1-1-2014

Generative models of conformational dynamics.

Christopher J. Langmead
Carnegie Mellon University, cjl@cs.cmu.edu

Published In
Advances in experimental medicine and biology, 805, 87-105.

Published in final edited form as:
Adv Exp Med Biol. 2014 ; 805: 87–105. doi:10.1007/978-3-319-02970-2_4.

Generative Models of Conformational Dynamics

Christopher James Langmead, Ph.D.
Carnegie Mellon University, Pittsburgh, PA, USA, cjl@cs.cmu.edu

Abstract

Atomistic simulations of the conformational dynamics of proteins can be performed using either Molecular Dynamics or Monte Carlo procedures. The ensembles of three-dimensional structures produced during simulation can be analyzed in a number of ways to elucidate the thermodynamic and kinetic properties of the system. The goal of this chapter is to review both traditional and emerging methods for learning *generative models* from atomistic simulation data. Here, the term 'generative' refers to a model of the joint probability distribution over the behaviors of the constituent atoms. In the context of molecular modeling, generative models reveal the correlation structure between the atoms, and may be used to predict how the system will respond to structural perturbations. We begin by discussing traditional methods, which produce multivariate Gaussian models. We then discuss GAMELAN (GrAphical Models of Energy LANdscapes), which produces generative models of complex, non-Gaussian conformational dynamics (e.g., allostery, binding, folding, etc) from long timescale simulation data.

1 Introduction

Atomistic simulations are widely used to investigate the conformational dynamics of proteins and other molecules (e.g., [22, 24]). The raw output from any simulation is an ensemble of three-dimensional conformations. These ensembles can be analyzed using a variety of methods, ranging from simple descriptive statistics (e.g., average energies, radius of gyration, etc) to generative models (e.g., normal mode analysis, quasi-harmonic analysis, etc). Here, the term 'generative' refers to any model of the joint probability distribution, $P(X_1, \ldots, X_n)$, over a set of user-defined random variables, $\mathbf{X} = \{X_1, \ldots, X_n\}$, representing the system's degrees of freedom (e.g., distances, fluctuations, angles, etc). In this chapter, we focus on techniques for learning generative models from conformational ensembles.

Generative models provide important insights into conformational dynamics. In particular, they elucidate the inter-atomic correlations that give rise to collective motions within and across dynamical domains. Consequently, generative models can be used to estimate important quantities, including the magnitudes of atomic fluctuations (e.g., [50]), configurational entropies (e.g., [21]), and free energies (e.g., [17, 18, 19]). They can also be used to predict how the system will respond to local structural changes (e.g., ligand binding) (e.g., [40]).

Many techniques exist for learning generative models from conformational ensembles. Well-known examples include: Normal Modes Analysis [6, 13, 25], Quasi Harmonic Analysis [21, 26], Essential Dynamics [1], and Elastic Network Models [50]. Ultimately, the differences between these methods amount to: (a) which variables are modeled, and (b) the mathematical form used to define $P(\mathbf{X})$. This chapter contrasts several strategies for specifying $P(\mathbf{X})$ (whether implicitly or explicitly), starting from simple harmonic models (where $P(\mathbf{X})$ takes the form of a multivariate Gaussian), and proceeding to more expressive models that are better suited for anharmonic (i.e., non-Gaussian) motions. This latter category is presented in the context of GAMELAN (GrAphical Models of Energy LANdscapes), which is a new framework for learning generative models from conformational ensembles.

GAMELAN is motivated by recent developments in atomistic simulation technologies. In particular, advances in hardware and software (e.g., [5, 15, 32, 35, 41, 47]) have dramatically increased the timescales accessible to simulation. Microsecond ($\mu s = 10^{-6}$ sec.) and millisecond ($ms = 10^{-3}$ sec.) simulations are increasingly common, but the resulting conformational ensembles pose significant challenges. First and foremost, the conformational dynamics observed on the μs and *ms* timescales are usually very complex. In particular, they are not well suited to harmonic approximations. GAMELAN addresses this problem by providing users the option of learning multi-modal, non-Gaussian, and even time-varying generative models from the ensemble. This is achieved through a combination of parametric, semi-parametric, and non-parametric models. The second challenge is the size of the ensemble, which naturally increases with both the size of the system and the timescale. GAMELAN addresses this challenge by using efficient, but provably optimal algorithms for estimating the parameters of the generative model.

2 Conformational Ensembles

As previously noted, atomistic simulations can be performed using Molecular Dynamics (MD) and/or Monte Carlo (MC) sampling. Molecular dynamics simulations involve numerically solving Newton's laws of motion for a system of atoms whose interactions are defined according to a given force field. Monte Carlo simulations involve iteratively modifying an existing structure. Each modification is either accepted or rejected, stochastically, according to its energy, as defined by a force field. The theory and practice behind MD and MC algorithms is beyond the scope of this chapter. Here, we will simply assume that each method produces an ensemble of m conformations. The ensemble will be denoted as $\mathbf{C} = \{C(1), \ldots, C(m)\}$, where $C(i)$ specifies the cartesian coordinates for each atom in the ith conformation.

In principle, generative models can be constructed from the raw ensemble, **C**, but it is much more common to limit the analysis to a limited number of covariates. Most analyses operate on either: (a) the cartesian coordinates of a subset of the atoms (e.g., non-solvent molecules, or even just the alpha carbons); (b) atomic fluctuations (i.e., displacements from a reference conformation); (c) pairwise distances between atoms; or (d) dihedral angles. The methods discussed in this chapter can be applied to any set of covariates, and so we will not restrict them to any particular type. Let $\mathbf{X} = \{X_1, \ldots, X_n\}$ be a vector encoding the n covariates to be

analyzed, and recall that a generative model encodes the joint probability distribution $P(\mathbf{X})$. The parameters of the model are estimated from a set of data, $\mathbf{D} = \{\mathbf{X}(1), \ldots, \mathbf{X}(m)\}$, where $\mathbf{X}(i)$ is a vector containing the values of the n covariates extracted from $C(i)$.

3 Learning Generative Models from Conformational Ensembles

This section presents several methods for learning generative models from a set of data, **D**, starting with simple Gaussian models and progressing to non-Gaussian models.

3.1 Simple Gaussian Models

The most straightforward way to produce a model of the joint distribution, $P(\mathbf{X})$, is to fit a multivariate Gaussian distribution to the data. This can be accomplished, for example, by computing the n-dimensional empirical mean vector, $\mu = \frac{1}{m}\sum \mathbf{X}(i)$, and the $n \times n$ empirical covariance matrix $\Sigma = E[(\mathbf{X} - \mu)(\mathbf{X} - \mu)^T]$. Given these parameters, the probability density for any n-vector $\mathbf{x} = \{x_1, \ldots, x_n\}$ is:

$$P(\mathbf{x}) = \frac{1}{Z}\exp\left\{-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right\}, \quad (1)$$

where $Z = \sqrt{(2\pi)^n|\Sigma|}$ is the partition function and $|\Sigma|$ denotes the determinant of Σ.

Well-known methods for building harmonic models, including Normal Modes Analysis [6, 13, 25], Quasi Harmonic Analysis [21, 26], and Essential Dynamics [1], also produce multivariate Gaussian models, but not in the manner outlined above. Instead, they transform the data in some way. Quasi-Harmonic Analysis, for example, performs Principle Components Analysis (PCA) on a mass-weighted covariance matrix of atomic fluctuations. PCA diagonalizes the covariance matrix, producing a set of eigenvectors and their corresponding eigenvalues. Each eigenvector can be interpreted as one of the principal modes of vibration within the system or, equivalently, as a univariate Gaussian with zero mean and variance proportional to the corresponding eigenvalue. The eigenvectors are orthogonal by construction, and so the off-diagonal elements of the correlation matrix are zero.

Principal Components Analysis operates on covariance matrices, which capture pairwise relationships between variables. It is sometimes desirable to capture the relationships between tuples of variables (triples, quadruples, etc). Here, Tensor Analysis may be used instead of PCA [36, 37]. The model produced via Tensor Analysis is also Gaussian.

Computing with Gaussian Models—When appropriate, multivariate Gaussian models have a number of attractive properties. For example, the Kullback-Leibler divergence[1] between two different models, M_0 and M_1 can be computed analytically:

[1]The Kullback-Leibler divergence is a non-symmetric measure of the difference between two distributions. It is non-negative and zero if and only if the two distributions are identical. The divergence can be symmetrized by taking the sum or average of $KL(M_0 \| M_1)$ and $KL(M_1 \| M_0)$.

$$KL(M_0 \parallel M_1) = \frac{1}{2}(\mathrm{trace}(\Sigma_1^{-1}\Sigma_0) + (\mu_1 - \mu_0)^T \Sigma_1^{-1}(\mu_1 - \mu_0) - \ln(|\Sigma_0|/|\Sigma_1|) - n). \quad (2)$$

The ability to quantify the differences between two models has a number of practical uses. For example, the symmetric version of the Kullback-Leibler can be used to cluster a set of models, or to compare models learned from independent sources (e.g., different simulations of the same system).

