on ($z_{ij} = 1$), the network restraint pulls with a force constant of $1/\lambda_1 = 25$. If the bond is switched off, the pulling force is five times weaker $1/\lambda_0 = 5$.

As a first application of the network model, we considered the simulation of structural transitions for which an atomic resolution structure of the target state is known. To guide the simulation toward the target state, we used positional restraints (Equation 8 in Methods). The conditional posterior distribution over the conformational degrees of freedom is:

$$\Pr(\theta|z_{ij}, I) \propto \exp\{-E_{\mathrm{target}}(\theta) - E_{\mathrm{network}}(\theta) - E_{\mathrm{phys}}(\theta)\} \tag{14}$$

where $E_{\mathrm{target}}(\theta) = -\log L_{\mathrm{target}}(\theta)$.

The algorithm for generating structural transitions iterates over the following updates (where t denotes an iteration index):

$$\begin{aligned} \theta^{(t+1)} &\sim \Pr(\theta|z_{ij}^{(t)}, I) \\ z_{ij}^{(t+1)} &\sim \Pr(z_{ij}|\theta^{(t+1)}, I) \\ w^{(t+1)} &\sim w^{\sum_{(i,j)} z_{ij}-1} (1-w)^{\sum_{(i,j)} (1-z_{ij})-1} \end{aligned} \tag{15}$$

The first update samples all conformational degrees of freedom θ by using Hybrid/Hamiltonian Monte Carlo.[36,37] In the second update, network bonds are switched on or off by sampling the indicators z_{ij} from Bernoulli distributions. In the last step, the fraction of preserved network bonds w is sampled from a Beta distribution.

We simulated 94 conformational transitions studied by Sfriso et al.[25] to assess the accuracy of the Gibbs sampler (Equation 15). The benchmark contains examples of varying degree of difficulty including several closed-to-open transitions, which pose problems to standard flexible fitting applications. For each pair of structures, we derived an initial network model and used positional restraints (Equation 8) to encode the information about the target structure. By running the Gibbs sampler (Equation 15), we generated stochastic morphs between the initial and the target structure (details about running times, memory consumption and the final RMSD can be found in the Supporting Information).

For all of the 94 transitions, our Gibbs sampler generated a final model that was closer than 2 Å to the target state. Figure 3A shows the RMSDs between the final structures and the target structure. On average the RMSD to the target state is 0.7 ±0.3 Å.

During the simulation of a structural transition, network bonds can break and form again. This is exemplified for the transition in 5′-nucleotidase (5′-NTase). The fraction of the preserved network bonds w decreases to 86%; after the target state has been reached, 98% of the bonds are recovered. By decreasing w our network model allows for the transient rupture of bonds.

We also studied the transition path generated during the morphing simulation. For 5′-NTase an atomic resolution structure of an intermediate state is available (PDB code 1oi8, chain A). Figure 3C shows the evolution of the RMSD of the sampled structure to the initial, the intermediate and the target state. The conformational change is sampled within the first 200 iterations of Gibbs sampling; structural changes sampled in later steps are smaller adaptations. During the structural transition simulated with the Gibbs sampler, conformations are generated that are close to the intermediate structure (the minimum RMSD for the forward simulation is 3.1 Å, for the reverse simulation the smallest RMSD is 1.9 Å).

In Figure 3D, we show a projection of the transition paths onto two collective variables: the domain opening angle χ^1 and the tilt angle χ^2.[40] The transitions sampled with our network model are similar to those reported by Krug et al.[41] and resemble the transition path sampled by GOdMD[42] (see Supporting Information for more details). Moreover, the opening and tilt angles of the experimental structures lie in close vicinity of the transition paths. Because the intermediate structures are sampled in torsion angle space, they are stereochemically intact and free of clashes. This is also reflected by the PROSA score[43] (Figure 3E), which does not deviate substantially from the value achieved by the initial structure.

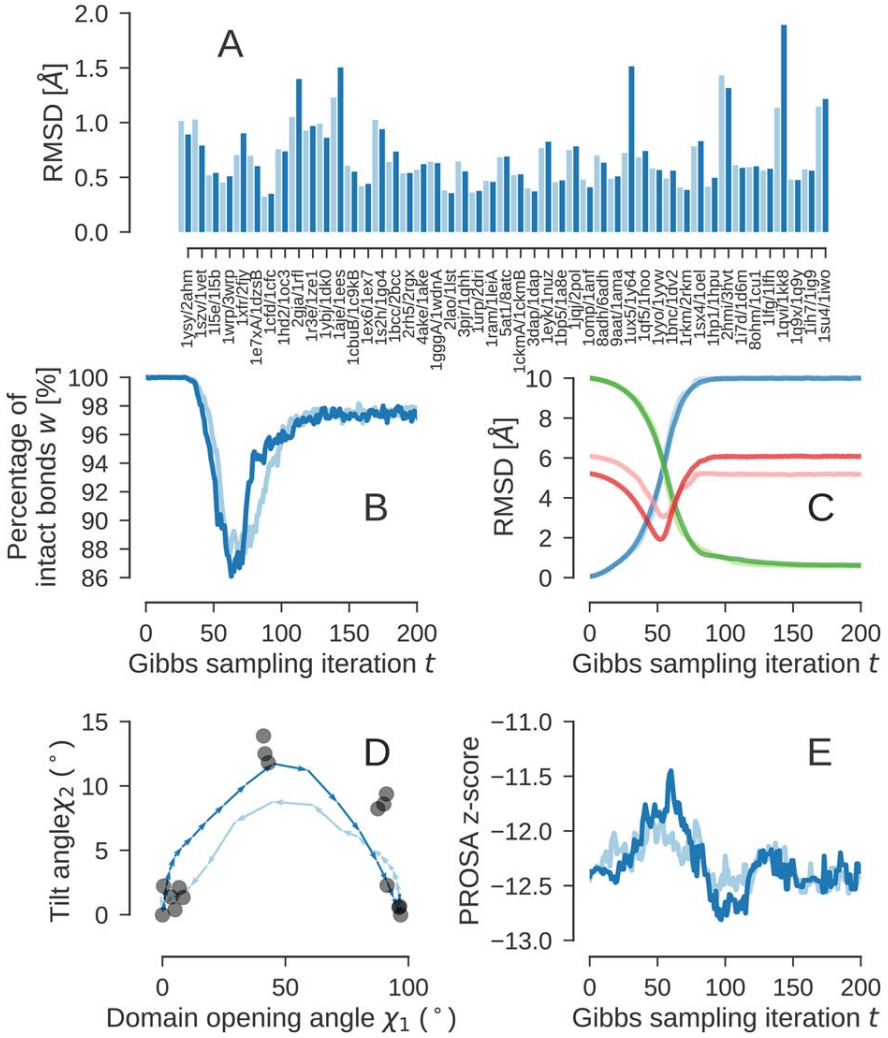

FIGURE 3 Simulation results for a benchmark of 94 structural transitions (light blue: forward transition, dark blue: reverse transition). (A) Cα RMSD between the final structure after 5000 steps of Gibbs sampling and the target state. (B) Evolution of the percentage of intact bonds during Gibbs sampling of the conformational change of 5′-NTase. (C) RMSD to the initial (blue lines), target (green lines) and an intermediate structure of 5′-NTase (red lines). The RMSDs of the forward transition (1hp1 → 1hpu) are shown in light colors, the reverse transition (1hpu → 1hp1) is shown in dark colors. (D) Projection of the forward and reverse paths onto the domain opening and tilt angles. Black dots indicate the values of the collective variables found in the experimental structures. (E) Evolution of the PROSA *z* scores during the forward and backward transition of 5′-Ntase [Color figure can be viewed at wileyonlinelibrary.com]

More examples of transition paths and a comparison to high-resolution structures of intermediate states are discussed in the Supporting Information, which also includes movies showing the forward and reverse transition of adenylate kinase and GroEL.

3.4 | Flexible fitting with crosslinking restraints

As an integrative modeling application of the adaptive network model, we studied the use of data from crosslinking/mass spectrometry (XL-MS) for modeling conformational changes. XL-MS is a powerful method to characterize the structure of large assemblies and provides useful distance information that is complementary to other structural data obtained, for example, with cryo-electron microscopy. Here, we consider crosslinking data to model an alternative conformational state that has not been characterized by high-resolution structure determination methods. This application is highly relevant for modeling large assemblies whose subunits undergo large conformational changes upon complex formation.

Crosslinking data provide only sparse distance information, which in its own right is not sufficient to build a complete model of the alternative conformational state. Therefore, we combine the crosslinking restraints with an adaptive network model encoding the local structure of the rigid domains. The domain boundaries are not known and will be determined implicitly by breaking some of the network bonds.

