Time-Varying Gaussian Graphical Models of Molecular Dynamics Data

Narges Sharif Razavian[1], Subhodeep Moitra[1], Hetunandan Kamisetty[2], Arvind Ramanathan[3], Christopher James Langmead[2,3,*]

May 2010
CMU-CS-10-119

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

[1]Language Technologies Institute, [2]Department of Computer Science, [3]Lane Center for Computational Biology, Carnegie Mellon University, Pittsburgh, PA, USA 15213.
*E-mail: cjl@cs.cmu.edu

This research was supported by NSF IIS-0905193 and an award from Microsoft Research to CJL.

Keywords: Structure Learning, Generative Models, Probabilistic Graphical Models, L1-Regularization, Time-Varying, Proteins, Molecular Dynamics

Abstract

We introduce an algorithm for learning sparse, time-varying undirected probabilistic graphical models of Molecular Dynamics (MD) data. Our method computes a *maximum a posteriori* (MAP) estimate of the topology and parameters of the model (i.e., structure learning) using L1-regularization of the negative log-likelihood (aka 'Graphical Lasso') to ensure sparsity, and a kernel to ensure smoothly varying topology and parameters over time. The learning problem is posed as a convex optimization problem and then solved optimally using block coordinate descent. The resulting model encodes the time-varying joint distribution over all the dihedral angles in the protein. We apply our method to three separate MD simulations of the enzyme Cyclophilin A, a peptidylprolyl isomerase. Each simulation models the isomerization of a different substrate. We compare and contrast the graphical models constructed from each data set, providing insights into the differences in the dynamics experienced by the enzyme for the different substrate.

1 INTRODUCTION

This paper introduces a novel method for analyzing and modeling Molecular Dynamics (MD) data. Molecular Dynamics simulations are an important tool for investigating the *energy landscape* that governs a system's behavior. Conceptually, an energy landscape is a complicated surface in a high-dimensional space (one dimension for each conformational degree of freedom in the protein). The surface may contain many local minima (called *sub-states*) separated by energy barriers. Our method builds a time-varying, undirected probabilistic graphical model of the system's internal degrees of freedom including the statistical couplings between them. The resulting model automatically reveals the conformational sub-states visited by the simulation, as well as the transition between them.

A system's ability to visit different sub-states is closely linked to important phenomena, including enzyme catalysis [7] and energy transduction [11]. For example, the primary sub-states associated with an enzyme might correspond to the unbound form, the enzyme-substrate complex, and the enzyme-product complex. The enzyme moves between these sub-states through *transition states*, which lie along the path(s) of least resistance over the energy barriers. Molecular Dynamics provide critical insights into these transitions.

Our method is motivated by recent advances in Molecular Dynamics simulation technologies. Until recently, MD simulations were limited to timescales on the order of several tens of nanoseconds. Today, however, the field is in the midst of a revolution, due to a number of technological advances in software (e.g., NAMD [17] and Desmond [9]), distributed computing (e.g., Folding@Home [16]), and specialized hardware (e.g., the use of GPUs [19] and Anton [18]). Collectively, these advances are enabling MD simulations into the millisecond range. This is significant because many biological phenomena, like protein folding and catalysis, occur on μs to msec timescales.

At the same time, long timescale simulations create significant computational challenges in terms of data storage, transmission, and analysis. Long-timescale simulations can easily exceed a terabyte in size. Our method builds a compact, generative model of the data, resulting in substantial space savings. More importantly, our method makes it easier to understand the data by revealing dynamic correlations that are relevant to biological function. Algorithmically, our approach employs L1-regularization to ensure sparsity, and a kernel to ensure that the parameters change smoothly over time. Sparse models often have better generalization capabilities, while smoothly varying parameters increase the interpretability of the model.

The contributions of this paper are as follows:

- The first application of structure learning (i.e., learning both the topology and parameters of the graphical model) to Molecular Dynamics data.
- An algorithm for learning globally optimal time-varying models in a regularized fashion.
- An analysis of the dynamics of the peptidylprolyl isomerase Cyclophilin A, including a comparative study of the enzyme bound to three different substrates.

2 BACKGROUND

Molecular Dynamics simulations involve integrating Newton's laws of motion for a set of atoms. Briefly, given a set of n atomic coordinates $\mathbf{X} = \{\vec{X}_1, ..., \vec{X}_n : \vec{X}_i \in \mathbb{R}^3\}$ and their corresponding velocity vectors $\mathbf{V} = \{\vec{V}_1, ..., \vec{V}_n : \vec{V}_i \in \mathbb{R}^3\}$, MD updates the positions and velocities of each atom according to an energy potential. The updates are performed via numerical integration, resulting in a conformational *trajectory*. When simulating reaction pathways, as is the case in our experiments, it is customary to analyze the trajectory along the *reaction coordinate* which simply describes the progress of the simulation through the pathway.

The size of the time step for the numerical integration is normally on the order of a femtosecond (10^{-15} sec), meaning that a 1 microsecond (10^{-6} sec) simulation requires one billion integration steps. In most circumstances, every 100th to 1000th conformation is written to disc as an ordered series of *frames*. Various techniques for analyzing MD data are then applied to these frames.