The generative nature of the model means, among other things, that one can sample new conformations (i.e., those that weren't in **D**). In the case of a multivariate Gaussian, sampling can be accomplished in two steps. First, an *n*-dimensional vector of independent Gaussian random numbers is generated, $\mathbf{r} = [r_1, \ldots, r_n]$. Second, the random sample is produced by computing $\mathbf{x} = \mathbf{A}\mathbf{r} + \mu$, where **A** is the lower triangular matrix satisfying $\Sigma = \mathbf{A}\mathbf{A}^T$.

Finally, Gaussian models also make it easy to predict how the system will respond to local perturbations by computing conditional distributions. For example, let $\mathbf{V} \subset \mathbf{X}$ be an arbitrary subset of **X**, and let $\mathbf{W} = \mathbf{X} \setminus \mathbf{V}$ be the complement set. Here, **V** might correspond to an allosteric binding site. We can simulate a local structural change (e.g., due to binding) by setting **V** to some particular value, say **v**. Next, we can predict how the rest of the molecule will respond by conditioning the distribution on **v**, and computing the conditional distribution $P(\mathbf{W} \mid \mathbf{v})$. This conditional distribution is also a multivariate Gaussian with parameters $(\mu_{W \mid \mathbf{v}}, \Sigma_W)$ where:

$$\mu_{W|\mathbf{v}} = \mu_W + \Sigma_W^T \Sigma_{VV}^{-1}(\mathbf{v} - \mu_\nu) \quad (3)$$

$$\Sigma_W = \Sigma_{WW} - \Sigma_{WV}^T \Sigma_{VV}^{-1} \Sigma_{WV} \quad (4)$$

Here, $\Sigma = \begin{pmatrix} \Sigma_{WW} & \Sigma_{WV} \\ \Sigma_{WV}^T & \Sigma_{VV} \end{pmatrix}$. The vector $\mu^* = \mathbf{v} \bigcup \mu_{W \mid \mathbf{v}}$ is the mode of a new equilibrium distribution and is therefore the model's prediction for the most likely conformation, after the local perturbation. Significantly, this prediction is computed analytically via matrix operations. Alternatively, one might sample from the conditional distribution $P(\mathbf{W} \mid \mathbf{v}) \sim N(\mu_{W \mid \mathbf{v}}, \Sigma_W)$.

3.2 GAMELAN

The computational and analytical tractability of Gaussian models belies the fact that the conformational dynamics of proteins aren't normally distributed [1, 16]. Thus, while it is always possible to fit a Gaussian to a set of data, sometimes this approximation is valid, and sometimes it is not. In this section, we discuss a framework for creating generative models from conformational ensembles. This technique, called GAMELAN (GrAphical Models of Energy LANdscapes), is capable of producing a variety of generative models (including Gaussian), but it primarily intended for circumstances where the conformational dynamics are non-Gaussian.

GAMELAN produces generative models from a set of data by learning a *Probabilistic Graphical Model* (Fig. 1). Informally, a probabilistic graphical model is a factored encoding of a multivariate distribution, $P(\mathbf{X})$. It consists of a graph defined over the variables, and a set of functions defined over the nodes and edges in the graph. The topology of the graph distinguishes between indirect and direct couplings between random variables. The mathematical form of the functions determines the nature of the distribution. Here, there is a great amount of flexibility; depending on the choice of functions, GAMELAN can produce Gaussian distributions, multinomial distributions, circular distributions, and, most significantly, multi-modal distributions. Multi-modal distributions are essential if the system has more than one conformational substate [9, 10]. It is also possible to define the functions using molecular mechanics force-fields.

Like the simple Gaussian model discussed in Sec. 3.1, the models produced by GAMELAN can perform a variety of tasks efficiently, although not necessarily analytically. In particular, quantifying the difference between models, sampling, and computing conditional distributions are all possible using GAMELAN models. Additionally, if the node and edge functions are defined in terms of force-fields, the model can be used to estimate important quantities, like free energies [17, 18, 19].

3.2.1 Probabilistic Graphical Models—A probabilistic graphical model, $M = (G, \Theta)$, encodes a joint probability distribution $P(\mathbf{X})$ in terms of a graph, $G = (\mathbf{V}, \mathbf{E})$, and a set of functions $\Theta = (\theta_1, \ldots, \theta_p)$ defined over the nodes ($\mathbf{V} = (V_1, \ldots, V_n)$) — one for each random variable —- and edges ($\mathbf{E} \subseteq \mathbf{V} \times \mathbf{V}$) of the graph. For the models that GAMELAN produces, G is undirected (Fig. 1). We note that undirected probabilistic graphical models are often called *Markov Random Fields* in the Machine Learning and Statistics literature.

The probability density encoded by a GAMELAN model is:

$$P(\mathbf{X}) = \frac{1}{Z(\Theta)} \exp\left(\sum_{c \in \mathrm{Cliques}(G)} \theta_c(X_c) \right) \quad (5)$$

where the sum is over the fully connected subgraphs (i.e., cliques) of the graph and $X_C \subseteq \mathbf{X}$ is the subset of variables in clique c.

The topology of G defines the set of *conditional independencies* between the random variables. In particular, the absence of an edge between X_i and X_j means that the two variables are conditionally independent of each other. That is, given its neighbors in the graph, random variable X_i is independent of X_j (and visa-versa).

Informally, the notion of conditional independence means that any observed correlations between X_i and X_j can be explained in terms of the network of couplings in G. Note that the lack of an edge does *not* mean that two random variables are uncorrelated, only that the correlations are due to indirect couplings. By analogy, consider a mass-spring system. The motions of two masses may be correlated, even if they are not directly coupled by a spring. GAMELAN learns a minimal set of edges and the associated parameters that best explain

the correlations observed in the data. We note, however, that the 'springs' in GAMELAN models are not necessarily harmonic.

There are two basic tasks associated with probabilistic graphical models: learning and inference. In general, both tasks are non-trivial, and there are a number of algorithms for solving these problems. Here, we will highlight some of the key concepts, and direct the reader to [23] for more information on these topics, and on graphical models, in general.

3.2.2 Learning—Learning refers to a procedure for estimating the parameters of the model from data. If the topology of the graph is given (i.e., imposed), then learning is generally performed maximizing the log-likelihood of the parameters, given the data: $\Theta^* = \underset{\Theta}{\operatorname{argmax}} ll(\Theta|\mathbf{D};G)$. If the topology of the graph is not known, it can also be learned from the data. This is known as the structure learning problem. Finding the optimal topology and parameters simultaneously is much harder than finding the optimal parameters alone. This is because the number of undirected topologies over *n* variables is $2\binom{n}{2}$. Practical algorithms for solving the structure learning problem place a prior over graph topologies, often in the form of a regularization penalty, which penalizes dense graphs. The intuition behind this penalty is that for every edge that is added to the graph, the parameters associated with the corresponding edge function must be estimated. Highly-parameterized models risk over-fitting the data, and so sparse models are preferred over dense models.

When asked to solve the structure learning problem, GAMELAN optimizes a regularized version of a quantity known as the pseudo log-likelihood:

$$(G,\Theta)^* = \underset{G,\Theta}{\operatorname{argmax}} pll(G,\Theta|\mathbf{D}) - \lambda R(G,\Theta). \quad (6)$$

Here, *pll* is the pseudo log-likelihood of the graph and parameters, given the data, *R* is the regularization function, and λ is a parameter that controls the tradeoff between fitting the data (the first term) and having simple models. The pseudo log-likelihood is a consistent estimator (i.e., given enough data, it converges on the same solution as the exact log-likelihood), and is also much more efficient to compute [3]. The regularization penalty can be defined in a number of ways. GAMELAN uses an L_1 penalty, which encourages sparse graphs. L_1 regularization also has desirable statistical properties. Specifically, it leads to consistent models (that is, given enough data our algorithm learns the true topology) while enjoying high efficiency (that is, the number of samples needed to achieve the true model is small). The regularization parameters, λ, can be set in a number of ways, including AIC and BIC, or through a simple permutation test that finds the value of λ that yields no edges on randomized versions of the data (where all correlations have been eliminated).

Figure 2 illustrates the topology of the network learned by GAMELAN from a 2ns simulation of a complex consisting of gp120 (a glycoprotein on the surface of the HIV envelope), the CD4 receptor (a glycoprotein expressed on the surface of T helper cells) and Ibalizumab, a humanized monoclonal antibody that binds to CD4 and inhibits the viral entry process. The set of edges includes intra- and inter molecular pairs.

