Using Tensor Analysis to characterize Contact-map Dynamics of Proteins

Arvind Ramanathan[*] Pratul K. Agarwal[†]
Christopher J. Langmead[‡]

January 2008
CMU-CS-08-109

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

[*] Joint CMU-Pitt Program in Computational Biology, Carnegie Mellon University, Pittsburgh, PA, USA
[†] Computational Biology Institute, Computer Science and Mathematics Division, Oak Ridge National Lab, Oak Ridge, TN, USA
[‡] School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA
Email: cjl@cs.cmu.edu

Keywords: tensor analysis, multi-way analysis, proteins, molecular dynamics, MD-simulations, online analysis

Abstract

Molecular dynamics simulations provide vast amount of information about a protein's dynamics. To interpret a protein's dynamics and how it may relate to its function, traditionally, two-way analysis techniques such as principal component analysis have been used. However, two-way analysis techniques are usually limited by the fact that they have to be done *post-process*, i.e., after the simulations have been run and also cannot provide insights into temporal behavior of a protein. To overcome these limitations, we are proposing to use multi-way analysis techniques to understand and interpret protein dynamics as and when the simulations are progressing i.e., *online*. We model MD simulations in terms of a collection of *contact maps* and then modeling them as *tensors* to capture multiple dependencies. Using two recently developed techniques to perform online analysis of streaming data, we illustrate the use of this technique to describe and interpret the behavior of a protein complex in real time. We provide both experimental evidence to support our claims and also discuss the potential advantages and disadvantages of using tensor analysis techniques.

1 Introduction

With the proliferation of a number of protein structures in the PDB database [9], efforts are now on to systematically understand the relationship between structure and function [32]. A recent and widely acknowledged belief is that local dynamics (local changes to hydrogen bonds and hydrophobic interactions) drives global dynamics (large scale motions including that of domains) [41] and hence, its function [5]. To fully appreciate a protein's structure-function relationship, it would be essential to understand the intrinsic dynamics of proteins. Proteins, even under equilibrium conditions (constant temperature, pressure and solvent/ chemical conditions) undergo a wide range of motions in varying time-scales. Some of these motions may involve bond-stretching/ vibrations and have a time scale of typically a few femto-seconds, where as other motions, including breathing motions or rearrangements of subdomains may have a time-scale of micro- to milli-seconds. The wide gap in time-scales is often a problem in relating the dynamics of a protein to its function, and hence statistical sampling techniques such as Molecular Dynamics (MD) and/ or Monte-Carlo (MC) simulations are used in understanding the dynamics of a protein and how it may relate to its function.

Molecular dynamics (MD) simulation is perhaps the most widely used technique to understand how a protein functions [32]. In its simplest form, an MD simulation solves the Newton's equations of motions for every atom in a protein and updates their positions at every time step, given an initial configuration of the atoms within a protein [17]. The data from a MD simulation (called MD-trajectory) provides a detailed step-by-step view of a protein's behavior aggregated over the specified number of time-steps for which the simulation was carried out. Given a sufficiently long MD trajectory (assuming that the protein was simulated for a long time), MD can also provide accurate information about time-scales associated with long-range motions [30]. MD can also provide information about solvent motions and how they affect protein function (if solvent was included in the simulations) [4]. Thus, MD can be an extremely useful tool to understand protein dynamics and function. However the data from MD simulations can be quite noisy and not all the data present in the MD trajectory may be related to the protein's functionality. Similarly, visualizing and interpreting the trajectory data in high dimensions presents challenges that are quite difficult to address using current methods. In order to relate protein's dynamics to its function, a number of post-processing tools such as Principal Components Analysis (PCA) [22] and its flavors (Essential Dynamics [7], Quasi-harmonic analysis [23]) need to be employed to interpret and understand the results from MD-simulations.

PCA based techniques for MD construct a covariance matrix from the MD trajectories and decompose them into eigenvalues and eigenvectors [18]. A small number of eigenvalues and eigenvectors (or a linear combination of these) describe the largest motions in the protein during the course of the MD simulation [38]. However, the construction of the covariance matrix renders it time independent; hence, time dependencies associated with functionally relevant motions cannot be inferred from such techniques. Similarly, these techniques can be applied only after the simulation is completed, and hence, no information about functionally relevant motions can be obtained as the simulation is progressing.

A relevant question one may ask is whether any of these analyses can be done as the simulation is progressing. This might be relevant in several circumstances; when one wants to monitor the progress of a simulation towards a certain target (i. e., steering) or in situations where one is tracking the progress of a reaction (like catalysis). Hence, we propose to investigate MD as a streaming application. The idea would be to represent the simulation in the form of a stream of data (like video) and apply algorithms that already exist to analyze streaming data.

In the current work, we use a simple *contact map* representation [13, 20, 29] of a protein to capture how a residue interacts with its local environment. Based on this representation, we model the MD trajectory as a collection of *tensors*. Tensors represent a convenient way to capture multi-way dependencies [33], By employing a variety of *tensor-analysis* techniques [34], we evaluate how one can understand the contact dynamics of a protein. We also present the application of two novel techniques called Dynamic Tensor Analysis (DTA) and Streaming Tensor Analysis (STA) [37] to analyze and monitor MD simulations as and when they are progressing. We apply the method to a fairly large data-set of MD-trajectories to infer information about how a protein behaves during the course of a MD-simulation. Using a simple clustering technique, we extract information about dynamically coupled regions in a protein. We also illustrate how either DTA and STA could be used to monitor MD simulations, and how reconstruction error (used in DTA and STA) may point out events of dynamical interest in the simulation. We also provide a comprehensive analysis of the performance of the two methods mentioned above, and conclude with a possible outlook into how these methods may be applied to other problems in the realm of biology.

2 Modeling MD-simulations as Tensor Streams

2.1 Tensors and Tensor Operations

Tensors represent an abstraction of multiple dependencies that may exist in the underlying data, by succinctly capturing them in multiple dimensions. Formally, a *tensor*, $\mathcal{X}$ of M dimensions can be defined as a multi-dimensional array of real values,

$$\mathcal{X} \in \Re^{N_1 \times N_2 \times \ldots \times N_M} \tag{1}$$

where N_i represents the dimensionality of the i^{th} mode for $(1 \leq i \leq M)$. Thus, a scalar is a tensor of dimension zero, a vector is a tensor of dimension one, and matrix is a two dimensional tensor. All operations possible on a matrix are defined for tensors, however certain key operations do differ in their definition and usage; a review of the tensor operations are provided in Kolda, et al [24].

Any M-dimensional tensor can be converted into a matrix through the *matricization* process. The dimension-i matricizing or *unfolding* of a tensor $\mathcal{X} \in \Re^{N_1 \times N_2 \times \ldots \times N_M}$ is defined as vectors obtained by keeping index i fixed, while varying the other indices. For example, consider $\mathcal{X} \in \Re^{N_1 \times N_2 \times N_3}$. Unfolding the tensor in mode-1 will result in a matrix $\mathbf{X}_{(1)}$ having an order $(N_2 \times N_3) \times N_1$. An illustration of the same is provided in Figure 1.

It is also possible to multiply a tensor with a matrix. This operation is referred to as the *tensor mode product*, and can be defined as follows. The mode product of a tensor $\mathcal{X} \in \Re^{N_1 \times N_2 \times \ldots \times N_M}$

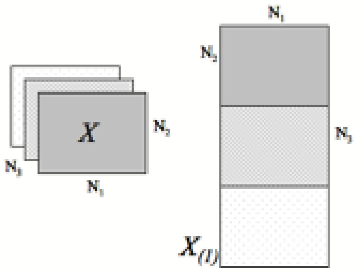


Figure 1: Tensor unfolding into a matrix.