Traditional methods for analyzing MD data involve monitoring changes in global statistics (e.g., the radius of gyration, root-mean squared difference from the initial conformation, total energy, etc), and identifying sub-states using techniques such as quasi-harmonic analysis [10, 12], and other Principal Components Analysis (PCA) based techniques [6]. Quasi-harmonic analysis, like all PCA-based methods, implicitly assumes that the frames are drawn from a multivariate Gaussian distribution. Our method makes the same assumption but differs from quasi-harmonic analysis in three important ways. First, PCA usually averages over time by computing a single covariance matrix over the data. Our method, in contrast, performs a time-varying analysis, giving insights into how the dynamics of the protein change in different sub-states and the transition states between them. Second, PCA projects the data onto an orthogonal basis. Our method involves no change of basis, making the resulting model easier to interpret. Third, we employ regularization when learning the parameters of our model. Regularization is a common strategy for reducing the tendency to over-fit data by, informally, penalizing overly complicated models. In this sense, regularization achieves some of the same benefits as PCA-based dimensionality reductions, which is also used to produce low-complexity models.

The use of regularization is common in Statistics and in Machine Learning, but it has only recently been applied to Molecular Dynamics data [13, 14]. Previous applications focus on the problem of learning the parameters of force-fields for coarse-grained models, and rely on a Bayesian prior, in the form of inverse-Wishart distribution [13], or a Gaussian distribution [14] for regularization. Our method solves a completely different problem (modeling angular deviations of the all-atom model) and uses a different regularization scheme. In particular, we use L1 regularization, which is equivalent to using a Laplace prior. The use of L1 regularization is particularly appealing due to its theoretical properties of consistency — given enough data, the learning procedure learns the true model, and high statistical efficiency — the number of samples needed to achieve this guarantee is small.

3 ALGORITHM

Our goal is to build a time-varying statistical model of the dihedral angles in the protein. The dihedral angles for a given frame are easy to calculate, given the atomic coordinates. Each residue a protein has three *backbone* dihedral angles ϕ, ψ, and ω[1] (Fig. 1). Each residue also has between zero and four *side-chain* dihedral angles (depending on amino acid type), labeled as $\chi_1, ..., \chi_4$.

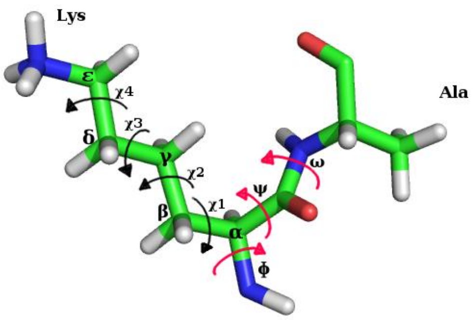

Figure 1: Backbone and side-chain dihedral angles. The dipeptide Lys-Ala is shown. The backbone angles ϕ, ψ, and ω are shown. The χ side-chain angles of Lys are also shown.

The ω angle (which spans the peptide bond) is normally fixed and thus often ignored. However, the three MD simulations we consider in this paper are of a peptidylprolyl isomerase — that is, an enzyme that transforms the ω angle of its substrate from either *cis* to *trans*, or *trans* to *cis* configuration (Fig. 2). The reaction coordinate for these simulations is the progress of the reaction from the *trans* to *cis* configuration (Fig. 3).

Let Θ be the set of dihedral angles in the protein and its substrate. Let $t \in [0, 1, ..., T]$ be an index along the reaction coordinate. Each position along the reaction coordinate corresponds to a subset of the frames in the trajectory. The probability distribution over Θ can therefore be modeled as a function of t. Our goal is to learn the distribution $f_{t=0...T}(\Theta(t))$, from the data. This distribution is encoded in the form of an undirected, time-varying *probabilistic graphical model* (PGM).

A stationary undirected PGM, also known as a Markov Random Field, is a tuple $\mathcal{M} = (\mathbf{V}, \mathbf{E}, \mathfrak{F})$. Here $\mathbf{V}$ is a set of nodes corresponding to a collection of random variables (in our case, one for each dihedral angle in the protein and its substrate), $\mathbf{E}$ is a set of undirected edges encoding the conditional independencies between random variables, and $\mathfrak{F}$ is a set of functions, also known as factors, on the nodes and edges. A Markov Random Field factors the joint distribution as the products of the factors. As previously mentioned, our method assumes that each random variable is distributed according a Gaussian distribution, giving rise to a so-called *Gaussian Graphical Model* (GGM).

We can extend the stationary model to the time varying case by making the set of edges and the parameters of the functions a function of t. That is, our model has the form: $\mathcal{M}(t) =$

[1] The N and C termini of the protein only have two and one backbone dihedrals, respectively.

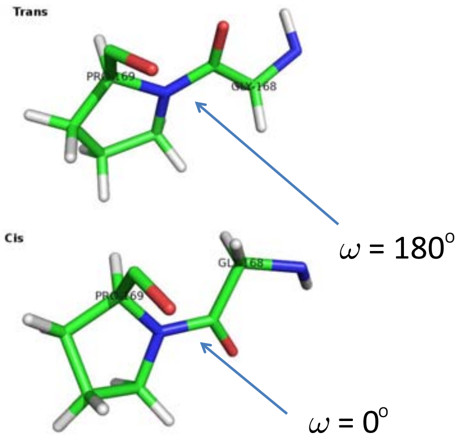

Figure 2: Top: *trans* configuration of Gly-Pro di-peptide. Bottom: *cis* configuration of the same di-peptide.

$(\mathbf{V}, \mathbf{E}(t), \mathfrak{F}(t))$. We learn the set of edges and the parameters of the functions in a regularized fashion. We also ensure that the edges and parameters evolve smoothly by integrating a kernel into the learning procedure. That kernel uses weighted combinations of the data from sequential positions along the reaction coordinate (i.e., $t \pm \tau : \tau > 0$) when estimating the parameters for position t.