We ran Xwalk[44] to predict crosslinking data for the closed conformation of GroEL (PDB code 1xck) and derived an adaptive network model based on the open conformation (PDB code 1aon). Xwalk identifies a total of 395 lysine-lysine crosslinks for the closed conformation

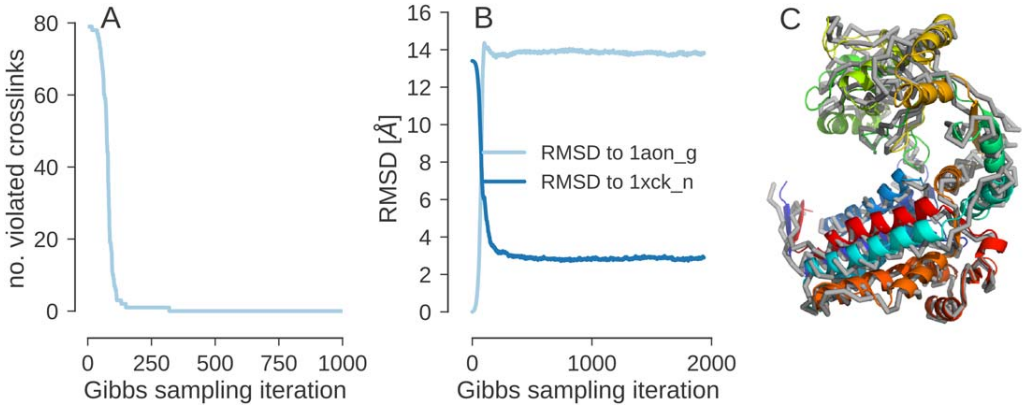

FIGURE 4 Crosslink based modeling of GroEL. (A) Number of violated crosslinks. (B) Evolution of the RMSD to the initial and the target structure during flexible fitting against the crosslinks. (C) Final structure obtained with the network model shown as colored cartoon, target structure (1xck) shown as gray ribbon [Color figure can be viewed at wileyonlinelibrary.com]

of GroEL. Most crosslinks are consistent with the open conformation, but a fraction of 79 crosslinks is violated. We incorporated all crosslinks into the model by using a likelihood based on a logistic model, which defines restraints between the Cα atoms of the crosslinked lysines and uses a threshold distance of $r_c = 34$ Å to assess the probability that a contact is formed in the model structure. The steepness of the logistic function α was set to a value of 100 $Å^{-1}$.

We ran the Gibbs sampler to generate a structural transition that minimizes the number of violated crosslinks, but still maintains the integrity of the structure. After ∼ 320 sampling steps, the number of violations dropped to zero (Figure 4A). The initial RMSD to the target structure is 13.4 Å. Within the first 200 steps of Gibbs sampling the RMSD to the target structure drops to ∼ 3 Å, while the RMSD to the initial structure increases to 13.6 Å. The final structure is shown in

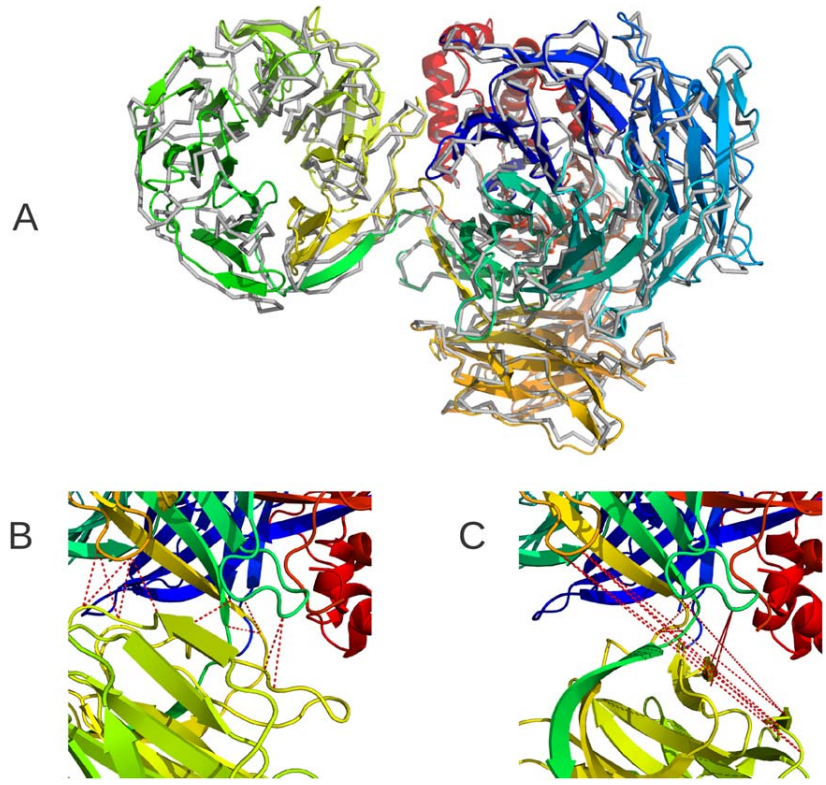

FIGURE 5 Crosslink based modeling of DNA damage-binding protein 1. (A) Superposition of the final crosslink based model (shown in colored cartoon representation) and the crystal structure (shown as gray ribbon). The bottom row shows close-ups of the interface between the three beta-propellers. Contacts that will be broken during the conformational transition are highlighted as red dashed lines. Panel (B) shows the initial structure, which was used to derive the adaptive network. Panel (C) shows the model obtained with Gibbs sampling. The initial contacts are heavily violated in the model structure

Figure 4C. If we switch off the network model and fit against the crosslinks only, the number of violated crosslinks will also be reduced to zero, first achieving a similar RMSD to the target structure. But as the simulation continues, the structure will partially unfold such that the RMSD increases to $\sim$ 10 Å over a period of 10^4 Gibbs sampling iterations. The simulation with adaptive network restraints, on the other hand, remains stable with an RMSD of $\sim$ 3 Å over the entire simulation.

As a second example illustrating the use of the adaptive network model, we modeled DNA damage-binding protein 1 (DDB1). DDB1 consists of three beta-propellers and has a total size of 1105 amino acids. The conformational transition involves a rotation of one beta-propeller relative to the other two beta-propellers. We used Xwalk to generate 394 intramolecular crosslinks for DDB1 in complex with DNA damage-binding protein 2 (PDB code 3ei3, chain A). The simulation started from the structure of the DDB1-CUL4A ubiquitin ligase machinery (PDB code 2hye, chain A). The RMSD between the initial and the target structure is 14.5 Å. In the initial structure, 32 crosslinks are violated. These crosslinks guide the simulation toward a model that satisfies all crosslinks and has an RMSD of $\sim$ 2.5 Å to the target structure (see Figure 5). During the conformational transition, 10 contacts are switched off driven by the crosslinks by setting the corresponding z_{ij} variables to zero. These contacts are located in the interface between the three β-propellers and detected automatically by our Gibbs sampling algorithm (see Figure 5).

4 | CONCLUSION

This article introduces a probabilistic network model that can be combined with experimental data to infer large conformational transitions in biomolecules. Our model is particularly suited to effectively sample conformational transitions when knowledge about the location of hinge regions and rigid domains is missing. Our adaptive network model is based on a statistical analysis of distance ratios in rigid domains, which can be described by a generalized lognormal distribution. We demonstrate that the network model can be combined with sparse experimental data such as crosslinking information detected with mass spectrometry. In the future, we plan to combine the probabilistic network model with various types of experimental data from solution scattering, nuclear magnetic resonance, and cryo-electron microscopy.

ACKNOWLEDGMENTS

This work has been supported by Deutsche Forschungsgemeinschaft (DFG) SFB 860, project B09. The authors declare no conflict of interest.

ORCID

Michael Habeck http://orcid.org/0000-0002-2188-5667

REFERENCES

[1] Henzler-Wildman K, Kern D. Dynamic personalities of proteins. *Nature*. 2007;450(7172):964–972.

[2] Gerstein M, Lesk AM, Chothia C. Structural mechanisms for domain movements in proteins. *Biochemistry*. 1994;33(22):6739–6749.

[3] Berman HM, Westbrook J, Feng Z, et al. The protein data bank. *Nucleic Acids Res*. 2000;28(1):235–242. Jan

[4] Fischer N, Neumann P, Konevega AL, et al. Structure of the E. coli ribosome-EF-Tu complex at <3 Å resolution by Cs-corrected cryo-EM. *Nature*. 2015;520(7548):567–570.

[5] Khatter H, Myasnikov AG, Natchiar SK, Klaholz BP. Structure of the human 80S ribosome. *Nature*. 2015;520(7549):640–645.

[6] Simonetti A, Marzi S, Myasnikov AG, et al. Structure of the 30S translation initiation complex. *Nature*. 2008;455(7211):416–420.

[7] Spahn CM, Penczek PA. Exploring conformational modes of macromolecular assemblies by multiparticle cryo-EM. *Curr Opin Struct Biol*. 2009;19(5):623–631.

[8] Fischer N, Konevega AL, Wintermeyer W, Rodnina MV, Stark H. Ribosome dynamics and tRNA movement by time-resolved electron cryomicroscopy. *Nature*. 2010;466(7304):329–333.

[9] Hanson JA, Duderstadt K, Watkins LP, et al. Illuminating the mechanistic roles of enzyme conformational dynamics. *Proc Natl Acad Sci USA*. 2007;104(46):18055–18060.

[10] Karplus M, McCammon JA. Molecular dynamics simulations of biomolecules. *Nat Struct Biol*. 2002;9(9):646–652.

