Residue-Specific Side-Chain Polymorphisms via Particle Belief Propagation

Journal:	*IEEE/ACM Transactions on Computational Biology and Bioinformatics*
Manuscript ID:	TCBBSI-2013-10-0294
Manuscript Type:	SI: GIW 2013
Keywords:	conformational ensemble, conformational polymorphism, mixture distribution, particle belief propagation, side-chain prediction, von-Mises distribution

Residue-Specific Side-Chain Polymorphisms via Particle Belief Propagation

Laleh Soltan Ghoraie, Forbes Burkowski, Shuai Cheng Li, Mu Zhu

Abstract—Protein crystals populate diverse conformational ensembles. Despite much evidence that there is widespread conformational polymorphism in protein side chains, most of the X-ray crystallography data are modeled by single conformations in the Protein Data Bank. The ability to extract or to predict these conformational polymorphisms is of crucial importance, as it facilitates deeper understanding of protein dynamics and functionality. In this article, we describe a computational strategy capable of predicting side-chain polymorphisms. Our approach extends a particular class of algorithms for side-chain prediction by modeling the side-chain dihedral angles more appropriately as continuous rather than discrete variables. Employing a new inferential technique known as particle belief propagation, we predict residue-specific distributions that encode information about side-chain polymorphisms. Our predicted polymorphisms are in relatively close agreement with results from a state-of-the-art approach based on X-ray crystallography data, which characterizes the conformational polymorphisms of side chains using electron density information, and has successfully discovered previously unmodeled conformations.

Index Terms—conformational ensemble; conformational polymorphism; mixture distribution; particle belief propagation; side-chain prediction; von-Mises distribution

1 Introduction

Due to the wide range of its motions, e.g., as shown by many studies using nuclear magnetic resonance (NMR) spectroscopy [23], [5], [41], a protein molecule can appear in many different conformations [13], [49]. As a result, it is insufficient to describe a protein molecule by a single model [34], [35]. One idea is to model the structure of such dynamic molecules as proteins more properly with conformational ensembles [3]. Capturing alternate conformations of a protein is of crucial importance for many applications, e.g., drug design, understanding disease mechanisms, etc; undoubtedly, doing so will bring crucial insight as well to deepen our understanding of how proteins fold, function, and bind to ligands [48], [14]. An important step in this direction is the ability to predict and describe the conformational polymorphism of each residue.

Since most residues belonging to structures in the Protein Data Bank (PDB; http://www.rcsb.org/pdb/) [2] are modeled by a single side-chain conformation, the majority of computational approaches for making side-chain predictions have focused on finding a single "best" conformation (more on this in Section 2 below). However, a few recent studies have started to reinvestigate crystallographic data, and to explore the phenomenon of side-chain polymorphism.

For example, van den Bedem et al. [47] developed a method to identify and model the conformational heterogeneity of proteins from electron density data. The output of their method is a co-called "multi-conformer model", or "an occupancy-weighted set of main-chain and side-chain conformations that collectively best represents the electron density" [47]. The word "occupancy" refers to the relative frequency of occurrence for each conformer in the crystal. The method first generates, in a sampling step, a large set of candidate conformations. In a subsequent selection step, the method fits the occupancies of this set of samples to the electron density map.

Recently, the Alber Lab at the University of California, Berkeley released a program called Ringer [29], which investigates side-chain conformational polymorphisms by sampling the electron density maps around the side-chain dihedral angles of each residue below the usual "1.0 sigma" threshold. Using Ringer, they uncovered evidence suggesting the presence of alternate, hitherto-unmodeled side-chain conformations, many of which are characterized by weak electron density features that were traditionally overlooked when building 3D models of proteins. They showed that their newly identified conformers are nonrandom and are biased towards low-energy rotational isomers. They also discovered, e.g., in Calmodulin, alternate side-chain conformations "not only on the surface but also within the structure" [40], where the protein is tightly packed and side-chain polymorphisms were rarely expected.

1.1 Our Contribution

We have developed a computational approach capable of predicting and describing side-chain polymorphisms — one that does *not* require experimental

L. S. Ghoraie (email: lsoltang@uwaterloo.ca), F. Burkowski and M. Zhu are with University of Waterloo, Canada; their research is supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada. S. C. Li is with City University of Hong Kong, Hong Kong; his research is supported by GRF/ECS Grant No. 9041805 [CityU 124512].