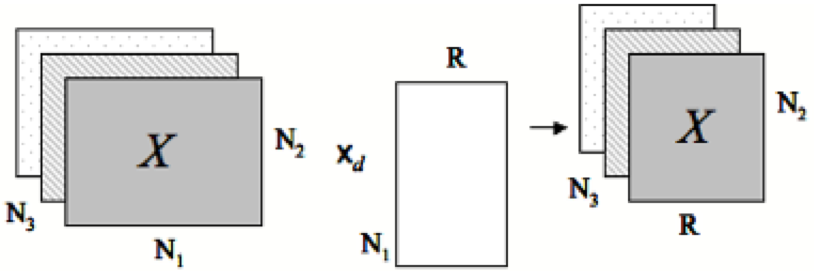


Figure 2: Tensor product illustration.

and a matrix $\mathbf{U} \in \Re^{N_i \times R}$ is a tensor of dimensions $\mathcal{Y} \in \Re^{N_1 \times N_2 \times \ldots \times N_{i-1} \times R \times N_{i+1} \times \ldots \times N_M}$ as:

$$\mathcal{X} \times_d \mathbf{U}(i_1, \ldots, i_{d-1}, j, i_{d+1}, \ldots, i_M) = \sum_{i_d=1}^{N_i} \mathcal{X}(i_1, \ldots, i_{d-1}, j, i_{d+1}, \ldots, i_M)\mathbf{U}(i_d, j) \qquad (2)$$

An example of such an operation is illustrated in Figure 2. Given a sequence of matrices $\mathbf{U}|_{i=1}^{M} \in \Re^{N_i \times R_i}$, then, one may multiply these matrices in sequence shown below:

$$\mathcal{X} \prod_{i=1}^{M} \times_i \mathbf{U}_i = \mathcal{X} \times_1 \mathbf{U}_1 \times_2 \ldots \times_M \mathbf{U}_M \qquad (3)$$

2.2 Representing Protein Structures as Tensors

Extending this representation to proteins, the most convenient way to model a protein as a tensor, would be to use a simple *contact-map* representation. In this representation, we consider the number of heavy atoms of residue i in *contact* with residue j in a particular MD snapshot. In this context *contact* means whether any of the heavy atoms in residue i comes within 4 Å of the heavy atoms in residue j. The choice of 4 Å can be even parameterized depending on whether we mean van der Waals contact between two hydrophobic atoms or between a hydrogen bond donor and acceptor atom. For the current study, a uniform distance cut-off of 4 Å was used. Based on this,

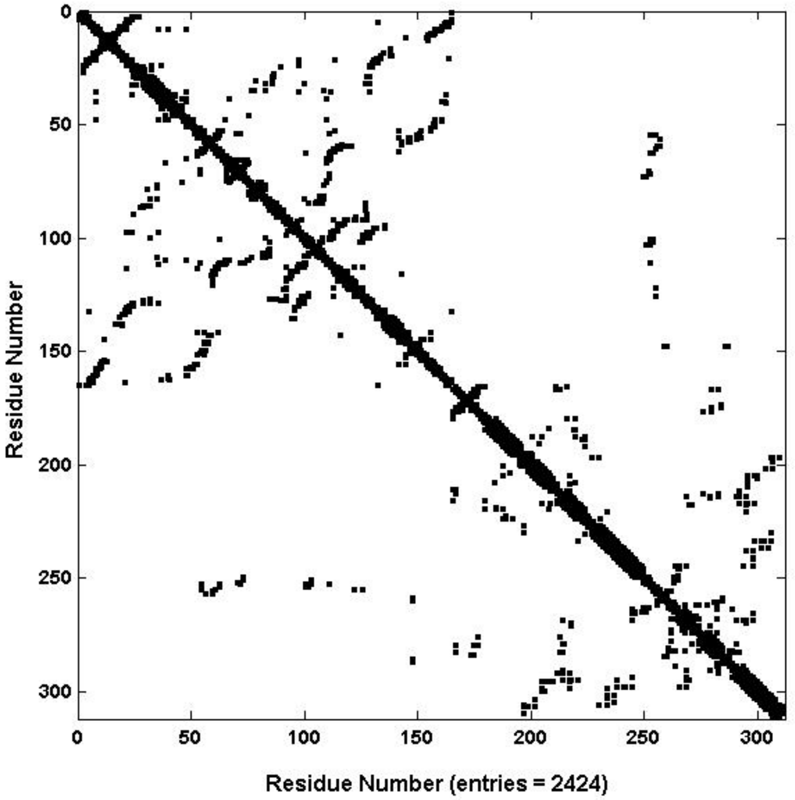


Figure 3: A contact map representation for a protein structure. The diagonal elements of the plot are set to zero, whereas immediate neighbors and the neighbors with which the protein maintains *contact* are set to a value that is proportional to the number of heavy atoms in contact with that residue as described in Equation4.

one can model the contact matrix $\mathbf{A}_{ij}(t)$ as follows:

$$\mathbf{A}_{ij}(t) = \frac{n_{ij}}{\sqrt{n_i.n_j}} \tag{4}$$

where $\mathbf{A}_{ij}(t)$ is the contact matrix. n_{ij} is the number of heavy atoms in residue i coming in contact with the heavy atoms in residue j. This value is normalized over the square root of the number of heavy atoms in residues i and j. This value captures a measure of the localized *density* of interaction within an amino acid's immediate neighborhood. This representation of a contact map has been used in several previous works including [14, 12]. For the purposes of clarity, we choose to ignore self contacts (all heavy atoms of residue i are in contact with it self) and hence the diagonal elements of **A** were set to zero. Since $\mathbf{A}(t)$ is also a second order tensor, we will use the standard tensor notation, $\mathcal{A}_t$ to represent this data. A simple contact plot for one of the proteins is shown in Figure 3.

At every time step t, MD updates the positions of the atoms and thus, a new instantaneous

$\mathcal{A}_t$ arrives. Although there is not a very drastic change between $\mathcal{A}_t$ and $\mathcal{A}_{t-1}$, its immediate predecessor, any change or reorganization of the contacts will be captured in this representation. In effect, we are able to capture the residue's interactions by tracking the instantaneous change in the localized density of a residue in a protein.

One can imagine the whole MD trajectory to be a collection of these second order tensors in series (and thus as a third order tensor). Such an ordered sequence of tensors is usually referred to as *tensor stream*. Formally, a tensor stream is defined as a sequence of M^{th} order tensors $\mathcal{X}_1, \mathcal{X}_2, \ldots \mathcal{X}_n$, where each of the $\mathcal{X}_i \in \Re^{N_1 \times N_2 \times \ldots \times N_M}$ and n is an integer that increases with time. For our particular application, we model the contact map tensor as a stream of tensors:

$$\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n \tag{5}$$

where $n = T/w$ represents the number of tensor windows in the MD simulation. T is the total number of steps (or time) for which the MD simulation was carried out, and w is the size of the window of snapshots involved in the analysis. $\mathcal{A}_i$ thus represents a slice of the MD trajectory which may involve w snapshots, where w represents a window in the MD trajectory. The size of the window w can be varied depending on the length of the MD trajectory.

2.3 Tensor Analysis for MD-simulations

Given the description of the tensor stream defined in Equation (5), the main objective of our analysis would be to find underlying patterns about how the contacts between a residue and its environment are changing with respect to each other (i.e., relative behavior) as well as with respect to time (i.e. temporal behavior). By studying the variation in contact maps, we are able to extract a global description about movements within residues. We term this description of the patterns observed in contact maps over the course of the simulation as *contact map dynamics*.