[11] Karplus M, Kuriyan J. Molecular dynamics and protein function. *Proc Natl Acad Sci USA*. 2005;102(19):6679–6685.

[12] Saunders MG, Voth GA. Coarse-graining methods for computational biology. *Annu Rev Biophys*. 2013;42:73–93.

[13] Ueda Y, Taketomi H, Gō N. Studies on protein folding, unfolding, and fluctuations by computer simulation. II. A. Three-dimensional lattice model of lysozyme. *Biopolymers*. 1978;17:1531–1548.

[14] Tirion MM. Large amplitude elastic motions in proteins from a single-parameter, atomic analysis. *Phys Rev Lett*. 1996;77(9):1905–1908.

[15] Haliloglu T, Bahar I, Erman B. Gaussian dynamics of folded proteins. *Phys Rev Lett*. 1997;79(16):3090–3093.

[16] Bahar I, Atilgan AR, Erman B. Direct evaluation of thermal fluctuations in proteins using a single-parameter harmonic potential. *Folding Des*. 1997;2(3):173–181.

[17] Temiz NA, Meirovitch E, Bahar I. Escherichia coli adenylate kinase dynamics: comparison of elastic network model modes with mode-coupling (15)N-NMR relaxation data. *Proteins*. 2004;57:468–480.

[18] Atilgan AR, Durell SR, Jernigan RL, Demirel MC, Keskin O, Bahar I. Anisotropy of fluctuation dynamics of proteins with an elastic network model. *Biophys J*. 2001;80(1):505–515.

[19] Whitford PC, Noel JK, Gosavi S, Schug A, Sanbonmatsu KY, Onuchic JN. An all-atom structure-based potential for proteins: bridging minimal models with all-atom empirical forcefields. *Proteins*. 2009;75:430–441.

[20] KurkcuogluBahar ZI, Doruker P. ClustENM: ENM-based sampling of essential conformational space at full atomic resolution. *J Chem Theory Comput*. 2016;12:4549–4562.

[21] Miyashita O, Gorba C, Tama F. Structure modeling from small angle X-ray scattering data with elastic network normal mode analysis. *J Struct Biol*. 2011;173(3):451–460.

[22] Orzechowski M, Tama F. Flexible fitting of high-resolution x-ray structures into cryoelectron microscopy maps using biased molecular dynamics simulations. *Biophys J*. 2008;95(12):5692–5705.

[23] Schröder GF, Brunger AT, Levitt M. Combining efficient conformational sampling with a deformable elastic network model facilitates structure refinement at low resolution. *Structure*. 2007;15(12):1630–1641.

[24] Maragakis P, Karplus M. Large amplitude conformational change in proteins explored with a plastic network model: adenylate kinase. *J Mol Biol*. 2005;352:807–822.

[25] Sfriso P, Emperador A, Orellana L, Hospital A, Gelpí JL, Orozco M. Finding conformational transition pathways from discrete molecular dynamics simulations. *J Chem Theory Comput*. 2012;8(11):4707–4718.

[26] Rieping W, Habeck M, Nilges M. Modeling errors in NOE data with a lognormal distribution improves the quality of NMR structures. *J Am Chem Soc*. 2005;127(46):16026–16027.

[27] Geman S, Geman D. Stochastic relaxation, gibbs distributions, and the Bayesian restoration of images. *IEEE Trans*. 1984;PAMI-6(6):721–741.

[28] McLachlan G, Peel D. *Finite Mixture Models*. New York: John Wiley & Sons; 2004.

[29] Lee RA, Razaz M, Hayward S. The DynDom database of protein domain motions. *Bioinformatics*. 2003;19(10):1290–1291.

[30] Hayward S, Berendsen HJ. Systematic analysis of domain motions in proteins from conformational change: new results on citrate synthase and T4 lysozyme. *Proteins*. 1998;30(2):144–154.

[31] Rieping W, Habeck M, Nilges M. Inferential structure determination. *Science*. 2005;309(5732):303–306.

[32] Habeck M, Nilges M, Rieping W. Bayesian inference applied to macromolecular structure determination. *Phys Rev E*. 2005;72(3):031912.

[33] Habeck M. Statistical mechanics analysis of sparse data. *J Struct Biol*. 2011;173(3):541–548.

[34] Habeck M, Rieping W, Nilges M. Weighting of experimental evidence in macromolecular structure determination. *Proc Natl Acad Sci USA*. 2006;103(6):1756–1761.

[35] Habeck M, Nilges M, Rieping W. Replica-exchange Monte Carlo scheme for Bayesian data analysis. *Phys Rev Lett*. 2005;94(1):018105–0181054.

[36] Duane S, Kennedy AD, Pendleton B, Roweth D. Hybrid Monte Carlo. *Phys Lett B*. 1987;195(2):216–222.

[37] Neal RM. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*; 2011;2(11).

[38] Beckstein O, Denning EJ, Perilla JR, Woolf TB. Zipping and unzipping of adenylate kinase: atomistic insights into the ensemble of open closed transitions. *J Mol Biol*. 2009;394(1):160–176.

[39] Michaud-Agrawal N, Denning EJ, Woolf TB, Beckstein O. Mdanalysis: A toolkit for the analysis of molecular dynamics simulations. *J Comput Chem*. 2011;32(10):2319–2327.

[40] Knöfel T, Norbert Sträter E. coli 5-nucleotidase undergoes a hinge-bending domain rotation resembling a ball-and-socket motion. *J Mol Biol*. 2001;309(1):255–266.

[41] Krug U, Alexander NS, Stein RA, et al. Characterization of the domain orientations of E. coli 5′-nucleotidase by fitting an ensemble of conformers to DEER distance distributions. *Structure*. 2016;24(1):43–56.

[42] Sfriso P, Emperador A, Orozco M, et al. Exploration of conformational transition pathways from coarse-grained simulations. *Bioinformatics*. 2013;29(16):1980–1986.

[43] Wiederstein M, Sippl MJ. Prosa-web: interactive web service for the recognition of errors in three-dimensional structures of proteins. *Nucleic Acids Res*. 2007;35(suppl_2):W407–W410.

[44] Kahraman A, Malmstrom L, Aebersold R. Xwalk: computing and visualizing distances in cross-linking experiments. *Bioinformatics*. 2011;27(15):2163–2164.

SUPPORTING INFORMATION

Additional Supporting Information may be found online in the supporting information tab for this article.

How to cite this article: Habeck M, Nguyen T. A probabilistic network model for structural transitions in biomolecules. *Proteins*. 2018;86:634–643. https://doi.org/10.1002/prot.25490

5.1 Support Information

Supporting information

A probabilistic network model for structural transitions in biomolecules

Michael Habeck[1,2,*], Thach Nguyen[2]

February 12, 2018

Running times and computational resources

We ran the 94 morphing simulations on a high throughput cluster (HPC)[1] where each node is a Ivy-Bridge Intel E5-2670 v2 2.5GHz processor with 64 GB memory. We simulated 5000 Gibbs sampling iterations for each morphing task. To sample a conformational transition, much fewer Gibbs sampling iterations ($\sim$ 200 iterations) are required (see Fig. 3 in manuscript). The running times, memory usage, initial and final RMSD are summarized in Supplementary Table S1.

Transition	Size (AA)	CPU time (s)	Avg memory (MB)	Initial RMSD (Å)	Final RMSD (Å)
1ysy_A $\rightarrow$ 2ahm_D	71	652.87	54.51	7.73	1.01
2ahm_D $\rightarrow$ 1ysy_A	71	1243.74	69.78	7.73	0.89
1szv_A $\rightarrow$ 1vet_B	91	838.47	71.3	6.31	1.02
1vet_B $\rightarrow$ 1szv_A	91	1784.59	96.32	6.31	0.79
1l5e_B $\rightarrow$ 1l5b_A	101	682.32	60.78	6.52	0.52
1l5b_A $\rightarrow$ 1l5e_B	101	808.26	63.72	6.52	0.55
1wrp_R $\rightarrow$ 3wrp_A	108	706.67	61.33	1.98	0.46
3wrp_A $\rightarrow$ 1wrp_R	108	760.08	62.74	1.98	0.53