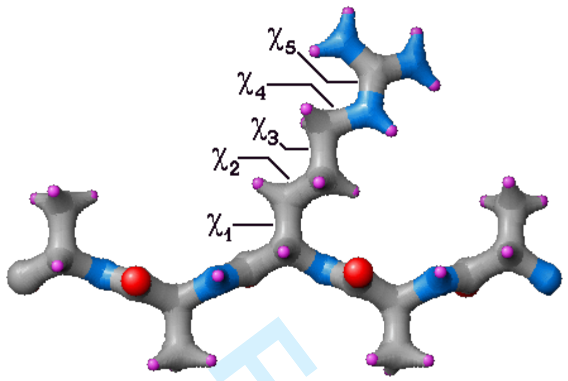


Fig. 1. Illustration of dihedral angles, courtesy of http://www.ccp14.ac.uk/ccp/web-mirrors/garlic/garlic/commands/dihedrals.html.

inputs such as electron density maps. Our approach is an extension of a particular class of algorithms for side-chain prediction that are based on belief propagation (BP) [43].

The conformation of a protein side chain can be parameterized by a sequence of dihedral angles (Fig. 1). Each side chain may rotate flexibly about its dihedral angles, as long as there are no steric collisions. These dihedral angles are continuous in nature, but most computational approaches discretize them. Our primary extension was to model these dihedral angles more appropriately as continuous variables rather than discrete ones. Straight-forward as such an extension may sound, it would have remained difficult within the BP framework if a variation called "particle belief propagation" (PBP) [22] had not become available.

Using PBP, we were able to make inferences about residue-specific distributions in the continuous domain, and it is clear that these distributions encode information about the conformational polymorphism of each residue. We then compared the polymorphisms that we predicted with the ones extracted from crystallography data by Ringer [29]. Overall, we found the two sets of results to be in reasonably good agreement with each other. Ringer has successfully uncovered side-chain conformations that were formerly considered mere artifacts of (or noise from) electron density data. While Ringer found these alternate conformations by re-evaluating electron density maps, we can predict them with an improved — and, in fact, fundamentally different — side-chain prediction algorithm, one that works in a continuous domain.

1.2 Outline

The rest of this article is organized as follows. In Section 2, we quickly review computational approaches for side-chain prediction. In Section 3, we describe our inference method. We first give a very brief review of belief propagation (Section 3.1), and then describe a recent variation, called particle belief propagation, that drives our main algorithm (Section 3.2). In Section 4, we review the von-Mises (VM) distribution for angular data, and explain how we have used mixtures of VM distributions to speed up our algorithm. In Section 5, we report some empirical experiments and their results. Finally, in Section 6, we summarize our main contributions and discuss some work that we have in mind for the future.

2 Review of Side-chain Prediction

To render the analysis more tractable, it is often assumed that the protein backbone is fixed (the dihedral angles in the backbone will not change). With these constraints in place, the side-chain prediction problem concerns finding a conformation for each residue so that the entire molecule achieves the lowest-energy configuration. Since each side-chain conformation is parameterized by a sequence of dihedral angles, the search space of the optimization problem is the infinite set of points with each point being a vector having components that represent the settings for all possible dihedral angles for all residues.

While our goal is to find this optimal solution by using strategies that recognize the continuity of the changes in the dihedral angles, many current computational approaches have reduced the problem to a combinatorial search problem by discretizing the allowed settings of the dihedral angles in the residues. This strategy capitalizes on the phenomenon of rotamericity. Even though a side chain has an infinite number of possible three dimensional conformations, it has been observed that a side chain will typically have a tendency to adopt a conformation that can be approximated by dihedral angles chosen from a small set of empirically observed settings. Each such possible conformation is called a rotamer. A rotamer library contains a discrete set of conformations for each residue type. For instance, the backbone-dependent rotamer library provided by the Dunbrack Lab [11] has been used by many researchers. Algorithms relying on these rotamer libraries essentially apply different heuristics to search for the optimal combination of rotamers, one for each residue. This combinatorial optimization problem is well-known to be NP-hard [1]. An exhaustive search is almost never possible. Many different heuristics have been proposed, such as dead-end elimination [9], [17], simulated annealing [30], and Monte Carlo techniques [19], [21], among many others.

A state-of-the-art heuristic is the SCWRL algorithm [4]. The main steps in SCWRL 3.0 [6] are as follows: first, a dead-end elimination procedure is applied to reduce the number of candidate rotamers for each residue; next, a graph is created by treating residues as nodes and by drawing edges between all nearby residues; then, the graph is clustered into many bi-connected components; finally, the optimization problem is solved separately on each subgraph, before the solutions are combined.