3.2.3 Inference—Once the model has been learned, it can be used to perform a variety of tasks. For example, Gibbs sampling, and related procedures, can be used to generate new conformations. Approximate versions of the KL divergence can be computed by calculating an empirical estimate for $KL(M_0 \parallel M_1) = \sum P_{M_0}(i) \log \frac{P_{M_0}(i)}{P_{M_1}(i)}$. Marginal and conditional distributions can be computed using message-passing algorithms on the graph, such as Belief Propagation [34] and Expectation Propagation [29], depending on the nature of the functions (see [23] for a complete discussion).

The remainder of this section discusses the range of models that can be produced by GAMELAN. We start with parametric models, such as the Gaussian and von Mises distribution, and then move on to semi- and non parametric models, which are more expressive.

3.2.4 Parametric Models—The simplest graphical model for continuous-valued random variables is the *Gaussian Graphical Model*, which is defined as the pair (**h**, Σ^{-1}). Here, Σ^{-1} is inverse of the covariance matrix (also known as the *precision or concentration* matrix) and *h* is an *n*-dimensional vector satisfying $\mu = h^T \Sigma$. Thus, a Gaussian Graphical Model can be constructed from the empirical mean and covariance (as in Sec. 3.1). Empirical estimates, however, are subject to over-fitting the data. Therefore, when asked to produce a Gaussian Graphical Model, GAMELAN computes a regularized estimate of the precision matrix (and hence a regularized version of its inverse, the covariance matrix). Specifically, GAMELAN learns a sparse precision matrix (i.e., one with many zeros among the off-diagonal elements) (see [40]). The non-zero elements of Σ^{-1} correspond to the edges in the graphical model.

Notice that unlike PCA-based methods, like Quasi-Harmonic Analysis, which produce Gaussian models after a change of basis, a Gaussian Graphical model is defined over the original variables, $\mathbf{X} = \{X_1, \ldots, X_n\}$. Thus while having a similar form, the resulting models are very different. In particular, PCA-based models encode the joint distribution in terms of *global* motions, since each eigenvector is a linear combination of the original variables. Gaussian Graphical Models, on the other hand, are defined in terms of a network of *local* couplings.

Some quantities are not well modeled using Gaussian variables. Angles, in particular, are defined on the circle, and so are best modeled using circular distributions. The circular analog to the Gaussian distribution is the von Mises distribution [8]. The univariate von Mises distribution over angle $\varphi \in (-\pi, \pi]$ is defined as:

$$P(\phi) = \frac{e^{\kappa \cos(\phi - \mu)}}{2\pi I_0(\kappa)}$$

where $I_0(\kappa)$ is the modified Bessel function of order 0, and the parameters μ and $\frac{1}{\kappa}$ are analogous to μ and σ^2 (the mean and variance) in the Gaussian distribution. κ is known as

the *concentration* of the variable, and so high concentration implies low variance. The bivariate von Mises distribution [28] over $\Phi = (\varphi_1, \varphi_2)$, can be defined as:

$$P(\Phi) = \frac{\exp\{\left[\sum_{i=1}^{2} \kappa_i \cos(\phi_i - \mu_i)\right] + \lambda g(\phi_1, \phi_2)\}}{Z(\mu_1, \mu_2, \kappa_1, \kappa_2, \lambda)},$$

where μ_1 and μ_2 are the means of φ_1 and φ_2, respectively, κ_1 and κ_2 are their corresponding concentrations, $g(\varphi_1, \varphi_2) = \sin(\varphi - \mu_1) \sin(\varphi_2 - \mu_2)$, λ is a measure of the dependence between φ_1 and φ_2, and $Z(\cdot)$ is the normalization constant.

A *von Mises Graphical Model* can be defined using a combination of uni- and bivariate von Mises distributions as the node and edge functions, respectively. GAMELAN can produce von Mises graphical models using existing algorithms for regularized structure learning [39], and inference [38].

Gaussian and von Mises graphical models are most useful when the ensemble being analyzed is well-approximated by a unimodal distribution (e.g., fluctuations within a single conformational substate). For more complex ensembles, spanning more than one conformational substate, or exhibiting substantial asymmetry, these approximations will be poor. An obvious example of a complex distribution is the Ramachandran distribution (Fig. 3).

There are a number of strategies for addressing this problem of non-Gaussian distributed data. One simple-minded solution is to modify existing PCA-based methods (e.g., quasiharmonic analysis) so that they perform Independent Components Analysis, instead. Independent Components Analysis also performs a change of basis, but onto a set of statistically independent bases (as opposed to merely uncorrelated bases, which is what PCA produces). Such modifications are straight-forward, but still define the joint distribution in terms of global motions. To address these same problem using graphical models, there are there are two basic options. The first is to discretize conformation space in some fashion, and then learn a multinomial model. For example, graphical models over discrete backbone or side-chains conformations (i.e., rotamers) have been developed (e.g., [18, 19]), and GAMELAN can construct such models. The second approach is to abandon parametric forms and utilize semior nonparametric graphical models. GAMELAN can produce these models too, as we discuss in the following sections.

3.2.5 Semi-Parametric Models—If the data are not well-approximated via a Gaussian distribution, one option is to apply a function to the data so that the transformed data are (approximately) Gaussian. This is the central idea behind the *nonparanormal* distribution [27]. Formally, random variable $\mathbf{X} = \{X_1, \ldots, X_n\}$ is distributed as a nonparanormal, denoted by $\mathbf{X} \sim NPN(\mu, \Sigma, f)$, if there exist a set of functions $f = \{f_i, \ldots, f_n\}$ such that $f(\mathbf{X}) \sim N(\mu, \Sigma)$. Here, $f(\mathbf{X}) = \{f_1(X_1), \ldots, f_n(X_n)\}$. Under the constraints that the functions are monotone and differentiable, the probability density for any n-vector $\mathbf{x} = \{x_1, \ldots, x_n\}$ is:

$$P(\mathbf{x}) = \frac{1}{Z}\exp\left\{-\frac{1}{2}(f(\mathbf{x}) - \mu)^T\Sigma^{-1}(f(\mathbf{x}) - \mu)\right\}\prod_i |f_i'(x_i)|, \quad (7)$$

where $Z = \sqrt{(2\pi)^n|\Sigma|}$.

GAMELAN learns the parameters of the nonparanormal graphical model (i.e., μ, Σ, and *f*) from the data. Briefly, the *fs* are approximated as: $f_i = \mu_i + \sigma_i h_i(x)$, where μ_i and σ_i are the empirical mean and standard deviation for the *i*th variable, and h_i is the inverse cumulative distribution function applied to the marginal empirical cumulative distribution

$F_i(t) = \frac{1}{m}\sum_j I(X_i(j) \leq t)$ for the *i*th variable. Here, *I* is the indicator function. Any operation that can be performed on a Gaussian (e.g., Sec 3.1) can also be performed on the nonparanormal. The functions, *f*, are invertible, so samples generated in the 'nonparanormal space' can be projected into the real space. Similarly, predictions made by calculating conditional distributions can be inverted.

Figures 4 and 5 demonstrate the nonparanomal. In Fig. 4 a scatter plot of two dimensional data is presented, along with histograms of the marginal distributions. The distribution is non-Gaussian. In Figure 5 the red circles are the same points as in Fig. 4, after the nonparanormal transformation. Notice that the marginals of the transformed data are now Gaussian.

Figure 6 illustrates the differences between making prediction using a Gaussian Graphical Model and a Nonparanormal Graphical Model. Models were fit to the same data as Fig. 4. The left-hand figure shows the predictions made for variable *y*, given different values of variable *x* under both models using Eqn. 3. Similarly, the right-hand figure shows the predictions made for variable *x*, given different values of variable *y*. In each figure, the red line is the prediction made using a Gaussian model and the green line is the prediction made by the Nonparanormal model. Notice that while the Gaussian predictions form a line, the Nonparanormal predictions curve, better reflecting the distribution.

3.2.6 Non-Parametric Models—The Nonparanormal Graphical Model is more expressive than a Gaussian Graphical Model, but there are models which are more expressive than the nonparanormal. GAMELAN provides two options: (i) Hilbert-space embeddings of graphical models, and (ii) mixtures of graphical models.

A *Hilbert space* is a complete vector space endowed with an dot product operation. A *Reproducing kernel Hilbert space* (RKHS), *H*, is a Hilbert space of functions with kernel κ satisfying the reproducing property:

$$f(x) = \langle f(\cdot), k(x, \cdot)\rangle,$$

and thus

$$k(x, x') = \langle k(x, \cdot), k(x', \cdot) \rangle.$$

The significance of RKHS's is that function evaluations can be performed via inner products.