[1] https://www.gwdg.de/application-services/high-performance-computing

1xfr_A → 2fjy_A	123	832.1	73.57	5.27	0.71
2fjy_A → 1xfr_A	123	1775.8	96.86	5.27	0.9
1e7x_A → 1dzs_B	129	806.11	68.37	3.4	0.69
1dzs_B → 1e7x_A	129	1064.5	68.9	3.4	0.6
1cfd_A → 1cfc_A	148	2237.67	120.81	5.21	0.33
1cfc_A → 1cfd_A	148	2520.83	120.79	5.21	0.34
1hd2_A → 1oc3_C	158	1086.58	86.92	8.56	0.76
1oc3_C → 1hd2_A	158	1134.79	86.47	8.56	0.73
2gja_B → 1rfl_A	162	2512.23	152.04	8.73	1.03
1rfl_A → 2gja_B	162	1103.06	92.05	8.73	1.39
1r3e_A → 1ze1_D	169	1823.95	154.31	1.53	0.92
1ze1_D → 1r3e_A	169	2062.04	133.78	1.53	0.96
1ybj_A → 1dk0_B	173	1123.31	93.68	5.64	0.99
1dk0_B → 1ybj_A	173	2435.46	125.75	5.64	0.88
1aje_A → 1ees_A	174	2799.43	182.67	6.75	1.24
1ees_A → 1aje_A	174	2545.91	144.22	6.75	1.51
1cbu_B → 1c9k_B	180	1165.33	89.65	3.11	0.62
1c9k_B → 1cbu_B	180	1107.19	91.32	3.11	0.55
1ex6_A → 1ex7_A	186	1232.52	96.29	3.64	0.42
1ex7_A → 1ex6_A	186	1388.97	96.85	3.64	0.43
1s2h_A → 1go4_D	190	1315.2	105.79	4.93	1.02
1go4_D → 1s2h_A	190	3587.34	179.48	4.93	0.93
1bcc_E → 2bcc_E	196	1263.48	98.75	7.45	0.63
2bcc_E → 1bcc_E	196	1227.62	98.75	7.45	0.73
2rh5_A → 2rgx_A	202	1349.35	104.29	5.85	0.55
2rgx_A → 2rh5_A	202	1259.65	104.38	5.85	0.54
4ake_A → 1ake_B	214	1527.12	109.95	7.14	0.56
1ake_B → 4ake_A	214	1349.17	106.12	7.14	0.63
1ggg_A → 1wdn_A	220	1624.13	112.31	5.34	0.65
1wdn_A → 1ggg_A	220	1369.03	109.18	5.34	0.63
2lao_A → 1lst_A	238	1707.31	115.12	4.7	0.39
1lst_A → 2lao_A	238	1448.57	114.31	4.7	0.35
3pjr_A → 1qhh_B	261	1813.44	135.3	8.31	0.64

1qhh_B → 3pjr_A	261	1929.56	128.72	8.31	0.55
1urp_D → 2dri_A	271	2292.07	147.4	4.2	0.36
2dri_A → 1urp_D	271	2096.51	129.76	4.2	0.38
1ram_B → 1lei_A	273	1916.94	133.83	3.07	0.47
1lei_A → 1ram_B	273	2168.85	132.73	3.07	0.46
5at1_C → 8atc_C	310	2166.19	146.9	2.36	0.68
8atc_C → 5at1_C	310	2485.8	145.75	2.36	0.68
1ckm_A → 1ckm_B	317	2227.09	159.69	3.49	0.52
1ckm_B → 1ckm_A	317	2115.98	151.3	3.49	0.53
3dap_B → 1dap_A	320	2228.18	169.47	4.28	0.41
1dap_A → 3dap_B	320	2019.66	146.78	4.28	0.38
1eyk_A → 1nuz_A	327	2283.2	188.2	4.54	0.76
1nuz_A → 1eyk_A	327	2203.42	150.46	4.54	0.82
1bp5_B → 1a8e_A	329	2299.09	187.58	6.78	0.45
1a8e_A → 1bp5_B	329	2289.36	153.91	6.78	0.47
1jqj_A → 2pol_A	366	2495.64	204.34	2.05	0.74
2pol_A → 1jqj_A	366	2404.76	169.51	2.05	0.78
1omp_A → 1anf_A	370	2606.05	203.76	3.77	0.48
1anf_A → 1omp_A	370	2503.97	170.32	3.77	0.4
8adh_A → 6adh_B	374	2514.05	215.74	1.35	0.7
6adh_B → 8adh_A	374	2545.7	219.51	1.35	0.63
9aat_A → 1ama_A	401	2735.77	237.72	1.66	0.48
1ama_A → 9aat_A	401	2814.81	242.44	1.66	0.51
1ux5_A → 1y64_B	411	2714.49	252.58	10.33	0.73
1y64_B → 1ux5_A	411	2779.33	257.7	10.33	1.51
1qf5_A → 1hoo_B	431	3964.11	287.62	2.17	0.68
1hoo_B → 1qf5_A	431	2952.76	256.8	2.17	0.74
1yyo → 1yyw	438	1733.47	114.92	17.46	0.59
1yyw → 1yyo	438	1706.23	119.44	17.46	0.56
1bnc_A → 1dv2_B	452	3130.23	207.14	3.92	0.48
1dv2_B → 1bnc_A	452	2976.04	274.85	3.92	0.56
1rkm_A → 2rkm_A	517	3777.25	298.17	3.08	0.42
2rkm_A → 1rkm_A	517	3598.92	332.81	3.08	0.38

1sx4_G → 1oel_F	524	3478.49	247.78	12.39	0.78
1oel_F → 1sx4_G	524	3296.45	323	12.39	0.83
1hp1_A → 1hpu_C	525	3710.32	279.07	10.01	0.41
1hpu_C → 1hp1_A	525	3356.42	246.47	10.01	0.49
2hmi_A → 3hvt_A	556	3664.31	282.81	3.45	1.43
3hvt_A → 2hmi_A	556	3672.59	266.81	3.45	1.32
1i7d_A → 1d6m_A	620	4234.92	299.65	3.4	0.61
1d6m_A → 1i7d_A	620	4207.54	421.49	3.4	0.58
8ohm_A → 1cu1_B	645	2964.66	195.97	4.49	0.6
1cu1_B → 8ohm_A	645	2951.66	227.06	4.49	0.6
1lfg_A → 1lfh_A	691	4814.59	399.16	6.43	0.56
1lfh_A → 1lfg_A	691	4711.94	488.74	6.43	0.58
1qvi_A → 1kk8_A	837	5185.55	647.53	27.4	1.14
1kk8_A → 1qvi_A	837	5317.37	594.08	27.4	1.89
1q9x_B → 1q9y_A	899	6244.06	764.38	5.43	0.48
1q9y_A → 1q9x_B	899	6323.5	664.86	5.43	0.47
1ih7_A → 1ig9_A	903	6169.42	808.31	6.49	0.57
1ig9_A → 1ih7_A	903	6303.24	731.22	6.49	0.56
1su4_A → 1iwo_A	994	6445.33	768.81	13.97	1.15
1iwo_A → 1su4_A	994	6445.56	928.44	13.97	1.21

Table S1: running times, memory usage and final RMSD for all 94 morphing simulations.

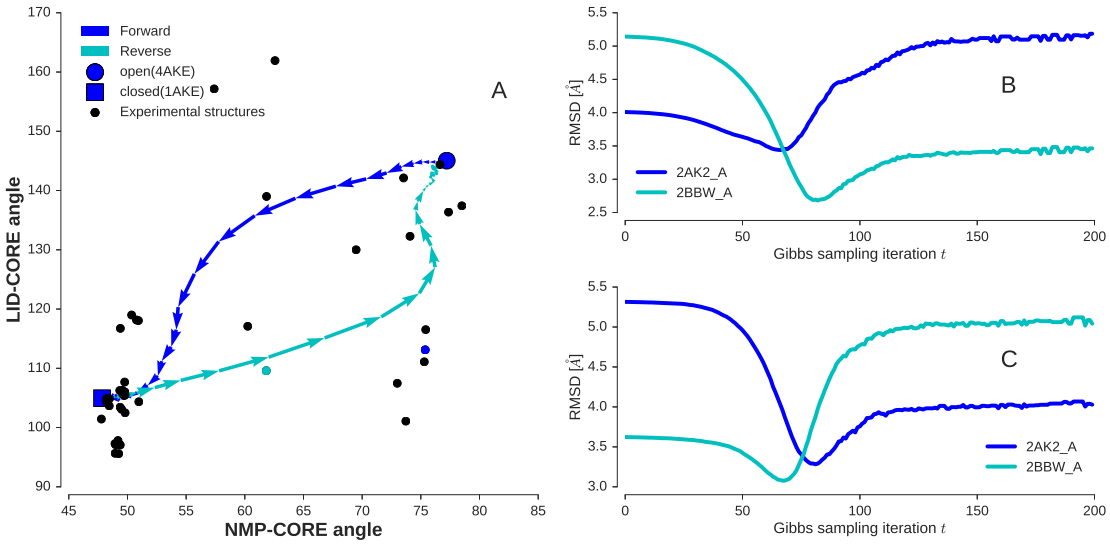

Figure S1: Forward and reverse transition in adenylate kinase. **(A)** The forward pathway connecting the open state (blue circle, PDB code 4ake) with the closed state (blue square, PDB code 1ake) is shown in blue, the reverse transition is indicated by cyan arrows. The pathways were projected onto the LID-CORE and NMP-CORE angle. Black circles indicate experimental structures. The blue and cyan circles mark intermediate structures (PDB entries 2ak2 and 2bbw). Panels **(B)** and **(C)** show the evolution of the global RMSD during the forward and reverse transition between our generated structures and the intermediate structures from PDB entries 2ak2 and 2bbw.

Detailed analysis of conformational transitions

Because structural transitions follow paths in a very high dimensional space, it is a non-trivial task to compare the transition paths generated by our Gibbs sampling algorithm with other pathways reported in the literature. We projected the transitions onto various reaction coordinates such intra-domain angles as well as principal components. We studied four examples in more detail.