The details of the model and the learning algorithm are presented in the following sections. We will start by the parameter estimation of a regularized Gaussian Graphical Model, and then give the extended algorithm to learn the time varying GGM.

3.1 Structure Learning for Multivariate Gaussian Graphical Models

Gaussian Graphical Models are multivariate probability distributions encoding a network of dependencies among variables. Let $\Theta = [\theta_1, \theta_2, .., \theta_n]$ be a set of n variables, such as n dihedral angles, and let $f(\Theta = D)$ be the value of the probability density function at a particular value D. A multivariate GGM factors this as:

$$f(\Theta = D) = \frac{1}{Z} \exp\{-\frac{1}{2}(D-\mu)^T \Sigma^{-1} (D-\mu)\} \tag{1}$$

Where $Z = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}}$ is the normalization coefficient. The parameters of this distribution are μ and Σ. μ is the vector of mean values of each variable, and Σ^{-1}, the inverse of the covariance matrix, contains the pairwise dependencies between the variables. A zero value in Σ^{-1} means that conditioned on the values of the other variables, the two corresponding variable are independent of each other.

Many algorithms have been proposed for learning the network and parameters of these models. Perhaps the most straight-forward approach simply maximizes the likelihood of the training data. This method simply estimates the μ and Σ by computing the average of each variable and the

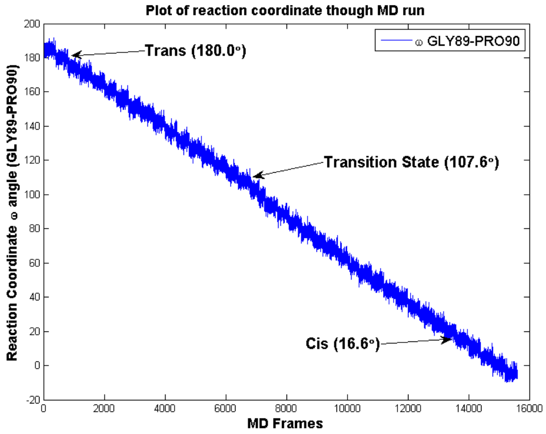

Figure 3: Reaction coordinate for our MD simulations. X-axis is the simulation frame number. The Y-axis is the ω angle of the Gly-Pro substrate. The simulation begins with the substrate in the *trans* configuration of the ω angle and ends in the *cis* configuration. The transition state, corresponding to the top of the energy barrier separating the two configurations is also labeled.

covariance between all pairs of variables. This method, while effective for small variable sets, is not applicable to the protein structure data for two reasons. First, proteins have, on average, about 5 dihedral angles per residue. Thus, even for a small 100-residue protein, $n \approx 500$, requiring n means and n^2 covariance estimates to be calculated. This is important because the number of training samples required in order to achieve a given confidence level in the parameter estimates increases polynomially in n. Second, a maximum-likelihood estimate of the parameters will lead to a very dense set of dependencies (i.e., Σ has few zero entries), making it difficult to interpret the resulting model. In contrast, we will use *regularization* to learn a sparse GGM, thereby addressing both issues, as discussed below.

3.2 Regularized Gaussian Graphical Models

There are two main approaches for learning sparse GGMs. The first approach uses a greedy approach to find the zero elements of the Σ^{-1} through neighborhood selection for each node [4, 15]. Such methods do not scale to large variable sets. The second approach uses a global regularization penalty, and then solves the resulting optimization problem by invoking algorithms for global optimization. Our method uses the L1-regularization penalty, which is currently the most common penalty function.

When estimating the parameters of a stationary GGM, the problem is formulated as follows: Given a set of training data $\mathbf{D} = D_{(1)}, ..., D_{(m)}$, where D_i is a $n \times 1$ vector, the sample covariance matrix is defined as:

$$S = \frac{1}{n}\sum_{k=1}^{m}(D_{(k)} - \mu)(D_{(k)} - \mu)^T \qquad (2)$$

where μ is a $n \times 1$ vector encoding the sample mean. The L_1 regularized log-likelihood of X the estimate of Σ^{-1} is then:

$$ll(X|D) = \sum_i \log f(\Theta = D_{(i)r}) - \lambda \|X\|_1$$

Here, $\|X\|_1$ is the L1-regularization penalty, which is defined as the sum of the absolute values of the elements of X and λ is the scalar coefficient that controls the size of regularization penalty, and is usually estimated through cross validation.

Using the functional form of f according to equation 1, this can be rewritten as:

$$= -\log(|X^{-1}|) - \sum_{k=1}^{m} (D_{(k)} - \mu) X (D_{(k)} - \mu) - \lambda \|X\|_1$$

using $|X^{-1}| = \frac{1}{|X|}$ and $trace(ABC) = trace(CAB)$,

$$= \log(|X|) - trace(\mathbf{D} - \mu) X (\mathbf{D} - \mu) - \lambda \|X\|_1$$

Plugging in the definition of S according to equation 2, we then get the MAP estimate of Σ^{-1}:

$$\Sigma^{-1} = \arg\max_{X \succ 0} \; \log|X| - trace(SX) - \lambda \|X\|_1 \qquad (3)$$

In the next section we extend this formulation to the regularized time varying GGM and provide the optimization algorithm that solves both the stationary and the time varying GGM parameter estimation problems.