In order to extract patterns underlying the original data, it is essential to describe the system in terms of reduced dimensions using a procedure like Singular Value Decomposition (SVD) or Principal Component Analysis (PCA), by minimizing the observed variance in the underlying data. SVD or PCA are applicable to two-dimensional problems such as analysis of covariance matrices from MD simulations. Let us now see how PCA will be applicable in this scenario. One may construct a covariance matrix based on the observed changes in contact maps over time, as defined:

$$cov(\mathbf{A}_{ij}) = \langle a_{ij} - \langle a_{ij} \rangle \rangle \tag{6}$$

where a_{ij} is the instantaneous contact value, defined in Equation 4 and $\langle . \rangle$ defines the average value over the entire simulation. Since $\mathbf{A}_{ij}$ is a symmetric positive semi-definite matrix, PCA would yield exactly N eigenvalues (Λ) and eigenvectors (U_1^{PCA}), which may be concisely represented as follows:

$$cov(\mathbf{A}_{ij}) = \mathbf{U}_1^{PCA} \Lambda \mathbf{U}_1^{T,PCA} \tag{7}$$

For tensors, patterns are extracted by a similar procedure called *tensor analysis* which is an extension of SVD into multiple dimensions. Thus, the objective function would be to minimize the variance observed in the tensors, across *every chosen dimension*. Formally, given a sequence of tensors $\mathcal{J}_1, \mathcal{J}_2, \ldots, \mathcal{J}_n$, where each $\mathcal{J}_i \in \Re^{N_1 \times N_2 \times \ldots \times N_M} (1 \leq i \leq M)$, we would like to find orthogonal matrices $\mathbf{U_i} \in \Re^{\mathbf{N_i} \times \mathbf{R_i}}$, one per dimension, such that the least-squared deviation e is minimized. The reconstruction error e is defined by:

$$e = \sum_{t=0}^{n} \| \mathcal{J}_t - \mathcal{J}_t \prod_{i=1}^{M} \times_i (\mathbf{U_i U_i^T}) \|_{\mathbf{F}}^{\mathbf{2}} \tag{8}$$

The operation $\mathcal{J}_t \prod_{i=1}^{M} \times_i (\mathbf{U_i U_i^T})$ is the approximation of $\mathcal{J}_t$ under the space spanned by $\mathbf{U_i}|_{\mathbf{i=1}}^{\mathbf{M}}$. This approximation in multiple dimensions is termed *Frobenius-norm* and can be formally defined as the sum of the squared elements of the tensor $\mathcal{J}$, shown below:

$$\|\mathcal{J}\|_F^2 = \sum_{i_1=1}^{N_1} \ldots \sum_{i_M=1}^{N_M} \mathcal{J}(i_1, i_2, \ldots, \imath_M)^2 \tag{9}$$

In order to construct the least-squared approximation in multiple dimensions, one can typically formulate it as minimizing the least-squared approximation across every pair of M dimensions separately. Thus, if we have M dimensions, one way of minimizing the least squared deviation is to construct individual covariance matrices across two different dimensions and minimize using the PCA technique in every one of the dimensions. Formally, this may be written down as:

$$cov(\mathbf{J}_{(1)}) \quad = \mathbf{U}_1 \Lambda_{(1)} \mathbf{U}_1^T \tag{10}$$

$$cov(\mathbf{J}_{(2)}) \quad = \mathbf{U}_2 \Lambda_{(2)} \mathbf{U}_2^T \tag{11}$$

$$cov(\mathbf{J}_{(M)}) \quad = \mathbf{U}_M \Lambda_{(M)} \mathbf{U}_M^T \tag{12}$$

In the case of simulations, one can substitute $\mathcal{A}$ in Equation (8) and minimize the deviation observed across the variation in contacts as well as time. There will be three core matrices that one will obtain since the whole MD trajectory is modeled as a third order tensor. By our representation, since $\mathcal{A}_t$ is symmetric, $\mathbf{U}_1$ and $\mathbf{U}_2$ orthogonal matrices will be the same. Orthogonal matrix $\mathbf{U}_3$ which represents the time dependent variation in the rearrangements will be dependent on the window size n that we previously defined.

In order to track simulations *on-the-fly*, it is essential that Equation(8) is performed as the simulations are progressing. Hence, tracking the variance in each one of the dimensions shown in Equations (10-12), need to be *incremental*. This is done by using two algorithms devised by Sun, et al [34]. These two algorithms called Dynamic Tensor Analysis (DTA) and Streaming Tensor Analysis (STA) provide an intuitive yet efficient way to update the variance observed in the tensors incrementally. The details of these algorithms can be obtained from [34, 35, 36, 37]. Here we provide a qualitative description of the two algorithms used.

The DTA procedure is illustrated in Figure 4. One can observe that it is possible to update the variance matrices (in a particular dimension d, represented by $\mathbf{C_d}$) incrementally, without having to store any prior information about previous tensors. Implicitly, every tensor stream arriving at time t is independent of the tensor at time $t-1$. Once the variance matrices are available, an SVD on the variance matrix is easy to compute using the relation $\mathbf{C}_d = \mathbf{U}_d \Lambda_d \mathbf{U}_d^T$, where $\mathbf{U_d}$ is an orthogonal matrix (eigenvectors) and $\mathbf{S_d}$ is a diagonal matrix (eigenvalues). Thus, the steps in the

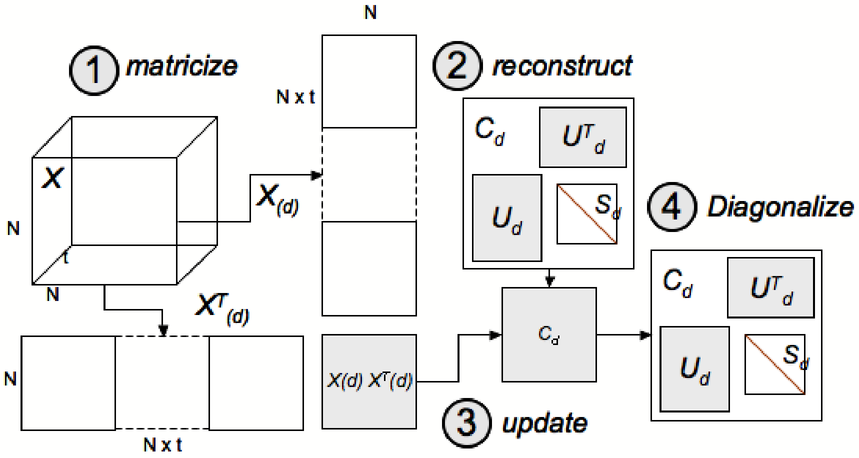


Figure 4: Illustration for the Dynamic Tensor Analysis Algorithm.

algorithm would entail the matricizing the new tensor $\mathcal{J}$ in the d^{th} mode, updating the variance matrix $\mathbf{C_d}$ with $\mathbf{C_d} \leftarrow \mathbf{C_d} + \mathbf{J^T_{(d)} J_{(d)}}$, and then computing the new projection matrices $\mathbf{U_d}$ by performing an SVD on the $\mathbf{C_d}$. This was illustrated in the previous discussion (refer Equations 10-12).

The most time-consuming step in DTA is the SVD that needs to be performed on the new variance matrix $\mathbf{C_d}$. STA is an efficient approximation of DTA without the necessity of this step. The schematic for STA is illustrated in Figure 5. Instead of diagonalizing the $\mathbf{C_d}$, it is also possible to track the changes in the variance matrices, by estimating the change in the projection matrix $\mathbf{U_d}$ and then updating $\mathbf{U}$ on the basis of the error e observed. This update is done by sampling a certain number of columns from $\mathbf{U}$ to estimate the change in error e that is observed and can be controlled by the user. Note that this is only a fast approximation to DTA. The technical details of both these algorithms and their complexity is discussed in detail in [35].

2.4 Related Work

The use of multi-way analysis has been particularly popular within the field of cheminformatics [33]. Although several of these applications deal mostly with data from experiments such as fluorescence and infra-red spectroscopy, only recently has it been applied for NMR [27, 28]. Multi-way analysis techniques have also been popular in other bioinformatics applications such as three-way analysis of micro-array data [42] and also in several medical applications [1, 2]. However, this is the first time, that multi-way analysis techniques to systematically analyze protein structures via MD-simulations. Recently, a comprehensive survey of multi-way analysis techniques has been provided in [24].