Adenylate kinase

Adenylate kinase (AdK) is a phosphotransferase that catalyzes the reaction converting ATP and AMP into 2 ADP molecules. AdK is composed of three domains: NMP binding domain (residues 30-60), LID (residues 115-160) and the CORE domain (residues 1-

30, 61-114, and 161-214). Figure S1 presents the conformational change between the open (PDB code 4ake, chain A) and the closed state (PDB code 1ake, chain B). This large-scale structural transition can be captured by two intra-domain angles θ_{NMP} and θ_{LID} (Beckstein *et al.*, 2009). The NMP-CORE angle θ_{NMP} is the angle between the centers of mass of two segments L115-V125 and L35-A55 relative to I90-G100 based on Cα positions. The LID-CORE angle θ_{LID} is the angle between the centers of mass based on Cα positions of segments I179-E185 and V125-L153 relative to L115-V125. The forward and reverse transitions generated by our Gibbs sampler follow different pathways. Visual comparison with the analysis by Seyler *et al.* (2015) reveals that our transition path is close to the path generated by GOdMD (Sfriso *et al.*, 2013) in that the LID-CORE angle changes first and is followed by a transition in the NMP-CORE angle. Our reverse transition is close to the pathway generated with ANMPathway (Das *et al.*, 2014). Among 45 experimental structures of AdK deposited in the PDB, we identified several structures that are close to our transition paths in angular space. The closest intermediate structures based on global RMSD are PDB entries 2bbw (chain A) and 2ak2 (chain A).

GroEL

To illustrate the transition path between the T state (PDB code 1oel, chain F) and R" state (PDB code 1sx4, chain G) in GroEL, we used a reaction coordinate (RC) similar to the one defined by Zheng and Wen (2017). We defined the reacion coordinate RC_{S} to measure the movement of some domain S as follows:

$$RC_{\mathrm{S}} = (\delta X_{\mathrm{S}} \delta X_{\mathrm{S,obs}}) / |\delta X_{\mathrm{S,obs}}|^2 \tag{1}$$

where δX_{S} is the displacement vector from the center of mass of domain S in the initial structure to the center of mass of domain S in the intermediate structure. Figure S2 shows the movement of the Apical (A) and Intermediate (I) domain relative to the Equatorial (E) domain measured by RC_{AE} and RC_{IE}.

5'-nucleotidase

Escherichia coli 5'-nucleotidase (5'-NTase) is an enzyme composed of an N-terminal domain (residues 26-351) and a C-terminal domain (residues 365-550) that move relative to each other. To elucidate the transition pathway of 5'-NTase, we used two angles

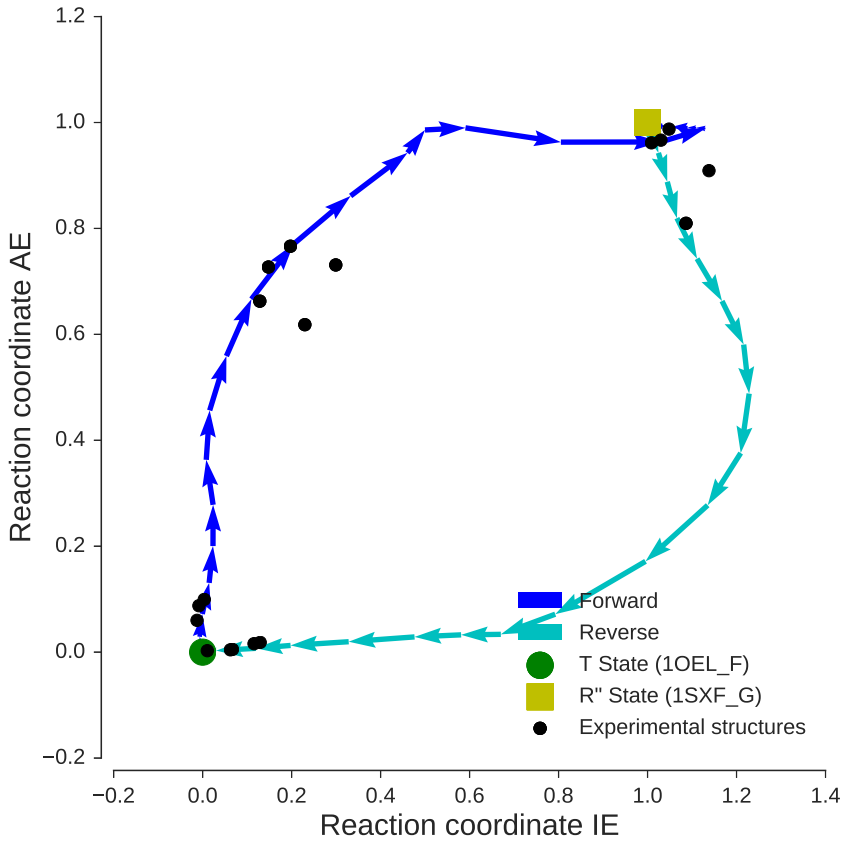

Figure S2: Analysis of the transition path of the T state (green circle) and R" state (yellow square) in GroEL. To visualize the transition, we used two reaction coordinates, RC_{IE} and RC_{AE}, where the A, I, and E domain were defined as in Xu *et al.* (1997) and computed by using equation 1. The black dots indicate experimental structures from the following PDB entries 4aaq (chains B, C), 4ab2 (chains A, B), 4aar (chains B, C), 4aau (chains B, C, J, K), 4pko (chains A, D, L, M), 3wvl (chains G, J), 1pf9 (chains A, D), 1sx4 (chain J), 2eu1 (chain F), 2c7e (chains L, M), 1xck (chain B), 1mnf (chain I).

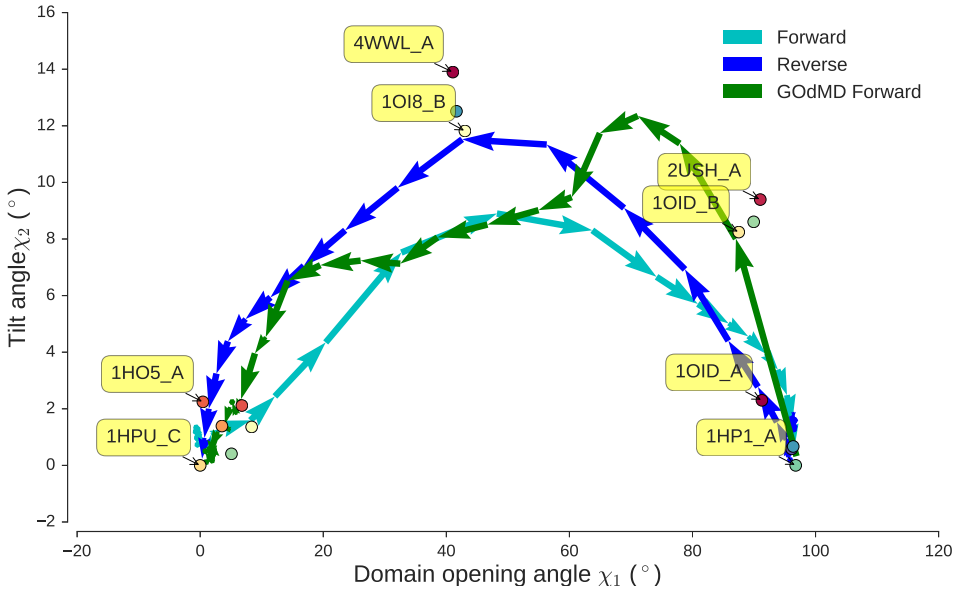

Figure S3: Transition paths in 5'-nucleotidase mapped onto two angles χ_1 and χ_2. Experimental structures (colored dots) are taken from PDB entries 1hpu (chains A, B, C, D), 1ho5 (chains A, B), 4wwl, 1oi8 (chains A, B), 2ush (chains A, B), 1oid (chains A, B), 1oie (chain A), 1ush (chain A), 1hp1 (chain A).

χ_1 (domain opening angle) and χ_2 (tilt angle) defined by Knöfel and Sträter (2001) and Krug *et al.* (2016). Figure S3 shows the evolution of χ_1 and χ_2 during the transition paths generated with our Gibbs sampler and highlights that the generated paths find an experimentally characterized intermediate state (PDB code 1oi8). For comparison, we also show the transition path for the forward direction generated with GOdMD by Sfriso *et al.* (2013), which steps through a similar sequence of collective variables (the GOdMD simulation for the reverse direction failed, producing a final structure with an RMSD of $\sim$7 Å to the target).

Ribonuclease III

Ribonuclease III (RNase III) is a ribonuclease that plays an important role in RNA processing. RNase III recognizes and cleaves *ds*RNA at several target locations to create mature RNAs (Gan *et al.*, 2005). We investigated the transition path of RNase III and compared our results with pathways generated by Orellana *et al.* (2016). We used principal component analysis (PCA) to project the transition path onto the first two principal components (PCs).
Figure S4 shows the forward and reverse transition starting from non-catalytic complex (PDB code 1yyo) and targeting the pre-catalytic complex (PDB code 1yyw). The forward transition comes close to the inactive dsRNA-bound state (PDB entries 1yyk, 2nue). The reverse transition comes close to the Mg2+ bound catalytic state (PDB entries 4m30, 4mz2).