Another highly competitive heuristic is the TreePack algorithm [51]. It also models the protein molecule as a graph. However, loops are removed and the graph is decomposed into clusters and modeled as a tree. The problem of assigning an optimal rotamer to each residue is then solved efficiently by traversing the tree. TreePack is as accurate as, but significantly faster than, SCWRL 3.0.

Other standard optimization techniques such as linear programming (LP) and integer programming (IP) also have been applied to solve the side-chain prediction problem. Yanover and Weiss showed that finding the minimum energy configuration of a protein's side chains is equivalent to finding the maximum-a-posteriori (MAP) configuration of an undirected graphical model, or a Markov random field (MRF) [52], [54]. This meant the side-chain prediction problem could be formulated as a MAP estimation problem, which could be solved using belief propagation (BP). They also considered a relaxed version of the IP problem and solved the resulting convex problem with BP [53].

Besides these optimization approaches, Li et al. [31] recently showed that side-chain conformations also can be decided from backbone information without optimization.

3 Main Method of Inference

We have extended the class of side-chain prediction algorithms that are based on BP. To model a protein molecule with a graphical model, the backbone is regarded as being fixed and the residues $r_1, r_2, ..., r_n$ are regarded as nodes. The side chain at each node is described by a sequence of dihedral angles, collectively stored as a vector, e.g., $r_i = (\chi_{i1}, \chi_{i2}, ..., \chi_{i4})$. The exact number of dihedral angles depends on the type of amino acid. The objective is to find the minimal-energy conformation,

$$\min_{r_1, r_2, ..., r_n} \left[\sum_{i}^{n} E_l(r_i) + \sum_{i}^{n} \sum_{j>i} E_p(r_i, r_j) \right], \qquad (1)$$

where E_l is the (local) intrinsic energy of a residue, E_p is the pairwise energy between two residues, and n is the total number of residues. The optimization algorithm itself is independent of the choice of the energy function. We will say more about the energy function later in section 4.3.

Consider a graphical model $\mathcal{G}$, with a set of vertices $\mathcal{V}$ and a collection of edges $\mathcal{E}$. If r_i denotes the random variable associated with node i, then the joint probability distribution of $\mathbf{r} = (r_1, r_2, ..., r_n)$ can be factorized as follows:

$$P(\mathbf{r}) = \prod_{i \in \mathcal{V}} \rho_i(r_i) \prod_{(i,j) \in \mathcal{E}} \rho_{ij}(r_i, r_j). \qquad (2)$$

The functions, $\rho_i(r_i)$ and $\rho_{ij}(r_i, r_j)$, are called node- and edge-potentials, respectively. For a given energy function E_{conf}, the Boltzmann distribution is given by

$$P_{conf}(\mathbf{r}) = \frac{1}{Z} \exp\left[\frac{-E_{conf}(\mathbf{r})}{T}\right] \qquad (3)$$

where T is a temperature parameter and Z is a normalizing constant. Clearly, using the energy function (1), the distribution (3) can be expressed in the form of (2), with

$$\begin{aligned} \rho_i(r_i) &\propto \exp\left[\frac{-E_l(r_i)}{T}\right], \\ \rho_{ij}(r_i, r_j) &\propto \exp\left[\frac{-E_p(r_i, r_j)}{T}\right]. \end{aligned} \qquad (4)$$

3.1 Belief Propagation (BP)

Belief propagation (BP) is an efficient local message-passing algorithm [43] for making inferences on graphical models. It performs exact inference if the graph $\mathcal{G}$ is a tree, and approximate inference for general graphs. If the graph contains cycles, the algorithm is often referred to as "loopy BP" and there is no convergence guarantee, but many groups have reported excellent results nonetheless, e.g., [15], [42], [16]. Indeed, loopy BP has been applied to the side-chain prediction problem, and the results have been comparable to such state-of-the-art software as SCWRL 3.0 [52], [54].