Recently, it has been shown that joint and conditional probability distributions can be embedded into a suitable RKHS [42, 45]. That is, different distributions correspond to different points in the RKHS. The significance of these embeddings is that it becomes possible to efficiently learn non-parametric graphical models and perform inference [43, 44]. The resulting distributions can be real-valued and multimodal.

The second approach to learning non-parametric models is to learn a mixture of graphical models. Here, the data can either be clustered beforehand into microstates, and a separate graphical model learned for each cluster (Gaussian, von Mises, Non-paranormal, or Hilbert-Space), or expectation-maximization can be used to identify the mixtures, and their parameters. Under a mixture model, each component is given a weight, w_i and the probability density for any n-vector = $\{x_1, \ldots, x_n\}$ is: $P(\mathbf{x}) = \prod_i w_i P(\mathbf{x}; G_i, \Theta_i)$, where $\Sigma_i w_i = 1$.

3.2.7 Time-varying, Rection-Coordinate Coupled, and Kinetic Models—

GAMELAN can also be used to learn time-varying models from the data. In particular, if the conformational ensemble has been produced via Molecular Dynamics simulations, it is natural to wonder how the distribution changes over time. This is easily accomplished by learning models from (possibly overlapping) windows of the data. The width of the window is selected based on the timescale of interest. The resulting sequence of graphical models encodes a diffusion process and users may examine how the topology and parameters change over time. Similarly, GAMELAN may be applied to conformations obtained via umbrella sampling (or similar) along a reaction coordinate. The resulting models reveal how the distribution changes along the reaction coordinate (Fig 7).

Alternatively, GAMELAN can be combined with Markov State Models [2, 30, 31, 46, 48] to produce a fully generative, kinetic model. Here, the data are clustered into microstates and the transition rates between states is estimated from the data. A graphical model is learned for each state, to facilitate sampling and inference. Unlike the time-varying model, Markov State Models are jump-processes.

4 Example

To illustrate the predictive accuracy of different models, GAMELAN was applied to data from a 50 μ*s* simulation of the engrailed homeodomain (Fig 8-left). We extracted the θ - τ angles from the data, which describes the configuration of the alpha carbons (Fig. 8-right). The data was partitioned into training-test splits. The training data were used to learn a Gaussian, von Mises, Nonparanormal, and Hilbert- Space Graphical models. Next, using the test data we conditioned each model on a random subset of the variables and predicted the

values of the remaining variables. Table 1 demonstrates that the von Mises Graphical model gives the lowest errors.

5 Discussion and Conclusion

Probabilistic Graphical Models of protein structures were first introduced in 2002 by Yanover and Weiss [51], who focused on predicting side chain configurations. Subsequent uses of graphical models for proteins have considered a wide range of problems, including density fitting [7], structure prediction [4, 14], protein design [12, 11, 3, 33, 49], free energy calculations [18], and predicting resistance mutations [20]. Their growing popularity in structural biology is due to their ability to represent complex distributions and solve challenging inference problems.

GAMELAN is the first graphical model specifically designed to aid in the analysis and modeling of conformational ensembles generated through simulation. The extreme complexity of the resulting distributions has necessitated the development of more expressive models. The Hilbert-Space embeddings presently represent the most powerful generative models of protein structure. Applying these models to application domains such as structure prediction and protein design is part of ongoing research. Other challenges include the development of graphical model based simulation algorithms where the model evolves in time along a reaction coordinate, generating conformations along the way.

Acknowledgments

This work is supported in part by US NSF grant IIS-0905193, US NIH RC2GM093307, and US NIH P41 GM103712.

References

1. Amadei A, Linssen ABM, Berendsen HJC. Essential dynamics of proteins. Proteins: Structure, Function, and Bioinformatics. 1993; 17(4):412–425. URL http://dx.doi.org/10.1002/prot.340170408.
2. Andrec M, Felts AK, Gallicchio E, Levy RM. Protein folding pathways from replica exchange simulations and a kinetic network model. Proceedings of the National Academy of Sciences of the United States of America. 2005; 102(19):6801–6806. URL http://www.pnas.org/content/102/19/6801.abstract. [PubMed: 15800044]
3. Balakrishnan S, Kamisetty H, Carbonell J, Lee S, C.J L. Learning Generative Models for Protein Fold Families. Proteins: Structure, Function, and Bioinformatics. 2011; 79(6):1061–1078.
4. Boomsma W, Mardia KV, Taylor CC, Ferkinghoff-Borg J, Krogh A, Hamelryck T. A generative, probabilistic model of local protein structure. Proc. Natl. Acad. Sci. 2008; 105(26):8932–8937. [PubMed: 18579771]
5. Bowers, KJ.; Chow, E.; Xu, H.; Dror, RO.; Eastwood, MP.; Gregersen, BA.; Klepeis, JL.; Kolossvary, I.; Moraes, MA.; Sacerdoti, FD.; Salmon, JK.; Shan, Y.; Shaw, DE. SC '06: Proceedings of the 2006 ACM/IEEE conference on Supercomputing. New York, NY, USA: ACM; 2006. Scalable algorithms for molecular dynamics simulations on commodity clusters; p. 84-96.URL http://dx.doi.org/10.1145/1188455.1188544
6. Brooks B, Karplus M. Harmonic dynamics of proteins: normal modes and fluctuations in bovine pancreatic trypsin inhibitor. Proceedings of the National Academy of Sciences. 1983; 80(21):6571–6575. URL http://www.pnas.org/content/80/21/6571.abstract.
7. DiMaio F, Soni A, Phillips G Jr, Shavlik J. Creating all-atom protein models from electron-density maps using particle-filtering methods. Bioinformatics. 2007; 23:2851–2858. [PubMed: 17933855]

8. Fisher, N. Statistical Analysis of Circular Data. Cambridge University Press; 1993.
9. Frauenfelder H, Parak F, Young RD. Conformational substates in proteins. Ann. Rev. Biophys. Biophys. Chem. 1988; 17:451–479. [PubMed: 3293595]
10. Frauenfelder H, Petsko GA, Tsernoglou D. Temperature-dependent x-ray diffraction as a probe of protein structural dynamics. Nature. 1979; 280(5723):558–563. [PubMed: 460437]
11. Fromer M, Yanover C. Accurate prediction for atomic-level protein design and its application in diversifying the near-optimal sequence space. Proteins: Structure, Function, and Bioinformatics. 2008 p. In Press.
12. Fromer M, Yanover C. A computational framework to empower probabilistic protein design. Bioinformatics. 2008; 24(13):i214–i222. [PubMed: 18586717]
13. Go N, Noguti T, Nishikawa T. Dynamics of a small globular protein in terms of low-frequency vibrational modes. Proceedings of the National Academy of Sciences. 1983; 80(12):3696–3700. URL http://www.pnas.org/content/80/12/3696.abstract.
14. Harder T, Boomsma W, Paluszewski M, Frellsen J, Johansson K, Hamelryck T. Beyond rotamers: a generative, probabilistic model of side chains in proteins. BMC Bioinformatics. 2010; 11(1):306. URL http://www.biomedcentral.com/1471-2105/11/306. [PubMed: 20525384]
15. Harvey M, Giupponi G, Fabritiis G. Acemd: accelerating biomolecular dynamics in the microsecond time scale. Journal of Chemical Theory and Computation. 2009; 5(6):16321639.
16. Hayward S, Kitao A, Go N. Harmonicity and anharmonicity in protein dynamics: a normal mode analysis and principal component analysis. Proteins. 1995; 23(2):177–186. URL http://dx.doi.org/10.1002/prot.340230207. [PubMed: 8592699]
17. Kamisetty H, Bailey-Kellogg C, Langmead C. A Graphical Model Approach for Predicting Free Energies of Association for Protein-Protein Interactions Under Backbone and Side-Chain Flexibility. Proc. Structural Bioinformatics and Computational Biophysics (3DSIG). 2009:67–68.
18. Kamisetty H, Ramanathan A, Bailey-Kellogg C, Langmead C. Accounting for conformational entropy in predicting bidning free energies of protein-protein interactions. Proteins. 2011; 79(2): 444–462. [PubMed: 21120864]
19. Kamisetty H, Xing E, Langmead C. Free Energy Estimates of All-atom Protein Structures Using Generalized Belief Propagation. J. Comp. Bio. 2008; 15(7):755–766.
20. Kamisetty, H.; Xing, E.; Langmead, C. Approximating Correlated Equilibria using Relaxations on the Marginal Polytope; Proc. of the 28th International Conference on Machine Learning (ICML); 2011. p. 1153-1160.
21. Karplus M, Kushick JN. Method for estimating the configurational entropy of macromolecules. Macromolecules. 1981; 14(2):325–332.
22. Karplus M, McCammon JA. Molecular dynamics simulations of biomolecules. Nature Structural Biology. 2002; 9(9)
23. Koller, D.; Friedman, N. Probabilistic Graphical Models: Principles and Techniques. MIT Press; 2009.
24. Landau, D.; Binder, K. A Guide to Monte Carlo Simulations in Statistical Physics. New York, NY, USA: Cambridge University Press; 2005.
25. Levitt M, Sander C, Stern PS. The normal modes of a protein: Native bovine pancreatic trypsin inhibitor. International Journal of Quantum Chemistry. 1983; 24(S10):181–199. URL http://dx.doi.org/10.1002/qua.560240721.
26. Levy RM, Srinivasan AR, Olson WK, McCammon JA. Quasi-harmonic method for studying very low frequency modes in proteins. Biopolymers. 1984; 23:1099–1112. [PubMed: 6733249]
27. Liu H, Lafferty J, Wasserman L. The nonparanormal: Semiparametric estimation of high dimensional undirected graphs. J. Mach. Learn. Res. 2009; 10:2295–2328. URL http://dl.acm.org/citation.cfm?id=1577069.1755863.
28. Mardia KV. Statistics of directional data. J. Royal Statistical Society. Series B. 1975; 37(3):349–393.
29. Minka TP. Expectation propagation for approximate bayesian inference. Uncertainty in Artificial Intelligence. 2001:362–369.