Supplementary movies

Our supplementary movies show forward and reverse transitions in cartoon representation generated by Pymol (DeLano, 2002).

- Movies Adk_forward.avi and Adk_reverse.avi show the forward and reverse transition in Adenylate kinase wher NMP, LID and CORE domain are colored in red, green and blue. Structures were superimposed onto the CORE domain.
- Movies Groel_forward.avi and Groel_reverse.avi show the forward and reverse transition in GroEL where the Apical, Intermediate and Equatorial domain are col-

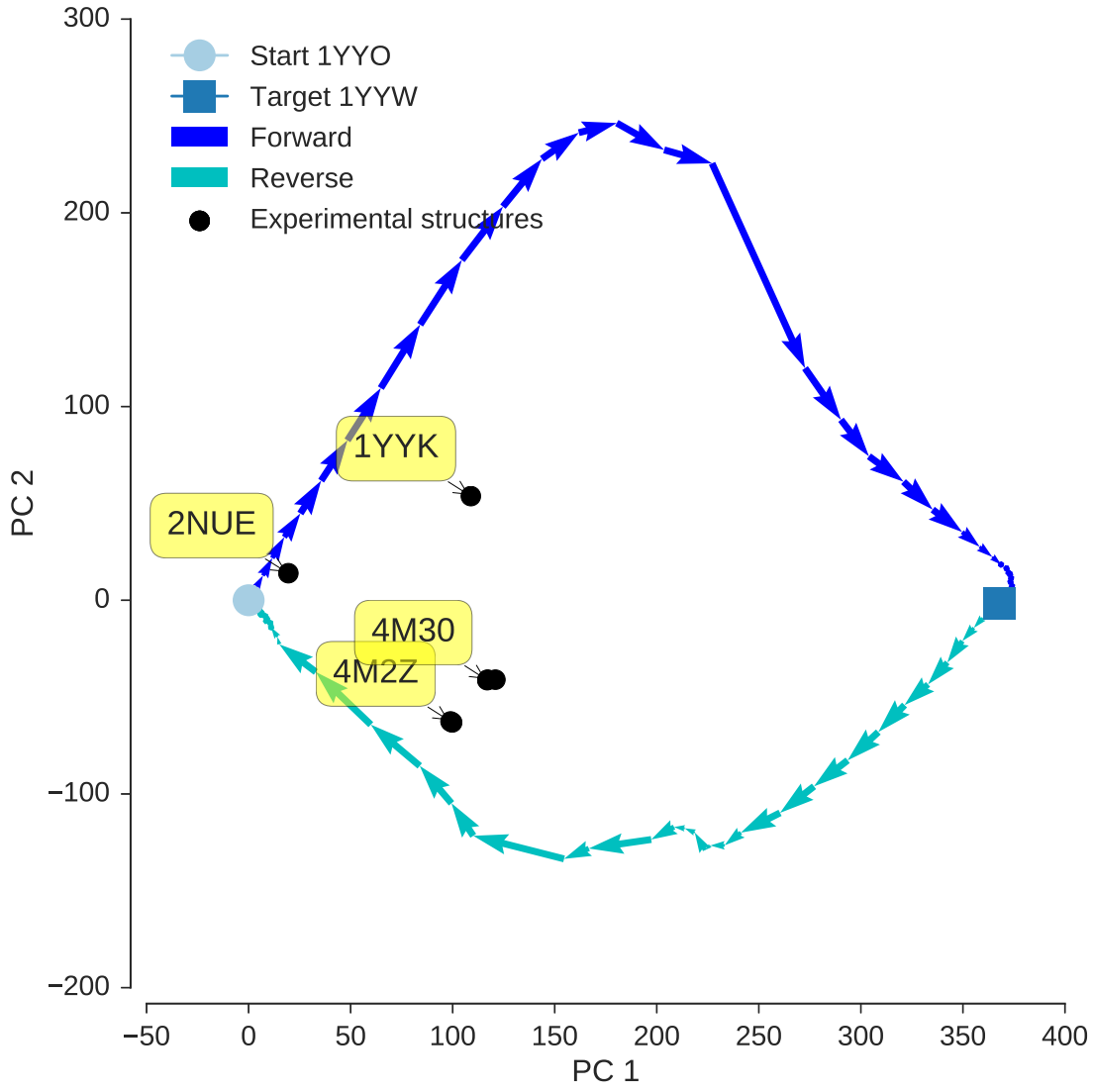

Figure S4: Conformational transition in RNase III starting from the non-catalytic complex (PDB code 1yyo) and targeting the pre-catalytic complex (PDB code 1yyw).

ored in red, green and blue. Structures were superimposed onto the Equatorial domain.

References

Beckstein, O., Denning, E. J., Perilla, J. R., and Woolf, T. B. (2009). *J. Mol. Biol.*, **394**(1), 160–176.

Das, A., Gur, M., Cheng, M. H., Jo, S., Bahar, I., and Roux, B. (2014). *PLoS Comput Biol*, **10**(4), e1003521.

DeLano, W. L. (2002).

Gan, J., Tropea, J. E., Austin, B. P., Court, D. L., Waugh, D. S., and Ji, X. (2005). *Structure*, **13**(10), 1435–1442.

Knöfel, T. and Sträter, N. (2001). *Journal of molecular biology*, **309**(1), 255–266.

Krug, U., Alexander, N. S., Stein, R. A., Keim, A., Mchaourab, H. S., Strater, N., and Meiler, J. (2016). *Structure*, **24**(1), 43–56.

Orellana, L., Yoluk, O., Carrillo, O., Orozco, M., and Lindahl, E. (2016). *Nature communications*, **7**.

Seyler, S. L., Kumar, A., Thorpe, M. F., and Beckstein, O. (2015). *PLoS Comput Biol*, **11**(10), e1004568.

Sfriso, P., Emperador, A., Orozco, M., *et al.* (2013). *Bioinformatics*, **29**(16), 1980–1986.

Xu, Z., Horwich, A. L., and Sigler, P. B. (1997). *Nature*, **388**(6644), 741–750.

Zheng, W. and Wen, H. (2017). *Current Opinion in Structural Biology*, **42**, 24–30.

Chapter 6

Discussion and Conclusion

6.1 Probabilistic model characterizes protein conformational change

In this thesis, we propose two probabilistic models that characterize protein conformational change. The first model segments proteins into rigid bodies, whereas the second model characterizes the protein in a more flexible way using an elastical network of contacts, which are preserved or break controlled by the experimental data from different conformational states. In the first model, we have developed a Gaussian mixture model that segments the protein structures into rigid bodies. The generative model parameters are sampled by a Gibbs sampler from the structure. However, this model requires us to set the number of rigid domains K, which we initially set to a relatively large value $K = 10$, the precise number of rigid bodies is determined by dropping the empty clusters. Unlike the other methods which usually support only pairs of structures, our first model supports multiple structures. We compare our results with others by focusing on particular examples. A stringent benchmark based on the Dyndom database was implemented. We evaluate the first result with more than 3000 original entries from Dyndom. In later work, we process around 400 entries, which show significant conformational changes. To get rid of the label switching problem, we transform the discrete label into label forgetting matrices. The evaluation against Dyndom achieves score based on the overlap (92%) and segmentation error (12%).
In our improved model (in preparation), we use fewer parameters while achieving better performance and more robust program. The initial conditions we obtain from the commu-

nity detection algorithm (the Louvain methods) (Blondel *et al.* (2008),Traag *et al.* (2011)) outperformed the results from the spectral clustering (Von Luxburg (2007)). In previous work, a problem of algorithms such as expectation-maximization and Gibbs sampling is that they only find local optima such that the quality of the segmentation strongly depends on the initial parameters. Here we try to find an efficient way of finding proper initial parameters based on the community detection algorithms of the Louvain methods.