Given potential functions as defined by (4), messages are computed along the edges of the graph. The sum-product BP algorithm is used to compute marginal distributions; its recursive message updating equation is as follows:

$$m^{(t)}_{i \to j}(r_j) = \sum_{r_i \in \mathcal{R}_i} \Bigg[\rho_i(r_i) \rho_{ij}(r_i, r_j) \times \prod_{k \in \mathcal{N}(i) \backslash j} m^{(t-1)}_{k \to i}(r_i) \Bigg], \qquad (5)$$

where $\mathcal{R}_i$ is the (discrete) state space of r_i, and $m^{(t)}_{i \to j}$ represents the message from node i to j at iteration t. The notation, $\mathcal{N}(i)$, denotes the set of nodes that are neighbours of i. For proteins, the discrete state space $\mathcal{R}_i$ is simply the set of rotamers for residue r_i.

The max-product BP algorithm is used for finding MAP estimates; its recursive updating equation is given by

$$m^{(t)}_{i\to j}(r_j) = \max_{r_i\in\mathcal{R}_i}\Bigg[\rho_i(r_i)\rho_{ij}(r_i,r_j)\times \prod_{k\in\mathcal{N}(i)\setminus j} m^{(t-1)}_{k\to i}(r_i)\Bigg]. \quad (6)$$

3.2 Particle Belief Propagation (PBP)

For many applications, e.g., in bioinformatics, computer vision, and other fields, the state space is continuous rather than discrete, or it can be discrete but very large so that enumerating all possible states at each iteration becomes very inefficient. Particle belief propagation (PBP) has been developed recently to address precisely such difficulties [22]. Our goal is to model the conformation of side chains more appropriately as continuous rather than discrete random variables. Hence, PBP is a crucial piece of technology for our work.

The key idea for PBP is the following: At iteration t, if we draw r_i from a certain trial distribution $W_i^{(t)}$, then (5) can be written as an "importance-sampling corrected expectation" [22]:

$$m^{(t)}_{i\to j}(r_j) = \mathbb{E}_{r_i\sim W_i^{(t)}}\Bigg[\frac{\rho_i(r_i)}{W_i^{(t)}(r_i)}\rho_{ij}(r_i,r_j)\times \prod_{k\in\mathcal{N}(i)\setminus j} m^{(t-1)}_{k\to i}(r_i)\Bigg]. \quad (7)$$

It is generally not possible to express the expectation $\mathbb{E}_{r_i\sim W_i^{(t)}}(\cdot)$ in analytic form, but it can be obtained using Monte Carlo techniques [10], [32]. Thus, for each node r_i, the idea is to sample a set of L particles $\{r_i^{(1)}, r_i^{(2)}, ..., r_i^{(L)}\}$ from $W_i^{(t)}$, typically using a Markov Chain Monte Carlo (MCMC) technique such as the Metropolis-Hastings algorithm [32], and then approximate (7) by

$$\widehat{m}^{(t)}_{i\to j}(r_j) = \frac{1}{L}\sum_{l=1}^{L}\Bigg[\frac{\rho_i\left(r_i^{(l)}\right)}{W_i^{(t)}\left(r_i^{(l)}\right)}\rho_{ij}\left(r_i^{(l)},r_j\right)\times \prod_{k\in\mathcal{N}(i)\setminus j} \widehat{m}^{(t-1)}_{k\to i}\left(r_i^{(l)}\right)\Bigg]. \quad (8)$$

In the simplest case, the particles' locations may remain unchanged [22] but, generally, each particle's location is updated at the end of each BP iteration. This is what allows PBP to explore a continuous state space and not be restricted to a fixed set of choices specified a priori, such as a rotamer library; and it is accomplished by re-sampling the particles at each iteration from the distribution, $W_i^{(t)}$, e.g., using the Metropolis-Hastings algorithm. At iteration t, a natural choice of $W_i^{(t)}$ is the current belief of node i,

$$W_i^{(t)}(r_i) \quad\propto\quad \rho_i(r_i)\times\prod_{k\in\mathcal{N}(i)}\widehat{m}^{(t-1)}_{k\to i}(r_i). \quad (9)$$

The max-product version for PBP was first given by Kothapa et al. [27]:

$$\widehat{m}^{(t)}_{i\to j}(r_j) = \max_{l=1,\dots,L}\Bigg[\rho_i\left(r_i^{(l)}\right)\rho_{ij}\left(r_i^{(l)},r_j\right)\times \prod_{k\in\mathcal{N}(i)\setminus j}\widehat{m}^{(t-1)}_{k\to i}\left(r_i^{(l)}\right)\Bigg]. \quad (10)$$

Their paper [27] explains in more detail why the factor $W_i^{(t)}$ does not appear in the square brackets of (10).