30. Pan AC, Roux B. Using path sampling to build better Markovian state models: Predicting the folding rate and mechanism of a tryptophan zipper beta hairpin. JOURNAL OF CHEMICAL PHYSICS. 2004; 121(1):415–425. [PubMed: 15260562]
31. Pan AC, Roux B. Building Markov state models along pathways to determine free energies and rates of transitions. JOURNAL OF CHEMICAL PHYSICS. 2008; 129(6)
32. Pande VS, Baker I, Chapman J, Elmer SP, Khaliq S, Larson SM, Rhee YM, Shirts MR, Snow C, Sorin EJ, Zagrovic B. Atomistic protein folding simulations on the sub-millisecond time scale using worldwide distributed computing. Biopolymers. 2003; 68(1):91–109. [PubMed: 12579582]
33. Parker, AS.; Griswold, KE.; Bailey-Kellogg, C. Structure-guided deimmunization of therapeutic proteins; Proceedings of the 16th Annual international conference on Research in Computational Molecular Biology; 2012. p. 184-198.
34. Pearl J. Fusion, propagation, and structuring in belief networks. Artif. Intell. 1986; 29(3):241–288.
35. Phillips J, Braun R, Wang W, Gumbart J, Tajkhorshid E, Villa E, Chipot C, Skeel R, Kale L, Schulten K. Scalable molecular dynamics with namd. J. Comp. Chem. 2005; 26:1781–1802. [PubMed: 16222654]
36. Ramanathan A, Agarwal PK, Kurnikova M, Langmead C. An Online Approach for Mining Collective Behaviors from Molecular Dynamics Simulations. J. Comp. Biol. 2010; 17(3):309–324.
37. Ramanathan A, Yoo J, Langmead C. On-the-fly Identification of Conformational Sub-states from Molecular Dynamics Simulations. Journal of Chemical Theory and Computation. 2011; 7(3):778–789.
38. Razavian, N.; Kamisetty, H.; Langmead, C. Tech. Rep. CMU-CS-11-108. Carnegie Mellon University, Department of Computer Science; 2011. The von mises graphical model: Expectation propagation for inference.
39. Razavian, N.; Kamisetty, H.; Langmead, C. Tech. Rep. CMU-CS-11-108. Carnegie Mellon University, Department of Computer Science; 2011. The von mises graphical model: Regularized structure and parameter learning.
40. Razavian N, Kamisetty H, Langmead C. Learning generative models of molecular dynamics. BMC Genomics. 2012; 13(Suppl 1)
41. Shaw DE, Deneroff MM, Dror RO, Kuskin JS, Larson RH, Salmon JK, Young C, Batson B, Bowers KJ, Chao JC, Eastwood MP, Gagliardo J, Grossman JP, Ho CR, Ierardi DJ, Kolossváry I, Klepeis JL, Layman T, McLeavey C, Moraes MA, Mueller R, Priest EC, Shan Y, Spengler J, Theobald M, Towles B, Wang SC. Anton, a special-purpose machine for molecular dynamics simulation. SIGARCH Comput. Archit. News. 2007; 35:1–12.
42. Smola, A.; Gretton, A.; Song, L.; Schölkopf, B. Algorithmic Learning Theory. Springer; 2007. A hilbert space embedding for distributions. Invited paper
43. Song, L.; Gretton, A.; Bickson, D.; Low, Y.; Guestrin, C. Kernel belief propagation; International Conference on Artifical Intelligence and Statistics (AISTATS); 2011.
44. Song L, Gretton A, Guestrin C. Nonparametric tree graphical models. Artificial Intelligence and Statistics (AISTATS). 2010
45. Song, L.; Huang, J.; Smola, A.; Fukumizu, K. Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09. New York, NY, USA: ACM; 2009. Hilbert space embeddings of conditional distributions with applications to dynamical systems; p. 961-968.URL http://doi.acm.org/10.1145/1553374.1553497
46. Sriraman S, Kevrekidis IG, Hummer G. Coarse master equation from bayesian analysis of replica molecular dynamics simulations. The Journal of Physical Chemistry B. 2005; 109(14):6479–6484. URL http://pubs.acs.org/doi/abs/10.1021/jp046448u. PMID: 16851726. [PubMed: 16851726]
47. Stone J, Phillips JC, Freddolino PL, Hardy DJ, Trabuco LG, Schulten K. Accelerating molecular modeling applications with graphics processors. J. Comp. Chem. 2007; 28:2618–2640. [PubMed: 17894371]
48. Swope WC, Pitera JW, Suits F. Describing protein folding kinetics by molecular dynamics simulations. 1. theory. The Journal of Physical Chemistry B. 2004; 108(21):6571–6581. URL http://pubs.acs.org/doi/abs/10.1021/jp037421y.

49. Thomas J, Ramakrishnan N, Bailey-Kellogg C. Protein design by sampling an undirected graphical model of residue constraints. IEEE/ACM Trans. Comput. Biol. Bioinformatics. 2009; 6(3):506–516.

50. Tirion MM. Large amplitude elastic motions in proteins from a single-parameter, atomic analysis. Phys. Rev. Lett. 1996; 77:1905–1908. URL http://link.aps.org/doi/10.1103/PhysRevLett.77.1905. [PubMed: 10063201]

51. Yanover C, Weiss Y. Approximate inference and protein folding. Advances in Neural Information Processing Systems (NIPS). 2002:84–86.