6.1.1 Scope of our model

Our first model and the improved version often characterize the rigidity of proteins. However, the model is not a universal fit for all protein conformational changes. Our rigid model was derived from the structural fluctuations. It cannot detect the rigid domain if the amplitude of conformational change is too small. Intrinsically, our model alone cannot work with a single structure. An example of DEAH Box helicase shows the limitation of the rigid domain model. The DEAH-box helicase Prp43 is a motor protein that unwinds is a critical player in pre-mRNA splicing as well as the maturation of rRNAs. We analyzed four conformations from Tauchert *et al.* (2017): ctPrp43DN.ADP complex structure (PDB: 5D0U), ctPrp43DN.U7.ADP.BeF3(PDB: 5LTA), ctPrp43DN.ADP.BeF3(PDB: 5LTJ) and ctPrp43DN.ADP.BeF3 (PDB: 5LTK) the first 60 residues are disordered therefore discarded. Figure 6.1 shows that our result agrees with the result from literature in the location of the first three domains RecA1, RecA2, and WH. The last two domains Ratchetlike and OB, show minimal conformational change. Therefore they are merged into one big rigid domain by our approach. In this example, we can only compare our results with Spectrus, which required manual tuning of their quality score; our model solved this example automatically. Moreover, other methods do not support more than two input structures. Our result is in closer agreement with the literature than Spectrus's result.
To solve the small fluctuation problem, one could generate the full ensemble using molecular dynamic simulation or a Gaussian network model, which requires less computational cost.
Another limitation of our model is that it cannot characterize protein conformational changes in fibrous proteins. In this example, elongate filaments are segmented into a large number of tiny rigid bodies that are overfitting our model. We take a close look at Calmodulin. Calmodulin is a Calcium ion binding protein. Because of its size, NMR has been used to determine most of the Calmodulin structures. In our database, Calmodulin has 288 entries. From literature, Camodulin consists of 2 domains, the C-Terminal and N

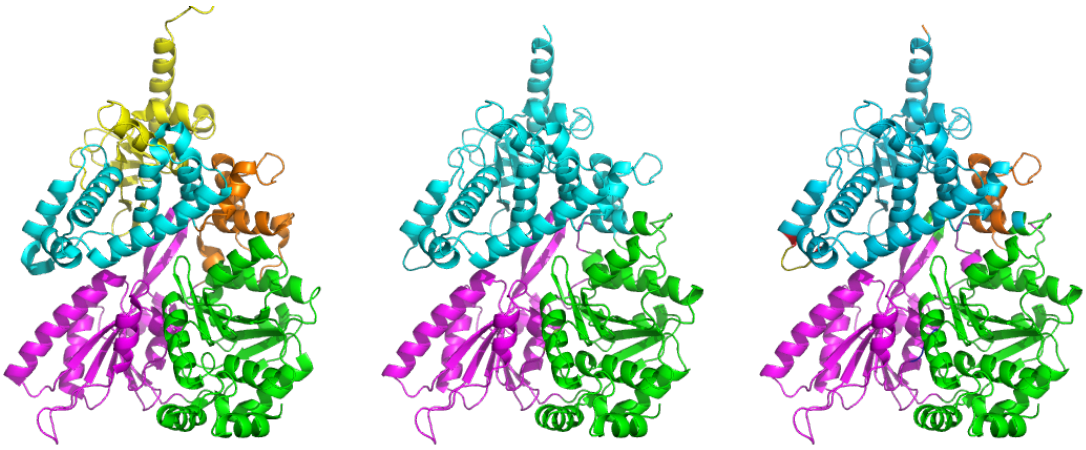

Figure 6.1: Structure of DEAH-box helicase Prp43 and segmentation set (left to right) from literature, Spectrus (Manually selected $K = 3$ based on quality score), and our method PRISM

terminal domain, which are well separated by a hinge in the middle of alpha-helix. Our model describes precisely two domains for pairs of input structure: 1CFC chain A and 1CFD chain A the Calcium free form. The small RMSD between the rigid domain shows that the model fits the structure well. However, the rigid model cannot characterize the rigid domain within the 1CLL chain A the Calcium-binding conformation, which indicates high RMSD between domains (See Figure 6.2). Our model generates four rigid domains with all conformation of the Calmodulin family. The two extra segmentation satisfies the mechanical transformation, which has small RMSD. The third problem is the model cannot characterize well for continuous conformational change. For example, when we process two input structures with several intermedia conformational, our model cannot well separate between rigid bodies. The example of doughnut shape like exportin CRM1 protein (Chromosome region maintenance 1) in two conformations the compact 3GB8(A) and extended 3GJX(A). Our model detects more segments that are fragmented into several small segments than the result from literature. Figure 6.3 show the heterogeneous structure of CRM1 entries with smooth distance transition. Those proteins can be characterized into a random coil or random filament. For these elastic conformational change, we propose the probabilistic network model for structural transition in biomolecules. Our model implements anharmonic spring that allows us to break or preserve with sampling

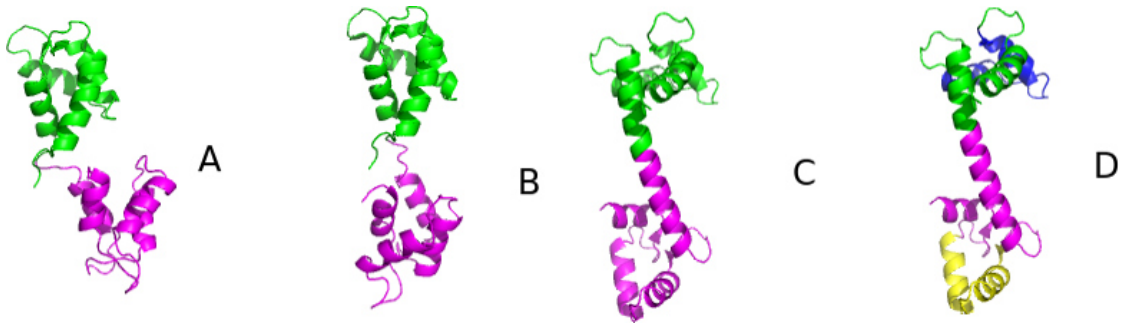

Figure 6.2: Our model describes three conformational structures of Calmodulin with PDB (chain) **(A)**: 1CFC(A), **(B)**: 1CFD(A), **(C)**: 1CLL(A). Structures are colored by segmentation domain obtained with from pairs of PDB(chain): 1CFC(A), 1CFD(A). Root mean square deviation (RMSD) of two conformations A versus B: 5.2 Å. RMSD between the first segments: 0.39 Å. RMSD of second segments 0.44 Å shows the model fit two conformations. Root mean square deviation (RMSD) of two conformations A versus C: 14.02 Å. RMSD between the first segments: 4.45 Å. RMSD of second segments 4.59 Å, show the model does not fit the third conformation. **(D)** Structures of 1CLL(A) are colored by our model for all three conformations

from experimental data. The probabilistic model generates the entire ensembles from one structure to the target structure. Our network model cooperates with crosslinking data to predict the conformational change in the large protein.
Our models require two or more structures. In case only one structure is available, we can use the Gaussian network model to generate the full ensemble of protein.

6.2 Webserver and computed dataset

We implement our model as a web service. We ran our program on the entire protein database. Protein clusters are derived from BlastClust (Altschul *et al.* (1997)) based on sequence similarity of greater than 95 %. Some webservers have similar functions as ours. Dyndom has two web implementations, the 1D and 3D versions, but supports only two input structures. The Molmov server and dataset by Gerstein (Gerstein and Krebs (1998)) and the recent Pdbflex (Hrabe *et al.* (2015)) extract the flexibility of protein and generate the morphing ensemble between conformations. However, it is unclear which regions are rigid and hinge parts. The Spectrus server (Ponzoni *et al.* (2015)) support multiple input structures, but the server requires human intervention ambiguous quality scores, which is heuristic and in some example, the smaller quality score gets better results, see examples in

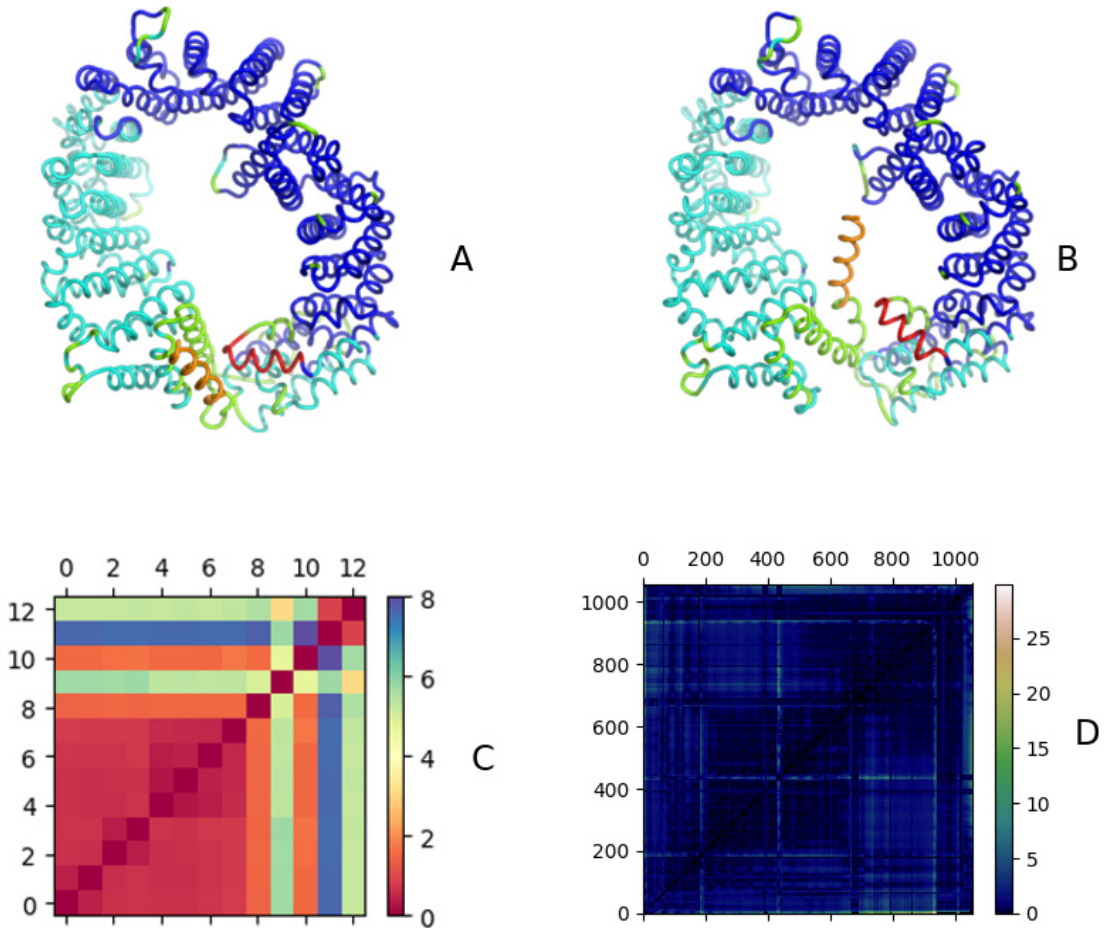

Figure 6.3: **(A)**, **(B)** Structures of CRM1 3GB8(A) and 3GJX(A) are colored by the rigid domain from our model. **(C)** the pairwise RMSD of 13 entries, **(D)** The average different distance matrix of each entry with the average distance matrix