3.3 Regularized Time Varying Gaussian Graphical Models

Having defined the problem of stationary GGM parameter estimation, we now discuss the time varying case. We will follow the formulation of Zhou [22] for this problem.

Problem Formulation

Let $D^{1..T}_{(1),..(m)}$ be the set of training data, where each $D^{t}_{(i)}$ is a sample represented by n variables. For instance, in our modeling of MD data, each $D^{t}_{(i)}$ is a protein conformation. The time varying GGM parameter estimation algorithm extends the stationary GGM parameter learning as follows:

$$\Sigma^{-1}(t) = \arg\max_{X \succ 0} \log|X| - trace(S(t)X) - \lambda \|X\|_1$$

Here, $S(t)$ is the $weighted\ covariance\ matrix$, and is calculated as follows:

$$S(t) = \frac{\sum_{s=1}^{T} \sum_{i=1}^{m} w_{st} (D_i^{(s)} - \mu)(D_i^{(s)} - \mu)^T}{\sum_{s=1}^{T} w_{st}}$$

The weights w_{st} are defined by a symmetric nonnegative kernel function.

Choice of the Kernel Function

The choice of the kernel function will let the model select for its specificity. A kernel with a larger span will cause the time varying model to be less sensitive to abrupt changes in the network and capture the slower and more robust behaviors. On the other hand, as the kernel function span decreases, the time varying will be able to capture more short term patterns of interaction.

In our experiments we used a kernel from a triangular family which spans over 5 simulations before and after the experiment (Fig. 4). Experimenting with other Kernel families, and different kernel spans in an important part of our future work, which we will mention in the final section of this paper.

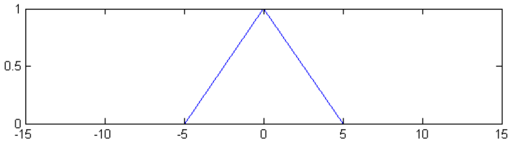

Figure 4: The Kernel functions of triangular family used in our experiment. $K = 1 - \frac{|x|}{5} * 1_{\{|x|<5\}}$

Convex Optimization for Parameter Estimation of Regularized Time Varying GGM

We use Block Coordinate Descent Algorithm to solve the stationary and time varying problems. This method has been proposed by Banerjee et. al. [5], and proceeds by forming the dual for the optimization case, and applying block coordinate descent to the dual form.

Recall that the primal form of both the stationary and time varying case is as follows:

$$\Sigma^{-1} = \arg\max_{X \succ 0} \log|X| - trace(SX) - \lambda\|X\|_1$$

To take the dual, we first rewrite the L1-norm as:

$$\|X\|_1 = \max_{\|U\|_\infty \leq 1} trace(XU)$$

where $\|U\|_\infty$ denotes the maximum absolute value element of the matrix U. Given this change of formulation, we can rewrite the primal form of the problem as:

$$\Sigma^{-1} = \max_{X \succ 0} \min_{\|U\|_\infty \leq \lambda} \log|X| - trace(X, S + U)$$

Thus, the optimal Σ^{-1} is the one that maximizes the worst case log likelihood, over all additive perturbations of the covariance matrix, S. Next, to obtain the dual form, we exchange the min and max, and the inner max objective function can now be solved analytically taking the gradient and setting it to zero. This results in the new form of the objective function:

$$U^* = \min_{\|U\|_\infty \leq \lambda} - \log|S + U| - n$$

where n is the number of features in each sample. Once we solve this problem, the optimal Σ^{-1} can be computed as $\Sigma^{-1} = (S + U^*)^{-1}$.

Performing one last change of variables $W = S + U$, and forming the dual of the problem will bring us to the final form of our objective:

$$\Sigma^* = \max\{\log|W| : \|W - S\|_\infty \leq \lambda\}$$

This problem is smooth and convex, and for small values of n it can be solved by standard optimization techniques like interior point method. For larger values of n the interior point method becomes too inefficient, and another method, called Block Coordinate Descent can be used instead [5].

Block Coordinate Descent

The Block Coordinate Descent algorithm works as follows. For any matrix A, let $A_{\backslash k \backslash j}$ denote the matrix produced by removing column k and row j of the matrix. Let A_j also denote the column j, with diagonal element A_{jj} removed. The Block Coordinate Descent algorithm proceeds by optimizing one row and one column of the variable matrix W at a time. The algorithm iteratively optimizes all columns until a convergence criteria is met. The algorithm is as follows:

```
====================================
Initialize $W^{(0)} := S + \lambda I$
Repeat until convergence
1. For $j = 1, \ldots n$
1$(a)$ $y^* = \arg\min_y\{y^T W^{(j-1)}_{\backslash j \backslash j} y : \|y - S_j\|_\infty \leq \lambda\}$
    Where $W^{(j-1)}$ denotes the current iterate.
1$(b)$ Update $W^{(j)}$ as $W^{(j-1)}$ with column/row $W_j$ replaced by $y^*$.
2. Let $W^{(0)} = W^{(n)}$
3. Test for convergence when the $W^{(0)}$ satisfies:
    $trace((W^{(0)})^{-1}S) - n + \lambda\|(W^{(0)})^{-1}\|_1 \leq \epsilon$.
====================================
```

The $W^{(j)}$s produced in each step are strictly positive definite. This property is important because the dual problem estimates the covariance matrix Σ, rather than the inverse covariance matrix. The network conditional dependencies which we are interested in are encoded in the inverse covariance matrix, Σ^{-1}, so the strictly positivity of $W^{(j)}$ will guarantee that the optimum Σ will be reversible, and that we can compute the final answer Σ^{-1} from the $W^{(j)}$.