On the fly analysis of multiple data streams has only been a recent development. Network attack monitoring has been successfully performed by using DTA and STA [35, 36]. Also, the techniques presented above have been used to perform multi-way latent semantic indexing on several large data-sets including DBLP, to mine and visualize information in multiple dimensions

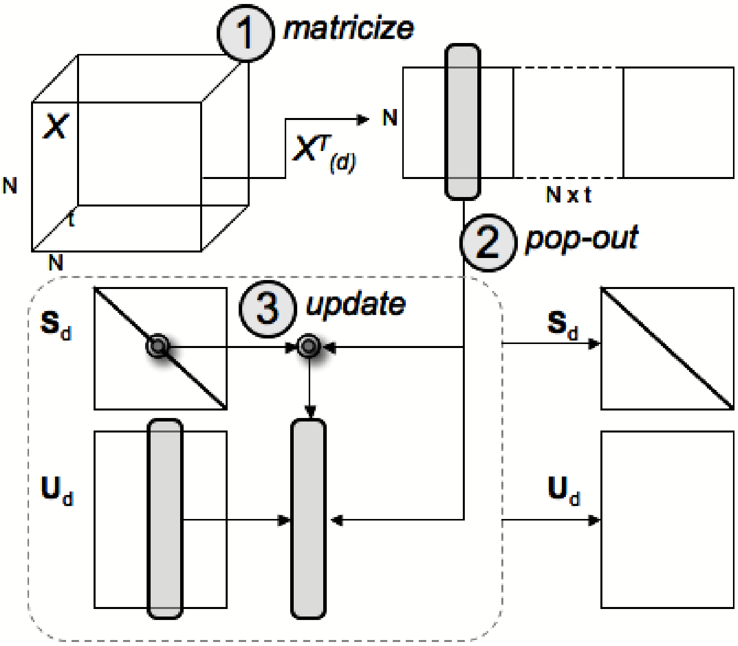


Figure 5: Illustration for the Streaming Tensor Analysis Algorithm.

simultaneously [37, 34].

A vast amount of literature exists on using contact map representation for proteins [13, 20, 29] and their use to model protein dynamics [19, 8, 31, 15]. Domany et al modeled protein folding in the contact map space using Monte Carlo simulations [8, 39]. Similarly a simple lattice based model was used to simulate protein folding in [19] as well as understand the kinetics behind a protein folding process in [31]. However, the process of using contact maps to track and monitor MD simulations has not been attempted before, and also the study of the evolution of contact maps to identify which parts of the protein are more susceptible to change (in terms of their dynamics) is also unique in the current work.

3 Results

3.1 MD and Data preparation

Our data set of protein simulations comprised of Cyclophilin A (CypA), a protein that catalyzes the peptidyl-prolyl cis-trans isomerization reaction [16, 10, 6, 3]. CypA is a protein formed by a β-barrel flanked by two α-helices on either side of the barrel, and a large number of functionally important loop structures. CA^N on the other hand, is a protein that is almost exclusively made up of α-helices with a large functional loop which binds to CypA. In this data set of MD simulations, we have basically studied the progress of the entire cis-trans isomerization using a simulation protocol outlined in [6]. The CypA structure used for this analysis had an entire protein (namely the HIV-1 capsid protein; CA^N). We first processed the MD trajectories to compute the contact map (Equation 4) at every time step. Since there is a large amount of data within the trajectory, we chose to use a smaller subset of the structures that spanned the entire reaction pathway as our input contact matrices. Each MD trajectory consisted of 18000 structures. Using a regular increment of 50 snapshots, we constructed a 3^{rd} order tensor consisting of T = 3600 structures, and chose a window size w of 10. Thus, for our case, n worked out to be 360. We ran both DTA and STA on the chosen snapshots and analyzed the results.

3.2 Analysis of CypA-CA^N dynamics with DTA

In this subsection, we will illustrate how the outputs from DTA can help one understand the contact dynamics of a protein such as CypA complexed with CA^N. The protein, as explained above provides an ideal platform for study since it involves the dynamics between two entirely different proteins with significant structural differences. First we examine the core matrices from DTA and illustrate how one may interpret information about the dynamics of CypA-CA^N complex. Then, we apply clustering techniques like k-means to interpret the orthogonal matrices. Also, we analyze the nature of dynamical events by tracking the reconstruction error.

3.2.1 Core matrices from DTA indicate differences in contact dynamics of CypA and CA^N

The core matrices $\mathbf{U_1}$ or $\mathbf{U_2}$ (which are identical in this case) represent a compact description of how contacts between residues evolved over the course of the entire simulation. One can visualize this as a summary of *contact dynamics*, which may give insights into how the protein behaves as a whole. First, we illustrate the core matrices $\mathbf{U_1}$ shown as a color map in Figure 6. As such, the plot (Figure 6) represents the correlation in contacts observed from the simulation. However, the column maximum values of the core matrix $\mathbf{U_1}$ indicate those residues that may play an important role in terms of the contact dynamics, a subsequent plot is illustrated in Figure 7. Observe that within the plot, there are several regions of CypA (1-165 on the x-axis) that stand out, particularly (a) 26-56, (b) 90-102 and (c) 140-150. Note that (a) is the location of the active site of CypA (52-56) and several of the residues in this region are conserved across species (ex. 30, 32, 35, 36 and 48). Similarly, within (b), a number of residues are also conserved: 90, 98, 100 and 102. A similar observation can also be made for CA^N.

It is also interesting to note that several of the residues that are implicated in CypA and CA^N function occupy the extremum points on the plot. To illustrate this, we plotted the mean and standard deviations for the column maxima in $\mathbf{U_1}$ for CypA and CA^N separately. An immediate observation that follows from this plot is that the CypA structure exhibits potentially more changes within its contact matrices compared to that of the CA^N. This is evident because (a) the mean for CypA's $\mathbf{U_1}$ is larger than that of CA^N and (b) the variance/ spread in CypA is also larger than that of CA^N. As is evident from experimental evidence the B-factors for CypA is also larger than that of CA^N, providing further support to the claim that CypA exhibits more changes in its contact dynamics. Another relevant experimental evidence might be also explained on the basis of the secondary structure content of the protein. CypA is clearly formed of a β barrel and with a number of loop regions than CA^N which is largely made up of α helices. The residues beyond the first

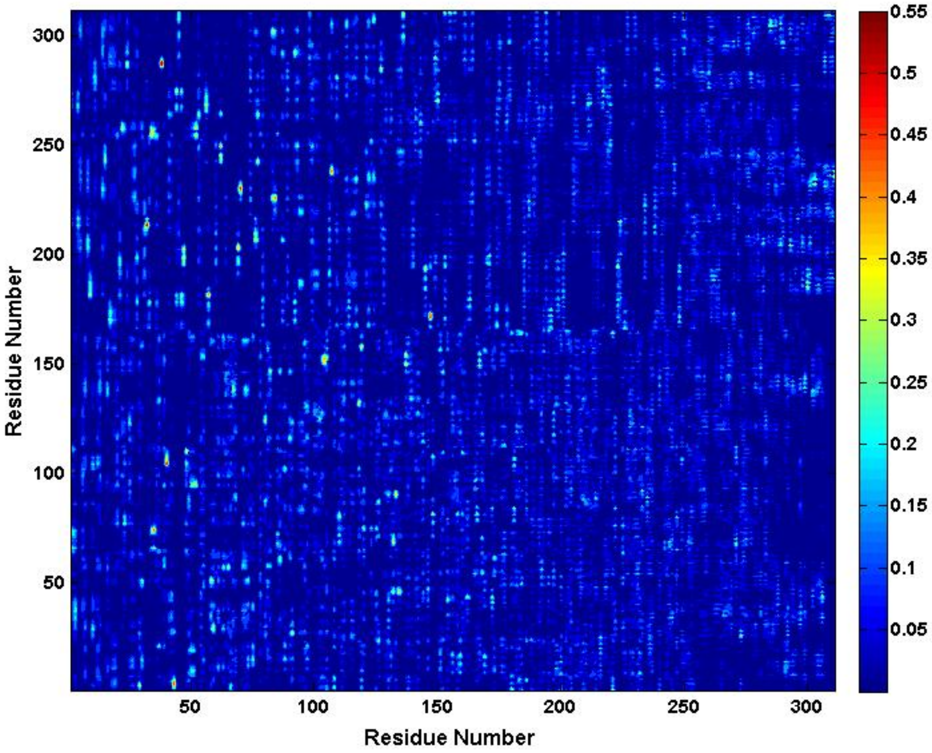


Figure 6: Absolute values of the core matrix $\mathbf{U_1}$ shown as a color map. The columns in $\mathbf{U_1}$ with the highest values represent important residues.