4 Fast Approximation of ρ_i, ρ_{ij}

We used mixtures of von-Mises distributions as a fast way to approximate the potential functions $\rho_i(r_i)$ and $\rho_{ij}(r_i,r_j)$. It is well-known that mixture densities can be used to approximate any arbitrary distribution. For example, in nonparametric belief propagation [46], mixtures of Gaussians are used to model and/or approximate $\rho_i(r_i)$ and $\rho_{ij}(r_i,r_j)$. Since we are working in the space of dihedral *angles*, the von-Mises distribution is more appropriate than the Gaussian distribution (see Section 4.1 below).

4.1 The von-Mises (VM) Distribution

The univariate von-Mises (VM) distribution is a probability distribution on a circle. The multivariate generalization was introduced by Mardia et al. [37]. In particular, $\theta\in\mathbb{R}^d$ is said to follow the multivariate von-Mises distribution, $\mathrm{MVM}(\mu,\kappa,\mathbf{\Lambda})$, if its density function is given by

$$f(\theta;\mu,\kappa,\mathbf{\Lambda}) = \frac{1}{Z(\kappa,\mathbf{\Lambda})}\times \exp\left[\kappa^T c(\theta) + \frac{s^T(\theta)\mathbf{\Lambda}s(\theta)}{2}\right] \quad (11)$$

where

$$c_u(\theta)\equiv\cos(\theta_u-\mu_u),\quad s_u(\theta)\equiv\sin(\theta_u-\mu_u)$$

for $u=1,2,...,d$, and $Z(\kappa,\mathbf{\Lambda})$ is a normalizing constant.

The parameter $\mu\in\mathbb{R}^d$ describes the location, i.e., the mean (or center), and the parameter $\kappa\in\mathbb{R}^d>\mathbf{0}$ describes the scale, i.e., the spread (or concentration). The parameter, $\mathbf{\Lambda}=[\lambda_{uv}]\in\mathbb{R}^{d\times d}$ is a matrix whose diagonal elements are zero ($\Lambda_{uu}=0$) and whose off-diagonal elements Λ_{uv} capture the correlation between θ_u and θ_v. It is clear from the definition above that the VM distribution is well suited for modeling angular data, and why it is sometimes referred to as the "Gaussian" distribution on the sphere.

4.2 Use of VM Distribution in Bioinformatics

The von-Mises distribution has been used to model dihedral angles in protein molecules. For example, Mardia et al. [36] used the EM algorithm to fit a mixture of bivariate von-Mises distributions to the two dihedral angles (ϕ, ψ) that describe protein backbones. To model higher-dimensional angular data (e.g., the dihedral angles for side chains), Mardia et al. [37] introduced the more general, multivariate von-Mises distribution (11) by extending the bivariate model of Singh et al. [45]. More recently, Mardia et al. [38] have extended single MVM distributions to mixtures of MVMs. For example, they fitted a 4-dimensional mixture of MVMs to model the two backbone dihedral angles (ϕ and ψ) and the first two side-chain dihedral angles (χ_1 and χ_2) of the amino acid, ILE.

In our work, we also used mixtures of MVMs (see Section 4.4 below). However, our work differs fundamentally from those of Mardia et al. [38]. While they fitted a *single* mixture to model the conformation of a given amino acid using data from different proteins, we used mixtures of MVMs to approximate the node- and edge-potential functions, and *different* mixture models were specified for each ρ_i and ρ_{ij} on a protein-by-protein basis.

4.3 Energy Functions

We now give more details about the energy function (1). We used a very simple energy function that essentially acted as a collision detector. This simple energy function was first popularized by SCWRL [11] and later adopted by TreePack [51] as well.

Given two atoms, a_1 and a_2, SCWRL approximates the van der Waals pairwise potential energy between them by

$$E_{apprx.vdw}(a_1, a_2) = \begin{cases} 0, & \text{if} \quad d > R_0; \\ -k_2 \frac{d}{R_0} + k_2, & \text{if} \quad k_1 R_0 \le d \le R_0; \\ E_{max}, & \text{if} \quad d < k_1 R_0, \end{cases} \qquad (12)$$

where d is the distance between a_1 and a_2; R_0 is the sum of their radii; $E_{max} = 10$; $k_1 = 0.8254$; and $k_2 = E_{max}/(1 - k_1)$. For Carbon (C), Nitrogen (N), Oxygen (O), and Sulfur (S), fixed radii of 1.6, 1.3, 1.7, and 1.7 were used, respectively.