The time complexity of this algorithm has also been estimated to be $O(n^{4.5}/\epsilon)$ [5], when converging to ϵ suboptimal solution. This complexity is better than $O(n^6/\log(\frac{1}{\epsilon}))$, which would have been achieved using the interior point method on the dual form [20].

We used this algorithm in our experiments to estimate a L1 Regularized Time Varying Gaussian Graphical Model on the MD simulation data. The experimental conditions, model selection and the result of the experiments will be presented in the next section.

4 RESULTS

We applied our method to three simulations of the human form of the enzyme *cyclophilin A (CypA)*. CypA isomerizes the ω bond of its substrate and it is an important receptor for several immunosuppresive drugs and HIV infection. Our three simulations correspond to three different substrates: (i) The hexa-peptide His-Ala-Gly-Pro-Ile-Ala from the HIV-1 capsid protein (PDB ID: 1AWQ); (ii) the dipeptide Ala-Pro (PDB ID: 2CYH); and (iii) the tetra-peptide Ala-Ala-Pro-Phe (PDB ID: 1RMH).

Previous studies have identified a set of 25 highly conserved residues in the cyclophilin family [3]. In particular, residues P30, T32, N35, F36, Y48, F53, H54, R55, I57, F60, M61, Q63, G65, F83, E86, L98, M100, T107, Q111, F112, F113, I114, L122, H126, F129 are all highly conserved. Experimental work [8] and MD simulations [1, 3] have also implicated these residues as forming a network that influences the substrate isomerization process. Significantly, this network extends from the flexible surface regions of the protein to the active site residues of the enzyme (residues R55, F60, M61, N102, A103, F113, L122, and H126). The previous studies identified this network by examining atomic positional fluctuations and the correlations between them. In contrast, our study focuses on the angular correlations, as revealed by our algorithm. Positional fluctuations are ultimately caused by the angular fluctuations, so our study is complementary to the previous work.

4.1 Simulations

The details of the three MD data sets have been reported previously [3]. Briefly, each data set consists is generated by performing 39 independent simulations in explicit solvent along the reaction coordinate. The first simulation starts with the substrate's ω angle at $180°$ (i.e., *trans*) from which 400 frames are extracted, corresponding to 400 ps of simulated time. The second simulation starts with the substrate's ω angle at $175°$, from which another 400 frames are obtained. Subsequent simulations increment the ω by $5°$ until the $0°$ (i.e., *cis*) configuration is reached. Each frame corresponds to one protein conformation, and is represented as a vector of dihedral angles – one for each variable. For each residue there is a variable for each of ϕ, ψ, ω, and the side chain angles χ (between 0 and 4 variables, depending on residue type). The time-varying graphical models are learned from the resulting 15,600 frames.

4.2 Model Selection

Our algorithm has one parameter, λ, which penalizes the number of edges in the learnt model. We first sought to identify a λ value for which the set of learned edges can be deemed significant. To do this, we performed a permutation study. The input to our algorithm is a $N \times M$ matrix, where N is the number of frames used to learn the model, and M is the number of dihedral angles in the system. The contents of each column was randomly permuted in order to decouple the angles. The algorithm was then run for different values of λ in order to find a value where the learned model had zero edges, under the assumption that the randomly permuted columns contained no significant couplings. The value $\lambda = 1,000$ was found to be the smallest value consistently giving zero edges across all three data sets. In our experiments we used a more stringent value ($\lambda = 5,000$) in order to ensure that our edges don't reflect spurious correlations. This conservative choice reflects the importance of not including any spurious correlations in our final results.

4.3 Edge Density Along Reaction Coordinate

As previously mentioned, each data sets comprises 39 individual simulations. The learning algorithm identifies a set of edges in each simulation, employing a kernel to ensure smoothly varying sets of edges. Figure 5 plots the number of edges for data set along the reaction coordinate. Qualitatively, the number of edges decreases until the transition state, and then rises for each substrate. The three substrates, however, also show significant differences in the number of local minima, the location and width of the minima, and the minimum number of edges.

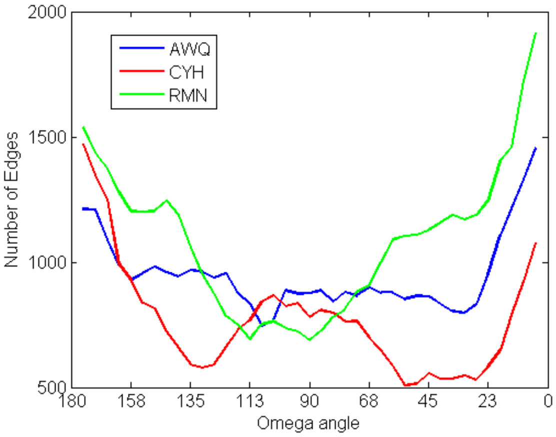

Figure 5: Edge Density Along Reaction Coordinate. The number of edges learned from the three MD simulations of CypA in complex with three substrates (AWQ, CYH, and RMH) are plotted as a function of the ω angle. AWQ is the largest substrate, CYH is the smallest substrate.