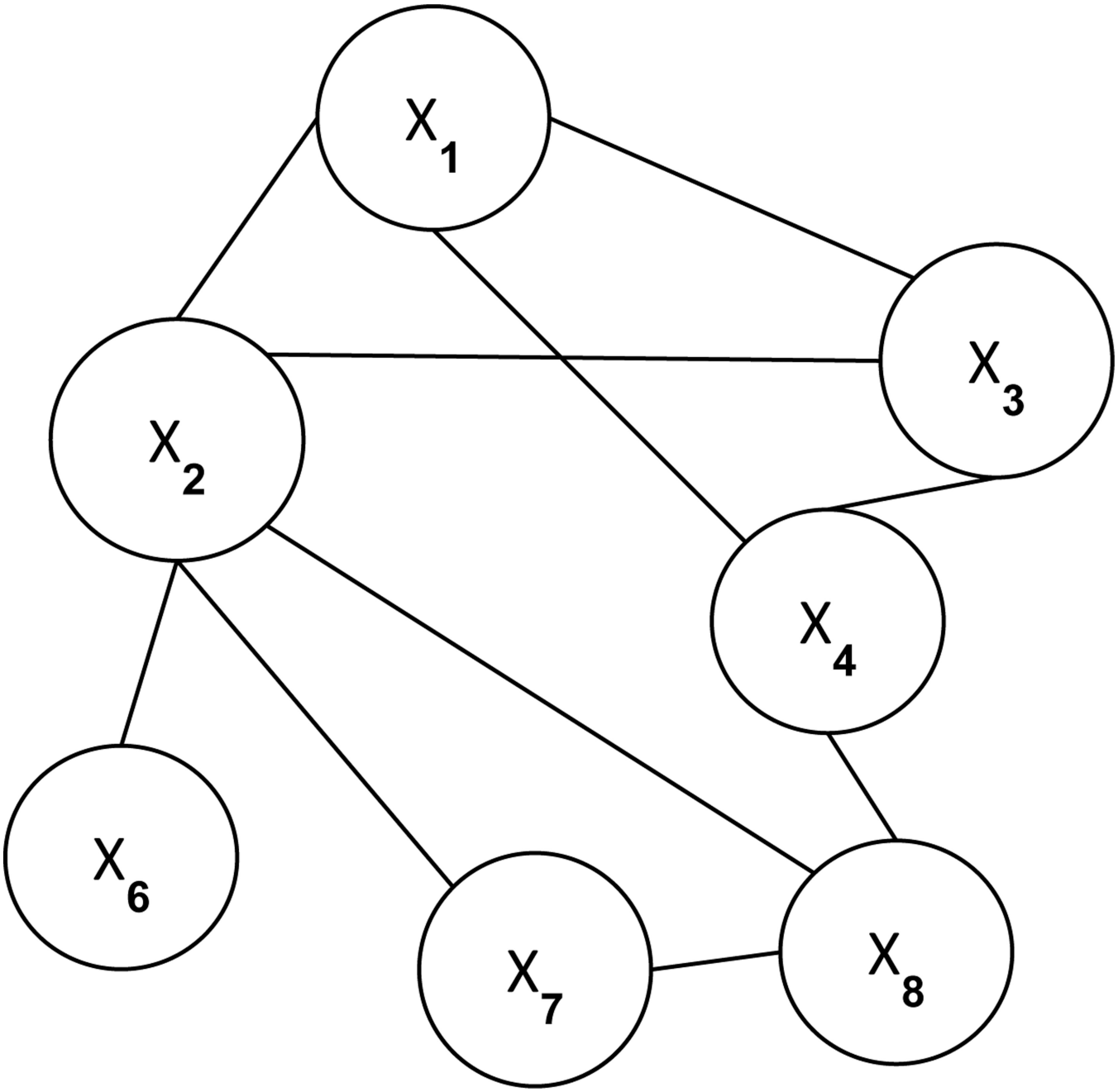

Fig. 1.

A probabilistic graphical model over eight random variables. Nodes correspond to random variables. Edges reveal the conditional independencies among the variables. Each node and edge is associated with a function. When combined, the graph and the functions encode the joint probability distribution over the variables, $P(X_1, \ldots, X_8)$. Graphical models of protein structures may have hundreds or thousands of nodes, depending on which covariates are being modeled.

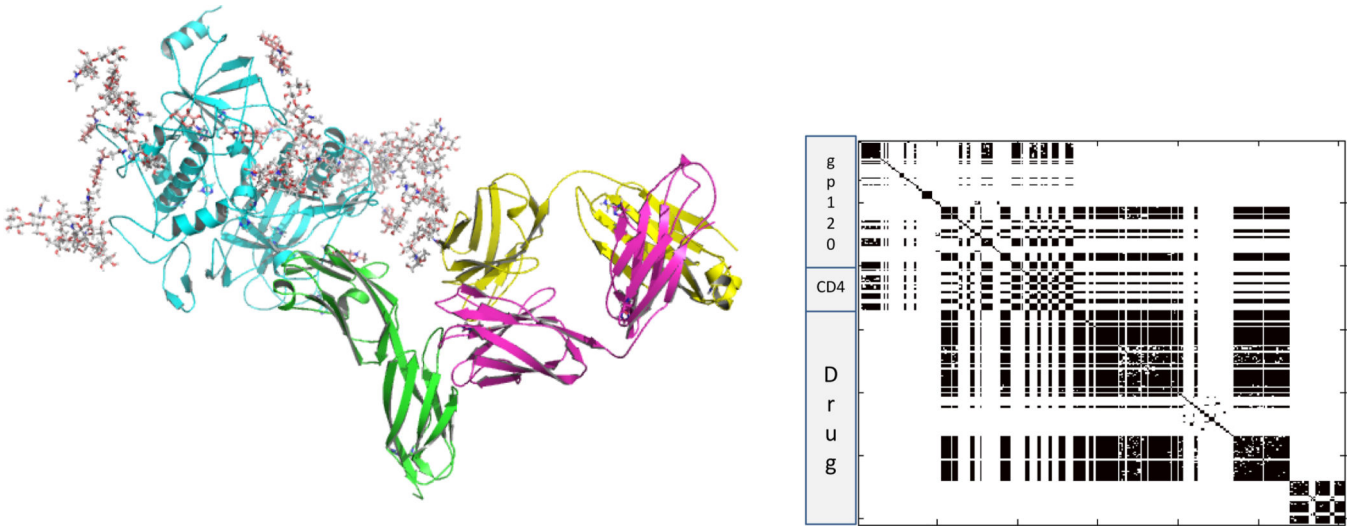

Fig. 2.

Left: A complex consisting of gp120 (cyan), CD4 (green), and Ibalizumab (magenta and yellow). Right: The topology of the graphical model learned by GAMELAN. A black dot means there is an edge between residues *i* and *j*.

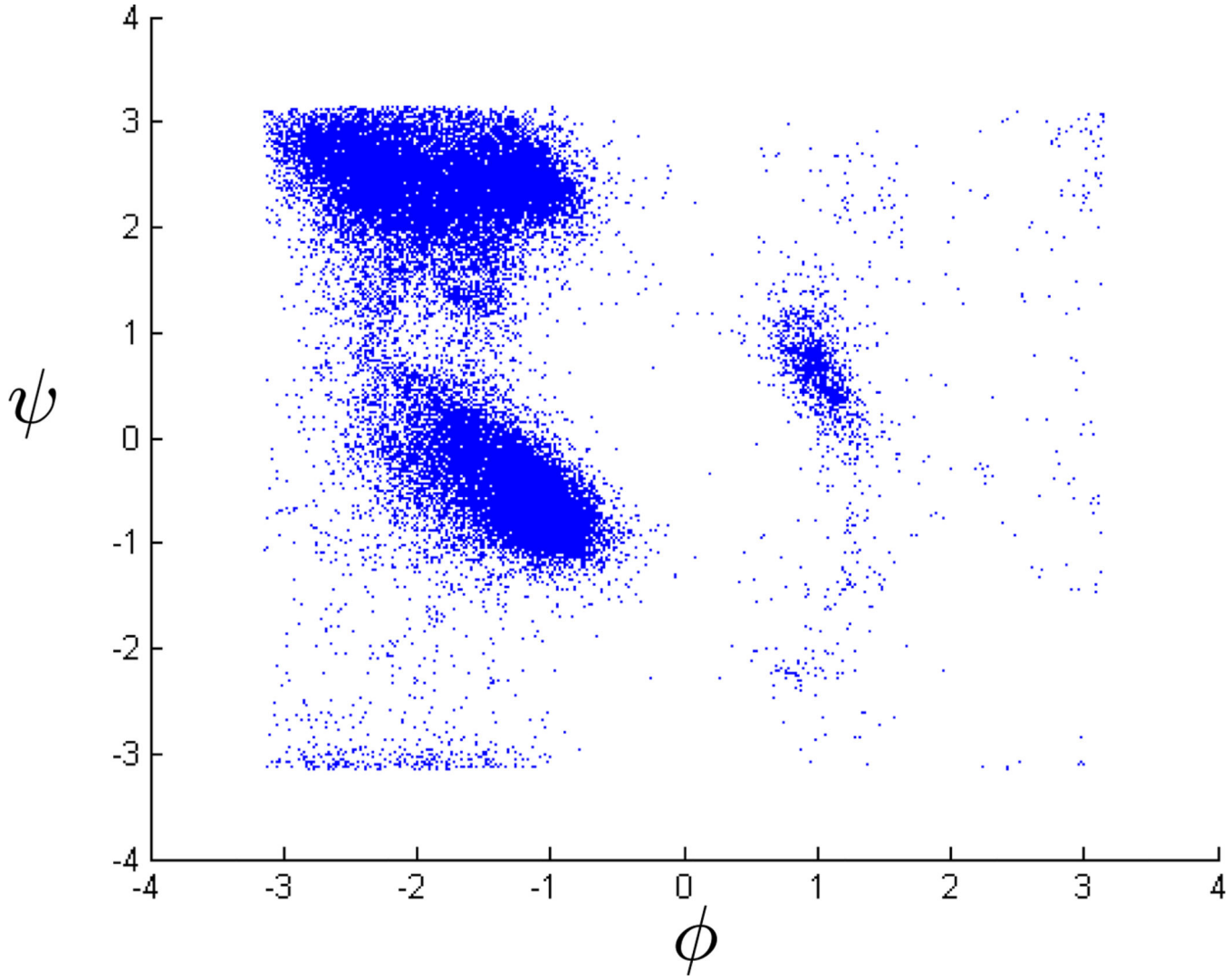

Fig. 3.
Ramachandran Plot. Axes are in radians.