standard deviation interval (both minima and maxima) on the plot are typically conserved across species (in CypA) and are involved in forming important contacts with the neighboring residues.

3.2.2 Clustering the core matrices from DTA can provide insights into *contact dynamics* of CypA-CA^N complex

In order to meaningfully interpret the correlations from the core matrices, we chose to cluster the core matrices using a simple clustering algorithm such as k-means. The output from the k-means clustering for CypA-CA^N complex is illustrated in Figure 8. One of the interesting observations was that at $k = 2$, we were able to observe that the CypA separated from the CA^N protein, indicating that the contact dynamics of the substrate (CA^N) and the enzyme (CypA) were quite different. Only those parts of the substrate in intimate contact with the enzyme seem to exhibit similar or co-ordinated contact dynamics with CypA. This is especially encouraging, since several previous studies have also indicated such a result.

It is interesting to note that the clustering not only gives an idea about the spatial locations of the regions in the protein exhibiting similar contact dynamics, but also gives insights into the relative difference in contact dynamics of CypA and CA^N. As already illustrated before from

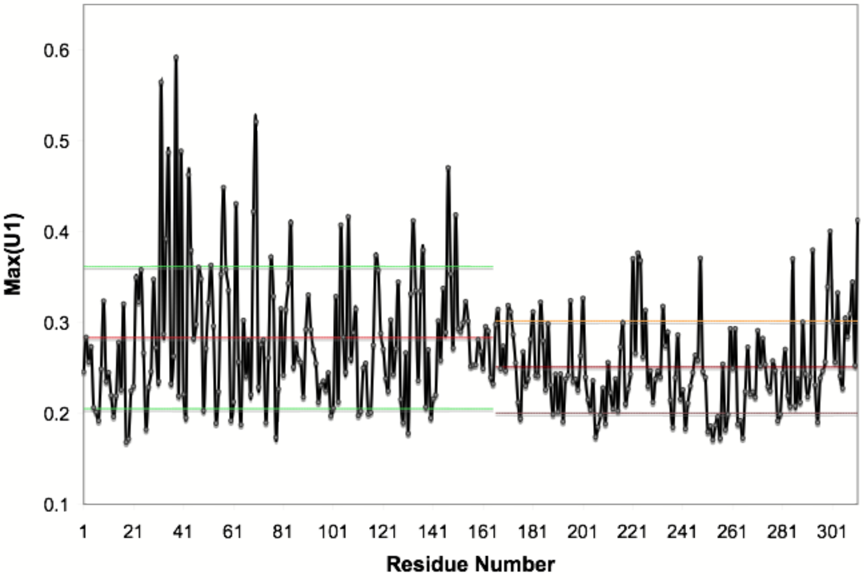
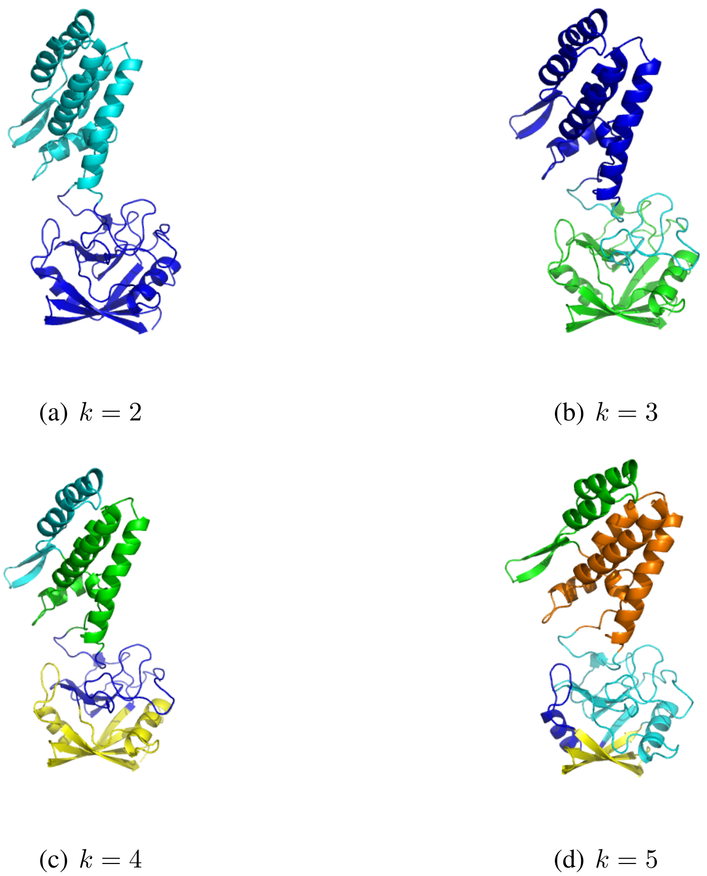


Figure 7: Maximum fluctuations in terms of contacts within cyclophilin A

Figure 6, CA^N is relatively *stable* in terms of its contacts, meaning that residues in CA^N do not change their contacts dynamically as much as CypA does. However, when we observe the clusters from $k = 3$, as shown in Figure 8(b) it is clear that the clusters from CypA have now separated into two clusters; one cluster showing loops from CypA that are in contact with the substrate (CA^N) and the other cluster showing the rest of CypA (shown in green). This separation is further clear when $k = 4$, as shown in Figure 8(c). The entire β-sheet proximal to the substrate and the loop structures involved in the catalytic process [3] are separated into a cluster (shown in blue) with the rest of the CypA structure forming the hydrophobic core of the protein (the two α-helices and the rest of the β-barrel) grouped into another cluster (shown in yellow). The substrate is also clustered into two regions. Of these two regions, the cluster shown is cyan is also dynamic, as evident from the B-factors determined from X-ray crystallography [21]. We have also illustrated $k = 5$, Figure 8(d) which shows the further division in the hydrophobic core of the protein by separating the three β-strands as well as the α-helix located in proximity of the active site of the protein.

A clearer picture of the relative contact dynamics between CypA and CA^N can be obtained by looking at the actual values of $\mathbf{U_1}$. Here we illustrate this for $k = 4$, shown in Figure 8(c). Note that the k-means clustering uses an arbitrary numbering scheme to illustrate its clusters and hence in order to understand the meaning of the clusters, one has to actually look at the values of $\mathbf{U_1}$. The clusters shown in green and cyan exhibit lower values on average compared to the rest of the structure. The cluster shown in yellow has relatively the lowest average values of $\mathbf{U_1}$, compared to the rest of CypA. However, the cluster shown in blue has the highest values in the core matrix. Note that this cluster is formed of residues in CypA that form the active site of the protein and/ or have residues implicated in the catalytic activity of the protein. These residues are all part of the three regions that we pointed out earlier in section 3.2.1. Also the residues from CA^N that



(a) $k = 2$

(b) $k = 3$

(c) $k = 4$

(d) $k = 5$

Figure 8: k-means clustering of the core matrix $\mathbf{U_1}$ from DTA mapped onto the structure of CypA-CA^N complex (1AWQ). The colors are arbitrarily selected via the k-means implementation in MATLAB. The significance of the clustering is discussed in the text.

are in contact with the enzyme also form a part of this cluster, further indicating that the contact dynamics (both spatial and temporal aspects) are similar.