Differences in the number and width of minima might be suggestive of differences in the kinetics of the reactions, although we have not been able to identify any published data on the isomerization rates for these specific substrates. We note, however, that the magnitude of the minima is correlated with the size of the substrate. In particular, the minimum value of the curve labeled AWQ (the largest substrate) is larger than the minimum value of the curve labeled RMH (the second largest substrate) which, in turn, is larger than the minimum value of the curve labeled CYH (the smallest substrate). Edge density corresponds to the total amount of coupling in the system. Thus, these results suggest that when approaching the transition state the angles tend to decouple. At the same time, the dependency on size suggest that larger substrates may require more coupling than smaller ones in order to pass through the transition state of the reaction coordinate.

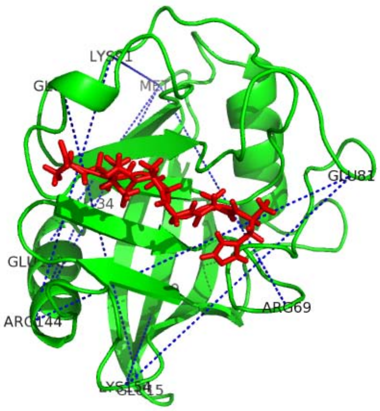

Figure 6: Top 10 Persistent Edges. For simplicity, only the top 10 couplings are shown.

4.4 Persistent, Conserved Couplings

We next examined the set of edges to identify the persistent couplings. That is, edges that are observed across the entire reaction coordinate and in all three simulations. We computed $P^a_{i,j}$, the probability that edge (i, j) exists in substrate a. Then, we computed the product $P_{i,j} = P^a_{i,j} * P^b_{i,j} * P^c_{i,j}$ as a measure of persistence. We then identified the edges where $P_{i,j} > 0.5$, yielding a total of 73 edges (out of $\binom{165}{2} = 13,530$ possible edges). The top 10 of these edges are shown in Figure 6. Notice that the edges span large distances. Each of the top 10 edges relates how distal control could occur within CypA; these edges typically connect one network region with the other. For example, region 13-15 is connected to 146-152 which connect to farther off regions including 68-76 and 78-86.

Couplings to the Active Site and Substrate

According to our analysis of the dihedral angular fluctuations, the set of residues most strongly coupled to the substrate are residues 1, 13, 14, 125, 147, and 157. None of these residues is in the active site (residues 55, 60, 61, 102, 103, 113, 122, 126), although residue 125 is sequentially adjacent to an active site residue. The set of resides most strongly coupled to the active site include residues 1, 9, 13, 14, 81, 86, 91, 120, 125, 142, 151, 154, and 165. Of these, only residue 86 is among the previously cited list of highly conserved residues. Thus, the conservation of angular deviations observed across substrates is distinct from the residue conservation within the family. We can conclude that the conservation of angular deviation is an inherent feature of the structure of the protein, as opposed to its sequence.

4.5 Transient, Conserved Couplings

Next, we identified the edges that are found across all three substrates, but are only found in one segment of the reaction coordinate. To do this we first partitioned the reaction coordinate into three parts: (i) $\omega \in [180, 120)$; (ii) $\omega \in [120, 60)$; and (iii) $\omega \in [60, 0]$, which we will refer to as the *trans*, *transition*, and *cis* states, respectively. We then identified the edges that occur exclusively in the *trans* state, those occurring exclusively in the transition state, and those occurring exclusively in the *cis* state. Four such edges were found for the *trans* state: (49,81), (1,143), (143, 144), and (1 154); five edges were found for the transition state: (9,157),(82,140), (9,157), (91, 157), and (144, 157); and sixty one edges were found for the *cis* state. A subset of these edges are shown in Figure 7. The coupling of the edges reveal clues about how couplings between network regions varies with the reaction coordinate. In the trans state one can see couplings between network regions 142-156 and 78-86, while in the cis state there are couplings between network regions 13-15 and 89-93.

4.6 Substrate-Specific Couplings

Finally, we identified couplings that are specific to each substrate. As in the previous section, we partitioned the reaction coordinate into the *trans*, transition, and *cis* states. We then identified the edges that occur exclusively in the AWQ substrate, those occurring exclusively in the CYH substrate, and those occurring exclusively in the RMH substrate.

We found 62, 8, and 24 such edges, respectively. A subset of those edges are shown in Figure 8. Looking at the couplings one can notice that the edges lie on the network regions (13-15, 68-74, 78-86 and 146-152). However, the coupled residues change from substrate to substrate which implies a certain specificity in the dynamics.

5 DISCUSSION AND CONCLUSION

Molecular Dynamics simulations provide important insights into the role that conformational fluctuations play in biological function. Unfortunately, the resulting data sets are both massive and complex. Previous methods for analyzing these data are primarily based on dimensionality reduction techniques, like Principal Components Analysis, which involves averaging over the entire data set and projects the data into a new basis. Our method, in contrast, builds a time-varying graphical model of the data, thus preserving the temporal nature of the data, and presenting data in its original space. Moreover, our methods uses L1 regularization when learning leading to easily interpretable models. The use of L1 regularization also confers desirable theoretical properties in terms of consistency and statistical efficiency. In particular, given enough data, our method will learn the ‘true’ model, and the number of samples needed to achieve this guarantee is small.