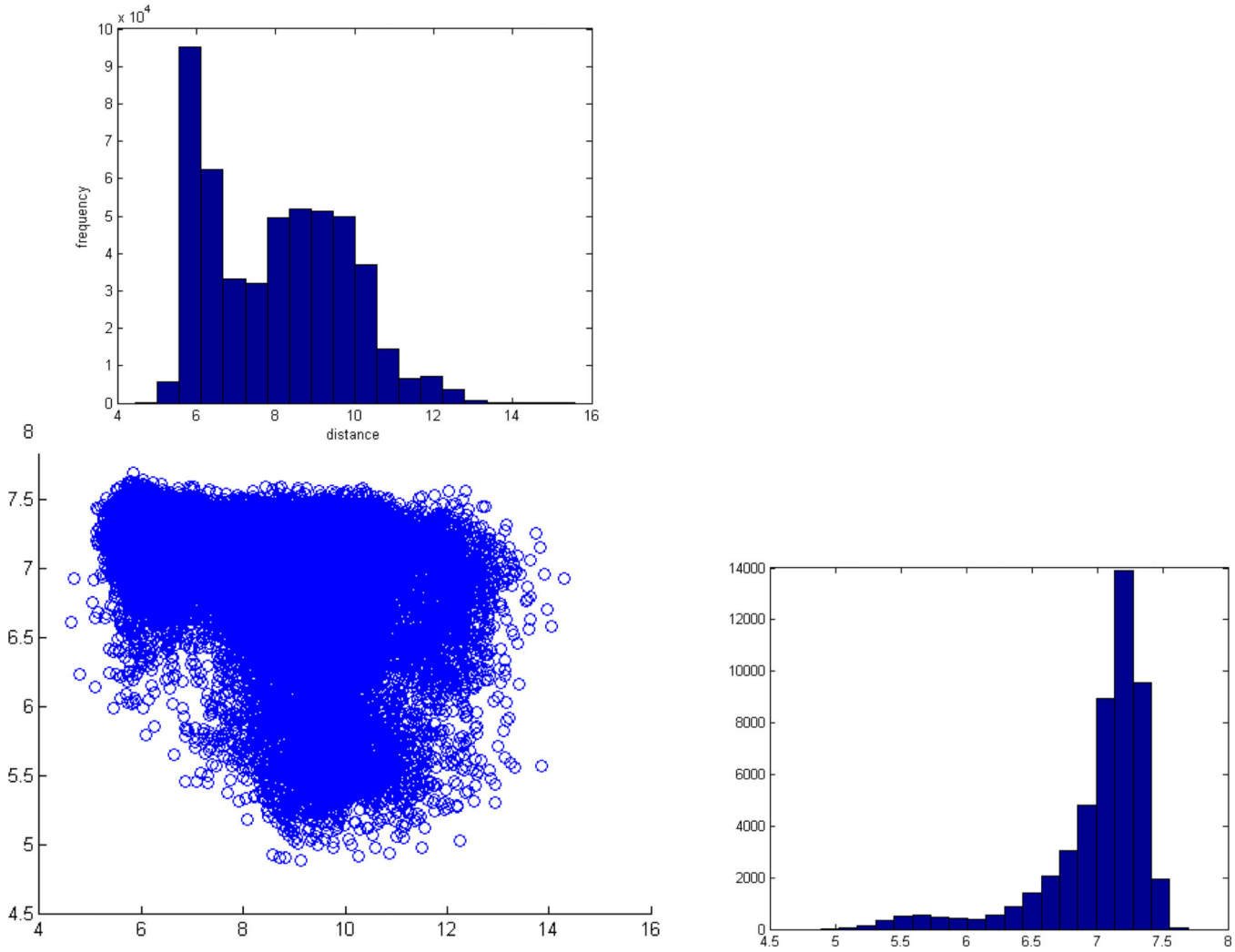

Fig. 4.

Scatter plot of two dimensional data and histograms of the marginal distributions.

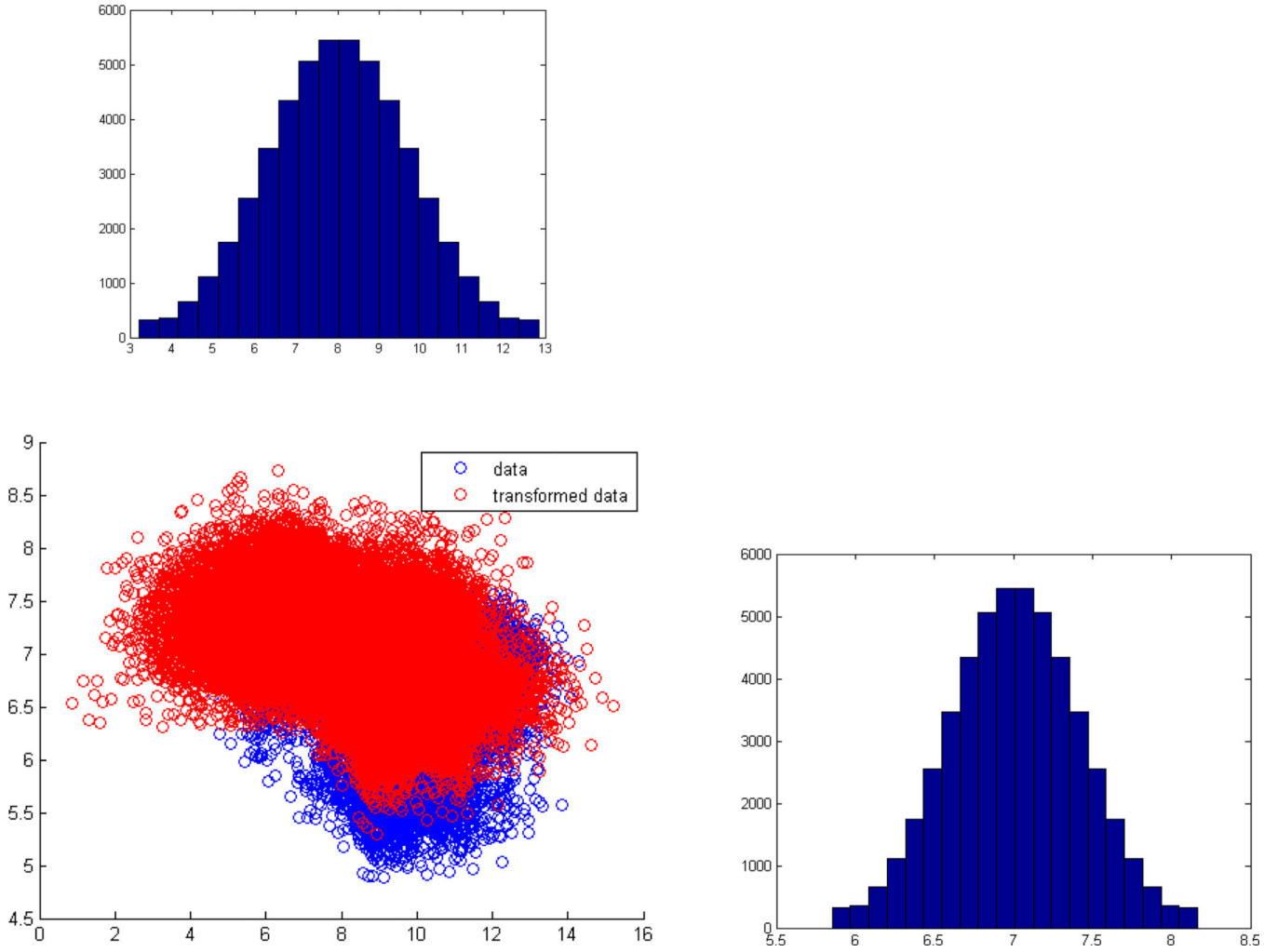

Fig. 5.

The red points are the data from Fig. 4 in the ‘nonparanormal space’. Notice that the marginal distributions are now Gaussian.