Note also that there is a distinct hierarchy within the clustering observed. This hierarchy is illustrated in Figure 9. The first level/ topmost level of clustering divides the substrate (CA^N) and enzyme (CypA) into two separate clusters (except for the regions from the substrate in contact with the enzyme). The second level divides the regions of CypA in contact with CA^N from the rest of CypA, however, the substrate remains unaffected. With k = 4, one can further observe that CA^N is further divided into two clusters; and with k = 5, CypA is further divided into one more cluster, as shown in 8. None of this information was provided as either as prior knowledge or as information to DTA for processing. Thus, DTA *learns* these clusters automatically, from the MD simulations (see discussion).

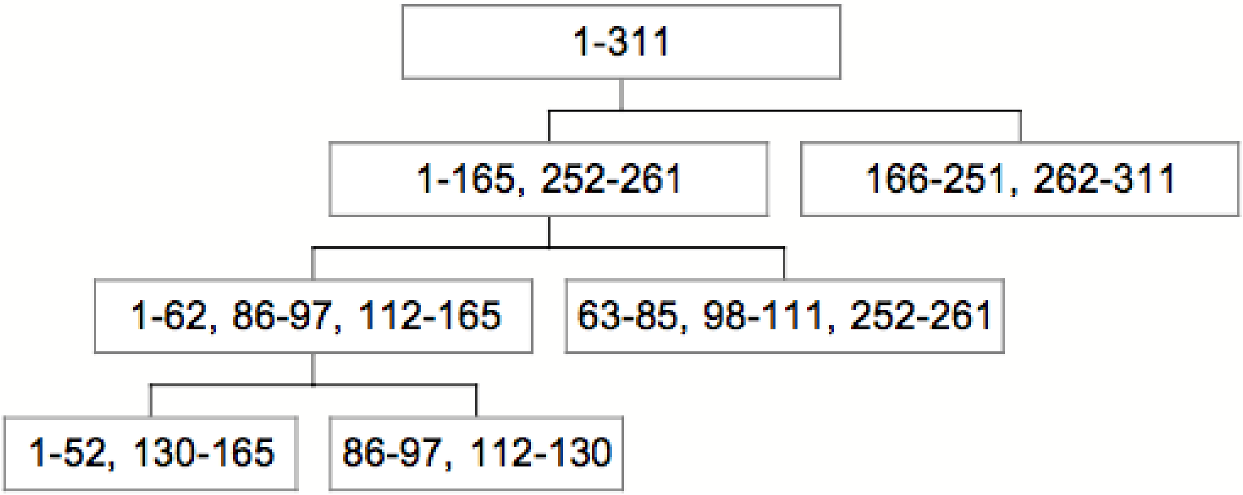


Figure 9: Cluster hierarchy from DTA for CypA-CA^N complex.The leaves at the bottom level illustrate clusters obtained from $k = 5$ and the levels higher illustrate clustering at $k = 2, 3$ and 4 respectively.

3.2.3 Tracking MD simulations using DTA

One of the interesting aspects of using DTA to analyze streaming data is that it can identify time-points where there was significant deviation from the previous tensor. This ability has been especially useful in identifying and tracking external attacks in a network monitoring application setting [35]. For MD simulations, we analyze here the reconstruction error (e) metric to track bio-molecular simulations. The reconstruction error e plotted against the tensor number (time) is shown in Figure 10. As seen from the plot, the errors are not that high, however, there are several points of interest across the entire trajectory where the contact maps have significantly deviated (beyond the second standard deviation interval plotted in dark red) from the mean reconstruction error. These time points indicate an event where *the contact map significantly varied from the original structure*. For example, in the plot shown below, the structures from tensor 14 (snapshots 700-750) were interesting, since they show significant deviation in the arrangement of the substrate (CA^N) within the active site location of CypA. Similar observations can be made with respect to the structures from tensor 59 (snapshots 2950-3000), tensor 86 (snapshots 4300-4400), tensor 202 (10100-10150) and many others. Note that of these tensors, tensor 202 is quite interesting because it represents structures that are close to the transition state of the enzymatic reaction.

3.3 Analysis of CypA-CA^N dynamics with STA

We investigated the applicability of STA to CypA-CA^N dynamics by analyzing the core matrices from STA, and also looking at the clustering of the core matrices. We also test the use of STA by

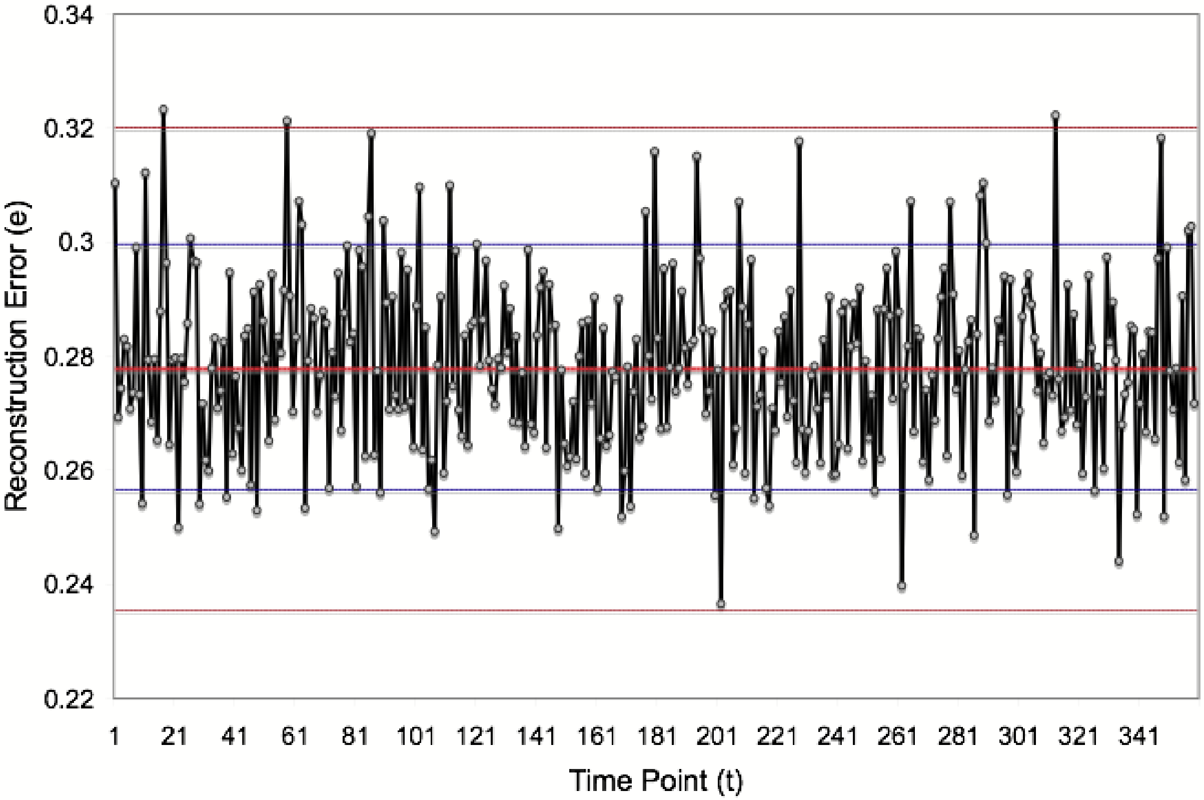
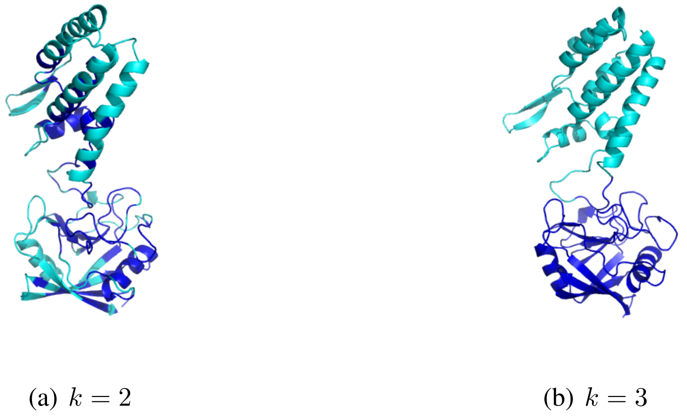