We demonstrated our method on three simulations of Cyclophilin A, revealing both similarities and differences across the substrates. Coupling tends to first decrease and then increase along the reaction coordinate. As observed from Fig. 5, the variation in simulations with longer peptides (1AWQ and 1RMH) show similar behavior in and around the transition state, while 1CYH, with the dipeptide shows an increase in the number of edges. This difference is perhaps a result of the

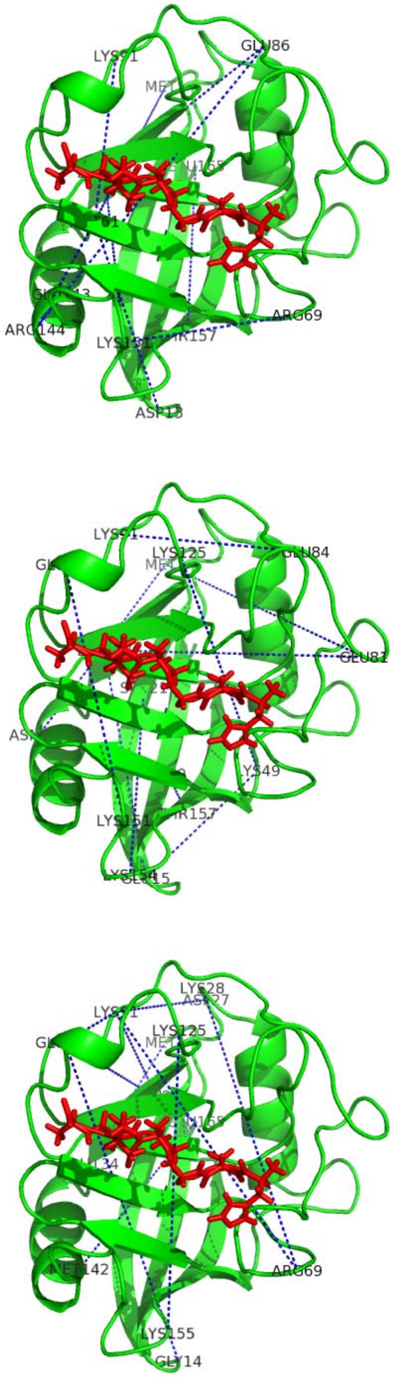

Figure 7: Transient Edges. The set of edges seen exclusively in the *trans* (top), transition (middle), and *cis* (bottom) states, respectively. For simplicity, only the top 10 couplings are shown.

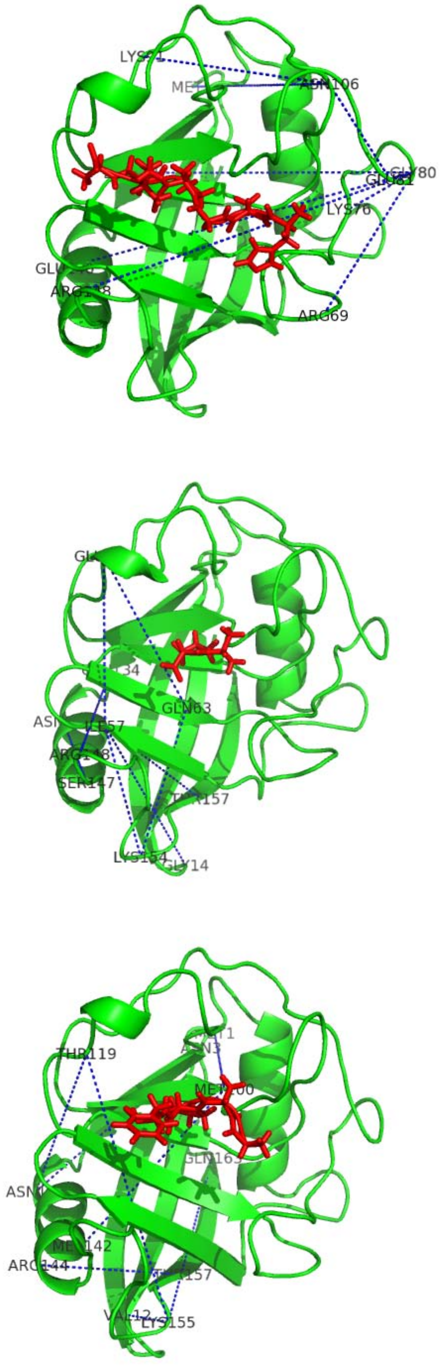

Figure 8: Substrate-specific Edges. The set of edges seen exclusively in the AWQ (top) CHY (middle), and RMH (bottom) substrates. For simplicity, only the top 10 couplings are shown.

fact that dipeptides such as Ala-Pro can potentially act as inhibitors for CypA [21]. Although, the significance of these differences cannot be discussed in the light of mechanistic behavior in CypA, the ability of our method to detect subtle, yet important changes during the course of such simulations is in itself a valuable tool for biologists.

There is also evidence that there are both state-specific and substrate-specific couplings, all of which are automatically discovered by the method. We have discovered that over the course of the reaction, the network regions as identified by previous work [2] couple directly to the active site residues (see Fig. 7). The method is also able to pick out subtle changes in the dynamics as seen by the edges that appear in substrate-specific couplings (see Fig. 8). These differences are present exactly on the network regions, implying that the alteration in the dynamics of these regions may be responsible for catalysis with respect to specific substrates. An interesting direction of further research is to study how presence of CypA inhibitors such as cyclosporin can alter the dynamics in these network regions to understand the mechanistic underpinnings of CypA function.