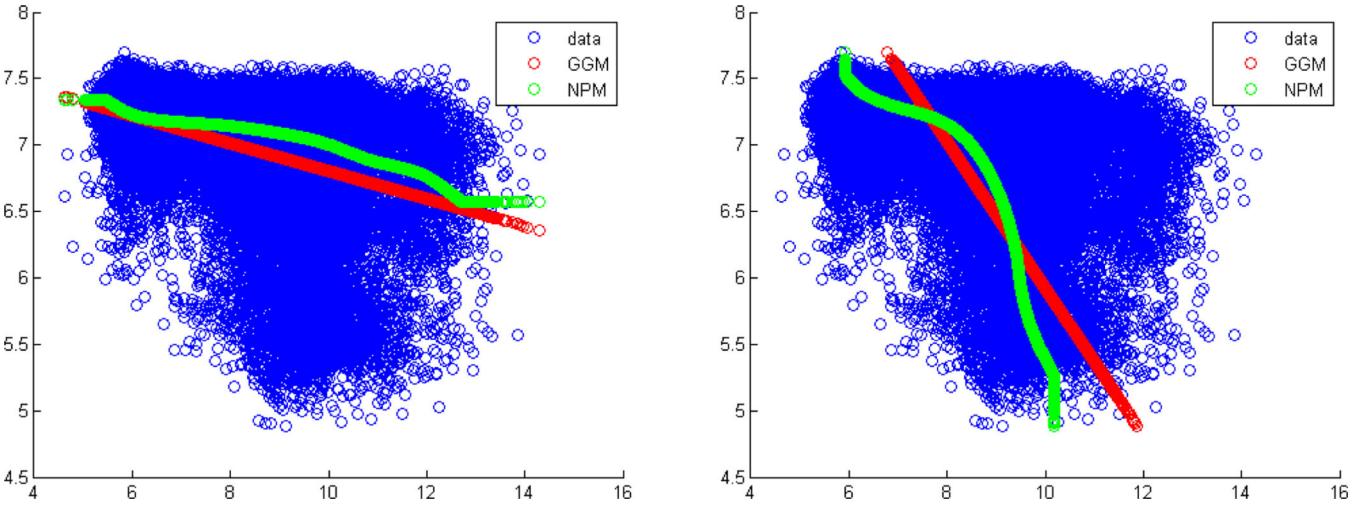

Fig. 6.

Left: The lines show the maximum likelihood value for variable *y*, for different values of *x*. Right: The lines show the maximum likelihood value for variable *x*, for different values of *y*. The red line is computed using a Gaussian Graphical Model. The green line is computed using a Nonparanormal Graphical Model.

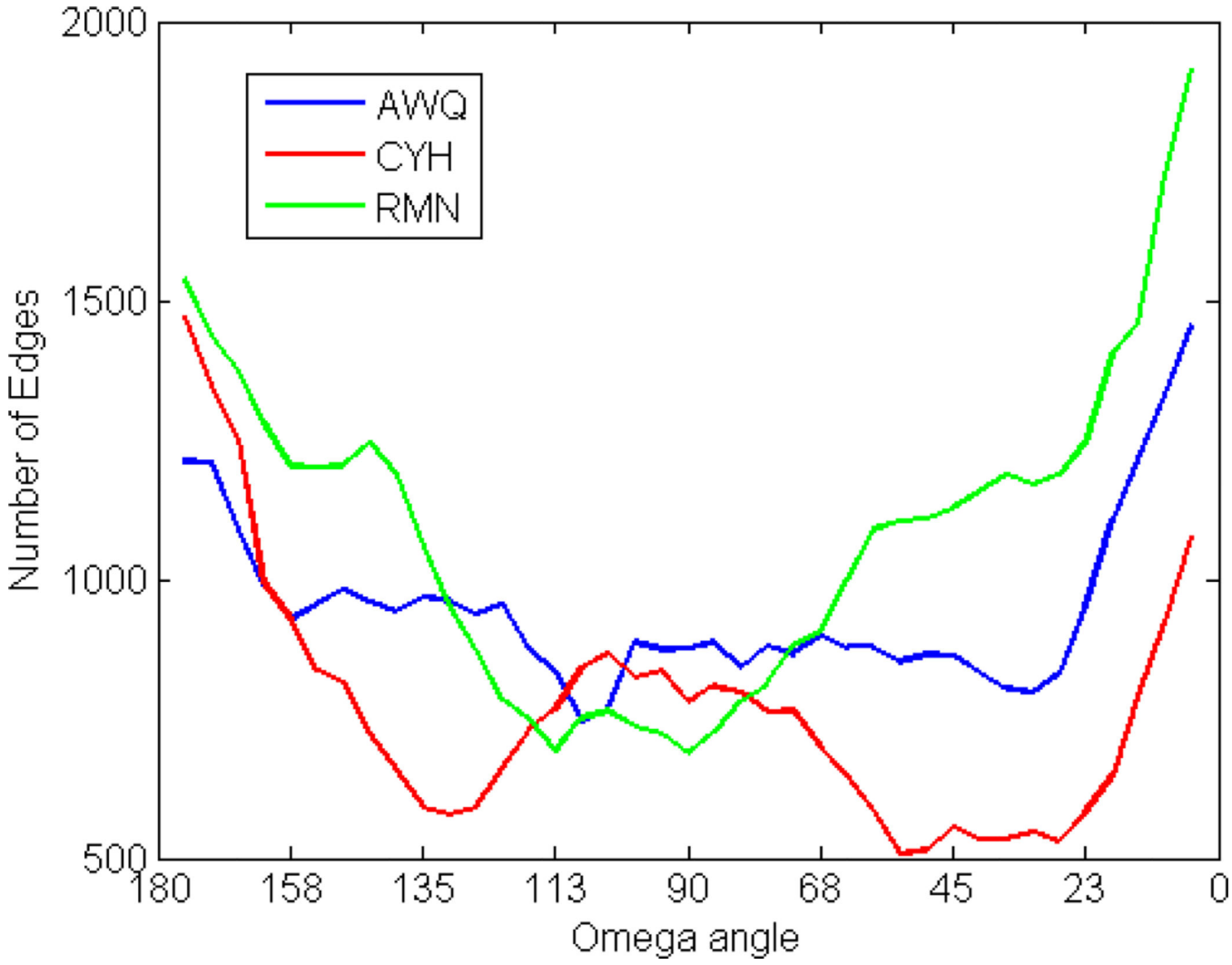

Fig. 7.

The enzyme cyclophilin A isomerizes the omega angle of its substrate. Here, the number of edges learned by GAMELAN is plotted against the reaction coordinate for three substrates.

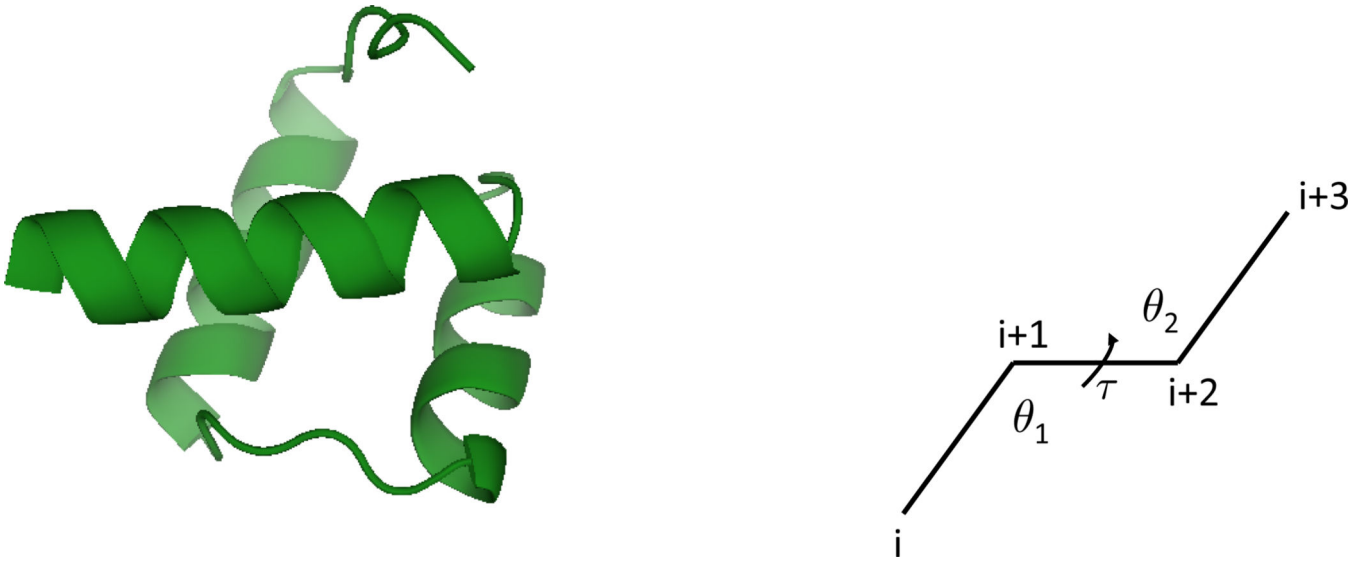

Fig. 8.

Left: The engrailed homeodomain. Right: θ - τ representation of the backbone is defined over the alpha carbons.

Table 1

Predictive accuracy on angular data

Model	Root Mean Square Error (degrees)
Gaussian	8.5
von Mises	6.9
Nonparanormal	8.4
Hilbert-Space	7.3