Chapter 2. Besides, sequential alignment is not integrated into the webserver. The Codnas database (Monzon *et al.* (2013)) classifies protein conformations into a hierarchy, integrates diverse information from the experimental condition, but no domain information is given. Together with the rigid body information, our data set integrate the experimental condition data from Codnas to enrich the protein supplemental information.

6.3 A graph-based algorithm for detecting rigid domains in protein structures

Along with the probabilistic models above, we implement an alternative approach to detect the rigid bodies in protein structure. The new algorithms use graph-based. In this approach, the protein ensemble is presented as a weighted graph. To employing the Viterbi algorithm, we reduce the size of the protein graph by graph clustering method. Our test results show that two methods are close performance, which gives us more options to solve the problem.

6.4 Summary

The goal of this thesis was to develop a probabilistic model that characterizes protein conformational change and applies Bayesian methods to infer model parameters from data. The interpretation of the model allows for new insights into protein structures. In the first part of this thesis, we introduce probabilistic models for protein conformational changes. The model segments protein conformational changes into rigid bodies. The algorithm for estimating the model parameter was introduced and improved in follow up work. We implemented a webserver and published a dataset for benchmarking other algorithms. We introduce a new algorithm to characterize structural transitions in proteins using a graph-based approach. We construct a graph from a set of protein conformations and detect rigid domains via an edge labeling strategy. In the second part of the thesis, we propose the probabilistic network model for simulating conformational transitions. Two models are opposite application, but they cover a large spectrum of protein dynamics. In the general case, we can use our two models as a meta-model or hybrid model, which supports two particular problems. The first model supports mechanical transformation rigid bodies, and the second model is more flexible.

6.5 Outlook

The probabilistic model we propose can be expanded by incorporating various prior information as the potential of the Bayesian framework. We can integrate more physiological information such as solvent accessible surface area (SASA), charged information, temperature, or hydrophobic effects. In case we have enough data, a useful pattern can use to detect the rigid part of a protein. Our first rigid segmentation model is purely inferred from conformational change. Therefore the rigid segmentation result is nothing than numeric value as other model parameters. We can categorize the rigid domain into hierarchy structure. We can incorporate the segmentation with chemical properties and extract the underlying biological information such as ligand-binding sites or chemical agents.

Bibliography

Abyzov, A., Bjornson, R., Felipe, M., and Gerstein, M. (2010). *PROTEINS: Structure, Function, and Bioinformatics*, **78**(2), 309–324.

Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997). *Nucleic acids research*, **25**(17), 3389–3402.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). *Journal of statistical mechanics: theory and experiment*, **2008**(10), P10008.

Cock, P. J., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B., *et al.* (2009). *Bioinformatics*, **25**(11), 1422–1423.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). *J. R. Stat. Soc. B*, **39**, 1–38.

Emekli, U., Schneidman-Duhovny, D., Wolfson, H. J., Nussinov, R., and Haliloglu, T. (2008). *Proteins: Structure, Function, and Bioinformatics*, **70**(4), 1219–1227.

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

Flores, S. C. and Gerstein, M. B. (2007). *BMC bioinformatics*, **8**(1), 215.

Frank, J. (2002). *Annu Rev Biophys Biomol Struct*, **31**, 303–319.

Geman, S. and Geman, D. (1984). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-6**(6), 721–741.

Gerstein, M. and Krebs, W. (1998). *Nucleic acids research*, **26**(18), 4280–4290.

Gerstein, M., Lesk, A. M., and Chothia, C. (1994). *Biochemistry*, **33**(22), 6739–6749.

Haliloglu, T., Bahar, I., and Erman, B. (1997). *Physical review letters*, **79**(16), 3090.

Hanson, J. A., Duderstadt, K., Watkins, L. P., Bhattacharyya, S., Brokaw, J., Chu, J. W., and Yang, H. (2007). *Proc Natl Acad Sci USA*, **104**, 18055–18060.

Hayward, S. and Berendsen, H. J. (1998). *Proteins Structure Function and Genetics*, **30**(2), 144–154.

Henzler-Wildman, K. and Kern, D. (2007). *Nature*, **450**, 964–972.

Hinsen, K. (1998). *Proteins: Structure, Function, and Bioinformatics*, **33**(3), 417–429.

Hirsch, M. and Habeck, M. (2008). *Bioinformatics*, **24**, 2184–2192.

Hrabe, T., Li, Z., Sedova, M., Rotkiewicz, P., Jaroszewski, L., and Godzik, A. (2015). *Nucleic acids research*, **44**(D1), D423–D428.

Kabsch, W. (1976). *Acta Cryst.*, **A32**, 922–923.

Kalev, I., Mechelke, M., Kopec, K. O., Holder, T., Carstens, S., and Habeck, M. (2012). *Bioinformatics*, **28**(22), 2996–2997.

Karplus, M. (2002). Molecular dynamics simulations of biomolecules.

Karplus, M. and Kuriyan, J. (2005). *Proceedings of the National Academy of Sciences*, **102**(19), 6679–6685.

Keating, K. S., Flores, S. C., Gerstein, M. B., and Kuhn, L. A. (2009). *Protein Science*, **18**(2), 359–371.

Kendrew, J. C., Bodo, G., Dintzis, H. M., Parrish, R., Wyckoff, H., and Phillips, D. C. (1958). *Nature*, **181**(4610), 662–666.

Monzon, A. M., Juritz, E., Fornasari, M. S., and Parisi, G. (2013). *Bioinformatics*, **29**(19), 2512–2514.

Murzin, A. G., Brenner, S. E., Hubbard, T., and Chothia, C. (1995). *J. Mol. Biol.*, **247**, 536–540.

Nguyen, T. and Habeck, M. (2016). *Bioinformatics*, **32**(17), i710–i717.

Nguyen, T. and Habeck, M. (2020). **in preparation**.

Nichols, W. L., Rose, G. D., Ten Eyck, L. F., and Zimm, B. H. (1995). *Proteins: Structure, Function, and Bioinformatics*, **23**(1), 38–48.

Orengo, C. A., Michie, A. D., Jones, S., Jones, D. T., Swindells, M. B., and Thornton, J. M. (1997). *Structure*, **5**(8), 1093–1109.

Ponting, C. P. and Russell, R. R. (2002). *Annual review of biophysics and biomolecular structure*, **31**(1), 45–71.

Ponzoni, L., Polles, G., Carnevale, V., and Micheletti, C. (2015). *Structure*, **23**(8), 1516–1525.

Sievers, F., Wilm, A., Dineen, D., Gibson, T. J., Karplus, K., Li, W., Lopez, R., McWilliam, H., Remmert, M., Söding, J., *et al.* (2011). *Molecular systems biology*, **7**(1), 539.

Sonnhammer, E. L., Eddy, S. R., and Durbin, R. (1997). *Proteins: Structure, Function, and Bioinformatics*, **28**(3), 405–420.

Tauchert, M. J., Fourmann, J.-B., Lührmann, R., and Ficner, R. (2017). *Elife*, **6**.

Tirion, M. M. (1996). *Physical review letters*, **77**(9), 1905.

Traag, V. A., Van Dooren, P., and Nesterov, Y. (2011). *Physical Review E*, **84**(1), 016114.

Voet, D. and Voet, J. G. (2010). *Biochemistry*. John Wiley & Sons.

Von Luxburg, U. (2007). *Statistics and computing*, **17**(4), 395–416.

Wriggers, W. and Schulten, K. (1997). *Proteins Structure Function and Genetics*, **29**(1), 1–14.

Wüthrich, K. (1976). *NMR in biological research: peptides and proteins*. North-Holland Amsterdam.

Wüthrich, K. (2001). *Nature Structural & Molecular Biology*, **8**(11), 923.

Xu, Z., Horwich, A. L., and Sigler, P. B. (1997). *Nature*, **388**(6644), 741–750.