There are a number of interesting directions for future work. First, while our method was used to learn graphical models over dihedral angle fluctuations, there is no reason why it can't also be used to learn models over positional fluctuations, or mixtures of angular and positional fluctuations. Indeed, the ability to combine different kinds of features into a single probabilistic framework is one of the key advantages of graphical models. For example, one might envision examining how angular fluctuations affect positional fluctuations, the presence of hydrogen bonds, etc. Second, our model assumes that the underlying distribution is multivariate Gaussian. One can imagine using different assumptions about the parametric form of the variables (e.g., multinomial, von Mises, etc). We are presently exploring such alternatives. Finally, our experiments were limited in that they only examined a triangular kernel. An obvious direction for future work is to examine the use of alternative kernels, including asymmetric varieties.

Acknowledgements

We would like to thank Dr. Pratul Agarwal for use of his MD data.

References

[1] P. K. Agarwal. Cis/trans isomerization in hiv-1 capsid protein catalyzed by cyclophilin a: Insights from computational and theoretical studies. *Proteins: Struct., Funct., Bioinformatics*, 56:449–463, 2004.

[2] P. K. Agarwal. Computational studies of the mechanism of cis/trans isomerization in hiv-1 catalyzed by cyclophilin a. *Proteins: Struct. Funct. Bioinform.*, 56:449–463, 2004.

[3] 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*, 43:10605–10618, 2004.

[4] A. Ahmed and E. Xing. Recovering time-varying networks of dependencies in social and biological studies. *Proceedings of the National Academy of Sciences*, 106(29):11878–11883, July 2003.

[5] O. Banerjee, L. El Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *Journal of Machine Learning Research*, 9:485–516, 2008.

[6] H. J. C Berendsen and S. Hayward. Collective protein dynamics in relation to function. *Current Opinion in Structural Biology*, 10(2):165–169, 2000.

[7] D.D. Boehr, D. McElheny, H.J. Dyson, and P.E. Wright. The dynamic energy landscape of dihydrofolate reductase catalysis. *Science*, 313(5793):1638–1642, 2006.

[8] 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.

[9] K. J. Bowers, E. Chow, H. Xu, R. O. Dror, M. P. Eastwood, B. A. Gregersen, J. L. Klepeis, I. Kolossvary, M. A. Moraes, F. D. Sacerdoti, J. K. Salmon, Y. Shan, and D. E. Shaw. Scalable algorithms for molecular dynamics simulations on commodity clusters. *SC Conference*, 0:43, 2006.

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

[11] David M. Leitner. Energy flow in proteins. *Annu. Rev. Phys. Chem.*, 59:233–259, 2008.

[12] R. M. Levy, A. R. Srinivasan, W. K. Olson, and J. A. McCammon. Quasi-harmonic method for studying very low frequency modes in proteins. *Biopolymers*, 23:1099–1112, 1984.

[13] P. Liu, Q. Shi, H. Daumé III, and G.A. Voth. A bayesian statistics approach to multiscale coarse graining. *J Chem Phys.*, 129(21):214114–11, 2008.

[14] L. Lu, S. Izvekov, A. Das, H.C. Andersen, and G.A. Voth. Efficient, regularized, and scalable algorithms for multiscale coarse-graining. *J. Chem. Theory Comput.*, 6:954ñ965, 2010.

[15] Nicolai Meinshausen, Peter Bhlmann, and Eth Zrich. High dimensional graphs and variable selection with the lasso. *Annals of Statistics*, 34:1436–1462, 2006.

[16] V. S. Pande, I. Baker, J. Chapman, S. P. Elmer, S. Khaliq, S. M. Larson, Y. M. Rhee, M. R. Shirts, C.D. Snow, E. J. Sorin, and B. Zagrovic. Atomistic protein folding simulations on the submillisecond time scale using worldwide distributed computing. *Biopolymers*, 68(1):91–109, 2003.

[17] J. C. Philips, R. Braun, W. Wang, J. Gumbart, E. Tajkhorshid, E. Villa, C. Chipot, R. D. Skeel, L. V. Kale, and K. Schulten. Scalable molecular dynamics with namd. *J. Comp. Chem.*, 26(16):1781–1801, 2005.

[18] D. E. Shaw, M. M. Deneroff, R. O. Dror, J. S. Kuskin, R. H. Larson, J. K. Salmon, C. Young, B. Batson, K. J. Bowers, J. C. Chao, M. P. Eastwood, J. Gagliardo, J. P. Grossman, C. R. Ho, D. J. Ierardi, I. Kolossváry, J. L. Klepeis, T. Layman, C. McLeavey, M. A. Moraes, R. Mueller, E. C. Priest, Y. Shan, J. Spengler, M. Theobald, B. Towles, and S. C. Wang. Anton, a special-purpose machine for molecular dynamics simulation. In *ISCA '07: Proceedings of the 34th annual international symposium on Computer architecture*, pages 1–12, New York, NY, USA, 2007. ACM.

[19] J.E. Stone, J. C. Phillips, P. L. Freddolino, D. J. Hardy, L. G. Trabuco, and K. Schulten. Accelerating molecular modeling applications with graphics processors. *J. Comp. Chem.*, 28:2618–2640, 2007.

[20] L. Vandenberghe, S. Boyd, and S.-P. Wu. Determinant maximization with linear matrix inequality constraints. *SIAM Journal on Matrix Analysis and Applications*, 19:499–533, 1998.

[21] Yingdong Zhao and Hengming Ke. Mechanistic implication of crystal structures of the cyclophilindipeptide complexes,. *Biochemistry*, 35(23):7362–7368, 06 1996.

[22] Shuheng Zhou, John D. Lafferty, and Larry A. Wasserman. Time varying undirected graphs. In *COLT*, pages 455–466, 2008.