Figure 10: The reconstruction error plotted as a function of the tensor snapshot from the MD simulation. Every tensor snapshot is a representation of 50 snapshots from the MD trajectory.

varying the sample from the current tensor to provide an estimate of the best results that can be compared with DTA. We varied the sampling parameter for STA from a minimum of 10% to 80% in increments of 10% to understand the implications of sampling and its impact on predicting the dynamic domains in CypA-CA^N complex. One of the very first observations from STA was that the clustering was more diffuse compared to that of the DTA. This meant that a number of residues that would have otherwise belonged to CypA were clustered to belong to CA^N and vice-versa, as shown in Figure 11(a). However, the maximum length of these clusters that were misclassified were about 8 in each segment, and hence we used a cluster refining method where by we assigned the smaller clusters into their respective domains. This led to a good improvement in the clustering as illustrated in Figure 11(b). However, STA does not seem to perform as well as DTA when we increase the cluster size to $k = 2, 3, 4$ and so on.


(a) $k = 2$

(b) $k = 3$

Figure 11: k-means clustering of the core matrix $\mathbf{U_1}$ from STA mapped onto the structure of CypA-CA^N complex (1AWQ). The colors are arbitrarily selected via the k-means implementation in MATLAB. The significance of the clustering is discussed in the text.

4 Discussion

4.1 Biological significance of DTA in interpreting contact dynamics

The contact matrix representation is a measure of how much each residue moves with respect to its local environment. DTA successfully captures the spatial and temporal behavior of residues with respect to their respective environments, and can learn how the enzyme and substrate are coupled. It cannot (and does not) however capture any physical correlations with respect to the dynamics of a protein, as one would expect from a correlation matrix observed from MD simulations. It can clearly identify regions coupled with respect to *contacts*, meaning, residues that are moving similarly in their environment are typically identified to belong to a cluster. This is especially clear, if one observes the clustering with $k = 2$,from DTA (Figure 8(a)). Clearly DTA identifies the enzyme and substrate to be in their respective clusters, whereas the loop regions within CA^N in contact with CypA are identified to be coupled with respect to the CypA cluster.

The residues at the interface of two clusters, i.e., cluster boundaries indicate residues where there is a significant deviation from the current cluster in terms of its contact dynamics. These residues and/ or its neighbors ones may play an important role in modulating a protein's function. Note that within CypA-CA^N complex, His252 (His87 in CA^N) and Gln260 (Gln95 in CA^N) are both identified to be making contacts with CypA, especially with residues 25-132. This sequence of residues in CypA contain most of the functionally implicated (and highly conserved) residues, which are known to be important to modulate the protein's function. It is also interesting to note that the rest of the enzyme (CypA) is clustered into a separate group, where conservation of sequence is quite variable, indicating that the role of this cluster may be only to stabilize the hydrophobic core of the protein.

As noted in the results section, DTA can extract the relative time scales of contact map dynamics. We had also shown that the rate at which contacts evolve within CypA are faster than that of

CA^N. This can be further examined with respect to the secondary structure content of the proteins in the complex. Note that CA^N is a compact, α-helical protein, with just two loop structures, one from the N-terminal part of the protein (166-182) and the part of the protein that makes contact with the enzyme (248-266). On the other hand, CypA is composed of a β-barrel and two α-helices flanking the hydrophobic core of the protein on either side. This gives rise to an interesting consequence observed from Figure 7: since the values of the core matrix $\mathbf{U_1}$ is lower for CA^N than in CypA, the change in contact maps occurs slower and is much lesser in CA^N than in CypA. If we assume that a contact made by a residue with its neighboring environment is a measure of its communicability, then CypA (and the loop regions of CA^N) have a higher communicability than CA^N. Thus, a protein with more β-strands represents its ability to transfer more information across regions in a protein than a more compact α-helical protein like CA^N. This also corresponds well with the idea about secondary structural elements, since in an α-helical protein the side-chain atoms are more constrained by their locality to move, than in a β-sheet/ barrel protein, where the packing of the atoms allows for freer movements between side-chain atoms [11].

One may also want to interpret the contact map dynamics in terms of how various residues are tethered to their respective environments. One may classify residues as *contact tethered*: residues that are relatively constrained by their locality (as seen in CA^N) and *contact untethered*: residues that are relatively easy to move in their local environment. This classification of residues allows one to interpret some more observations from the dynamics. Since on an average, CypA undergoes more contact rearrangements than CA^N, CA^N may be more rigid than CypA. This structural rigidity of the substrate may also be a biologically required aspect of enzyme catalyzed reactions, where by the placement of the substrate in the enzyme may control the reaction. CypA on the other hand exhibits potential flexibility to accommodate the substrate, while undergoing conformational changes that allow it to catalyze the reaction. However, the overall structure of CypA remains more or less the same, except for the regions in the protein implicated in catalysis [10, 6].

4.2 DTA as an unsupervised learning algorithm for MD-simulations

DTA is ideally suited for providing both an *analysis* tool as well as a *tracking* tool for MD-simulations. Clearly, by using the core matrices from DTA, one can understand physically important details with respect to protein dynamics as well as make reasonable predictions about the system regarding how contact dynamics evolve over time. This provides the incentive of using DTA as an unsupervised learning method for MD-simualtions. Part of the difficulty in interpreting the data from an MD-simulation lies in the fact that it is high-dimensional as well as noisy. But, DTA can overcome this obstacle and provide reasonable insights into protein dynamics while being independent of any parameters that the user has to pre-define.

DTA also corresponds well with other existing methods to analyze protein structure. Especially interesting to note is the similarity between DTA and the domain-partitioning algorithm via spectral clustering (SCK) [26]. The analysis of the core matrices in DTA correspond well with the domains that are identified by SCK, which is especially encouraging since both these methods use k-means clustering to partition the graph into domains. Similar observations can also be made with respect to other domain partitioning algorithms such as [25, 40]. The advantage of using DTA however is that it clearly captures a spatio-temporal dependency in how contacts evolve with respect to different parts of the protein, providing a clear insight into the various interactions and how they may be affecting its overall function.

5 Conclusion

We have outlined the development of a novel representation of protein dynamics using contact maps, and analyzed the same using tensor analysis techniques to capture specific spatial dependencies as well as temporal evolution. The use of DTA to capture these dependencies can be useful in a wide variety of applications including steered molecular dynamics as well as tracking applications real-time in the several experimental techniques including multi-dimensional NMR.

References

[1] Evrim Acar, Canan Aykut-Bingol, Haluk Bingol, Rasmus Bro, and Bulent Yener. Multiway analysis of epilepsy tensors. *Bioinformatics*, 23(13):i10–18, 2007.

[2] Evrim Acar, Canan Aykut Bingol, Haluk Bingol, Rasmus Bro, and Bulent Yener. Seizure recognition on epilepsy feature tensor. In *Annual International Conference of the IEEE Engineering in Medicine and Biology*, 2007.

[3] P. K. Agarwal. Protein dynamics and enzymatic catalysis: Investigating the peptidyl-prolyl cis-trans isomerization activity of cyclophilin a. *Proteins: Struct., Funct., Bioinformatics*, 56(3):449–463, 2004.

[4] P. K. Agarwal. Role of protein dynamics in reaction rate enhancement by enzymes. *J. Amer. Chem. Soc.*, 127:15248–15256, 2005.

[5] P. K. Agarwal. Enzymes: An integrated view of structure, dynamics and function. *Microbial Cell Factories*, 5, 2006.

[6] P. K. Agarwal, A. Geist, and A. Gorin. Protein dynamics and enzymatic catalysis: Investigating the peptidyl-prolyl cis-trans isomerization activity of cyclophilin a. *Biochemistry*, 43(33):10605–10618, 2004.

[7] A. Amadei, A. B. M. Lissen, and H. J. C. Berendsen. Essential dynamics of proteins. *Proteins: Struct., Funct., Genet.*, 17:412–425, 1993.

[8] G. Barkema, P. Grassberger, and W. Nadlers, editors. *Protein folding in the space of contact maps*. Proceedings of the HLRZ Workshop on Monte Carlo Approach to Biopolymers and Protein Folding, World Scientific, 1998.

[9] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E Bourne. The protein data bank. *Nucleic Acids Research*, 28:235–242, 2002.

[10] Daryl A. Bosco, Elan Z. Eisenmesser, Susan Pochapsky, Wesley I. Sundquist, and Dorothee Kern. Catalysis of cis/trans isomerization in native hiv-1 capsid by human cyclophilin a. *Proc. Natl. Acad. Sci. USA*, 99(8):5247–5252, 2002.

[11] Guillaume Bouvignies, Pau Bernado, Sebastian Meier, Kyuil Cho, Stephan Grzesiek, Rafael Bruschweiler, and Martin Blackledge. Identification of slow correlated motions in proteins using residual dipolar and hydrogen-bond scalar couplings. *Proc. Nat. Acad. Sci. U. S. A.*, 102(39):13885–13890, 2005.

[12] K. V. Brinda and Saraswathi Vishveshwara. A network representation of protein structures: Implications for protein stability. *Biophys. J.*, 89:4159–4170, 2005.

[13] H.S. Chan and K.A. Dill. Origins of structure in globular proteins. *Proc. Natl. Acad. Sci. USA*, 87:6388–6392, 1990.

[14] C. Chennubhotla and I. Bahar. Signal propagation in proteins and relation to equilibrium fluctuations. *PLoS Comput Biol*, 3(9):e172, 2007.

[15] G. Crippen and T. F. Havel. *Distance Geometry and Molecular Conformation*. Wiley, 1988.

[16] E. Z. Eisenmesser, D. A. Bosco, M. Akke, and D. Kern. Enzyme dynamics during catalysis. *Science*, 295(5559):1520–1523, 2002.

[17] Daan Frenkel and Berend Smit. *Understanding Molecular Simulation: From Algorithms to Applications*. Computational Science Series. Academic Press, San Diego, 2002.

[18] S. Hayward and N. Go. Collective variable description of native protein dynamics. *Annual Review of Physical Chemistry*, 46(1):223–250, 1995.

[19] D. A. Hinds and M. Levitt. Exploring conformational space with simple lattice model of protein structures. *J. Mol. Biol.*, 243:668–682, 1994.

[20] L. Holm and C. Sander. Protein structure comparison by alignment of distance matrices. *J. Mol. Biol.*, 233(1):123–138, 1993.

[21] B. R. Howard, F. E. Vajdos, S. Li, W. I. Sundquist, and C. P. Hill. Structural insights into the catalytic mechanism of cyclophilin a. *Nat Struct Biol*, 10:475–481, 2003.

[22] Ivan T. Jolliffe. *Principal Component Analysis*. Springer, 2002.

[23] Martin Karplus and Joseph N. Kushick. Method for estimating the configurational entropy of macromolecules. *Macromolecules*, 14(2):325–332, 1981.

[24] Tamara G. Kolda and Brett W. Bader. Tensor decompositions and applications. Technical report, Sandia National Laboratories, 2007.

[25] Sibsankar Kundu, Dan C. Sorensen, and Jr. George N. Phillips. Automatic domain decomposition of proteins by a gaussian network model. *Proteins: Struct., Funct., Bioinformatics*, 57:725–733, 2004.

[26] Steffen Lang. Protein domain decomposition using spectral graph partitioning. 2007.

[27] D. Malmodin and M. Billeter. Multiway decomposition of nmr spectra with coupled evolution periods. *J. Am. Chem. Soc.*, 127(39):13486–13487, 2005.

[28] D. Malmodin and M. Billeter. Robust and versatile interpretation of spectra with coupled evolution periods using multi-way decomposition. *Mag. Reson. Chem.*, 44:S185–S195, 2006.

[29] L. Mirny and E. Domany. Protein fold recognition and dynamics in the space of contact maps. *Proteins: Struct., Funct., Genet.*, 26(4):391–410, 1998.

[30] Vijay S. Pande, Ian Baker, Jarrod Chapman, Sidney P. Elmer, Siraj Khaliq, Stefan M. Larson, Young Min Rhee, Michael R. Shirts, Christopher D. Snow, Eric J. Sorin, and Bojan Zagrovic. Atomistic protein folding simulations on the submillisecond time scale using worldwide distributed computing. *Biopolymers*, 68(1):91–109, 2003.

[31] A. Sali, E. I. Shakhnovich, and M. Karplus. Kinetics of protein folding. *J. Mol. Biol.*, 235:1614–1636, 1994.

[32] Tamar Schlick, R. D. Skeel, A. T. Brunger, L. V. Kale, J. Hermans, K. Schulten, and J. A. Board, Jr. Algorithmic challenges in computational molecular biophysics. *J. Comp. Phys.*, 151:9–48, 1999.

[33] Age Smilde, Rasmus Bro, and Paul Geladi. *Multi-way Analysis: Applications in the Chemical Sciences*. J. Wiley and Sons, Ltd., 2004.

[34] Jimeng Sun, Spiros Papadimitriou, and Philip Yu. Tensor analysis on multi-aspect streams. In *Learning from Data Streams*, pages 165–184. 2007.

[35] Jimeng Sun, Spiros Papadimitrou, and Christos Faloutsos. Distributed pattern discovery in multiple streams. Technical report, Carnegie Mellon University, 2006.

[36] Jimeng Sun, Spiros Papadimitrou, and Philip S. Yu. Window-based tensor analysis on high-dimensional and multi-aspect streams, 2006.

[37] Jimeng Sun, Dacheng Tao, and Christos Faloutsos. Beyond streams and graphs: Dynamic tensor analysis, 2006.

[38] D. M. F. Van Aalten, B. L. De Groot, J. B. C. Findlay, H. J. C. Berendsen, and Amadei A. A comparison of techniques for calculating protein essential dynamics. *J. Comp. Chem.*, 18(2):169–181, 1997.

[39] M. Vendruscolo and E. Domany. Efficient dynamics in the space of contact maps. *Folding and Design*, 3(5):329–336, 1998.

[40] Ying Xu, Dong Xu, and Harold N. Gabow. Protein domain decomposition using a graph-theoretic approach. *Bioinformatics*, 16(12):1091–1104, 2000.

[41] L. Yang, E. Eyal, C. Chennubhotla, J. Lee, A. M. Gronenborn, and I. Bahar. Insights into equilibrium dynamics of proteins from comparison of nmr and x-ray data with computational predictions. *Structure*, 15:1–9, 2007.

[42] Lizhuang Zhao and M. J. Zaki. Tricluster: An effective algorithm for mining coherent clusters in 3d microarray data. In *SIGMOD*, 2005.