Treating each residue simply as a set of atoms, the pairwise energy function, $E_p(r_i, r_j)$ in (1), is merely calculated by summing over all atom-pairs:

$$E_p(r_i, r_j) = \sum_{a \in r_i, b \in r_j} E_{apprx.vdw}(a, b).$$

The intrinsic energy $E_l(r_i)$ in (1) is computed by

$$E_l(r_i) = -K \log \frac{p(r_i|\phi,\psi)}{p_{max}(r_i|\phi,\psi)} + \sum_{\substack{j<i-1 \\ j>i+1}} E_p(r_i, b_j), \qquad (13)$$

where $p(r_i|\phi,\psi)$ is the rotamer probability specified by the rotamer library, which depends on the two backbone dihedral angles ϕ and ψ; $p_{max}(r_i|\phi,\psi)$ is the probability of the most probable rotamer among the rotamers listed in the library for residue i; and b_j represents the backbone part of residue j. The Dunbrack Lab [11] has suggested that the parameter K be set to 3. In order to calculate $E_p(r_i, b_j)$, r_i and b_j are again treated simply as two sets of atoms, and $E_p(r_i, b_j)$ is computed in the same fashion as (12) over all pairs of atoms in r_i and b_j.

4.4 Approximation of Potential Functions

Notice that the energy functions E_l and E_p given in the previous section — and hence the implied potential functions ρ_i and ρ_{ij}, given by (4) — depend on inter-atomic *distances*, whereas our state space is a set of dihedral *angles* that describe the conformation of each residue. Therefore, a conversion must take place every time the potential functions are evaluated. This is not difficult in principle, and there is existing, standard software for performing such a conversion, e.g., BALL [18]. In order to speed up our computation, however, we used a mixture of von-Mises distributions as a crude approximation to these potential functions.

For example, for residues described by four dihedral angles (e.g., the amino acid LYS), the approximation to the node potential function would be:

$$\widehat{\rho}_i(r_i) \quad = \quad \sum_{\tau} w_\tau f_\tau(\chi_{i1}, ..., \chi_{i4}), \qquad (14)$$

where $f_\tau \sim \text{MVM}(\mu_\tau, \kappa_\tau, \mathbf{\Lambda}_\tau)$ is a (multivariate) von-Mises density function, given by (11), and w_τ denotes the weight of the mixture component τ such that $\sum w_\tau = 1$.

We used simple *spherical* or *radial basis* mixtures, that is, we set $\mathbf{\Lambda}_\tau = \mathbf{0}$. This is the same as treating the dihedral angles as being locally independent — in the future, we plan to generalize this by modeling the local correlations among the χ-angles. We chose $\kappa_\tau = (10, 10, ..., 10)$ for all τ. For each residue i, the set of discrete rotamers from the (backbone-dependent) rotamer library were used as mixture centers, μ_τ. We specified the weight of each component τ to be

$$w_\tau = \alpha \; p(\mu_\tau|\phi,\psi) + (1-\alpha)\frac{1}{\#(\text{mixture components})},$$

where α was chosen to be 0.1, and $p(\mu_\tau|\phi,\psi)$ is the rotamer probability for rotamer μ_τ from the rotamer library.

For the edge potential, our approximation was:

$$\widehat{\rho}_{ij}(r_i, r_j) = \sum_{\tau} \sum_{\tau'} w_{\tau,\tau'} \times f_{\tau,\tau'}(\chi_{i1}, ..., \chi_{id_i}, \chi_{j1}, ..., \chi_{jd_j}), \quad (15)$$

where d_i and d_j are the number of dihedral angles for r_i and r_j, respectively; $f_{\tau,\tau'}$ is, again, a spherical MVM density function, with $\kappa_{\tau,\tau'} = (10, 10, ..., 10)$ and $\mathbf{\Lambda}_{\tau,\tau'} = \mathbf{0}$ for all τ, τ' as before. If the pairwise edge potential between two rotamers — one for residue r_i and another for r_j — exceeded 0.05, their dihedral angles were concatenated together and used as a mixture center, $\mu_{\tau,\tau'}$, with $w_{\tau,\tau'} \propto \rho(\mu_\tau, \mu_{\tau'})$.

The expressions (14) and (15) can be viewed as approximations of ρ_i and ρ_{ij} using a crudely specified single-layer radial basis function network (RBFnet) in angular space.

5 Experiments and Results

We used a data set containing 362 diverse proteins that were previously analyzed by the Alber group [29] with their Ringer program (http://ucxray.berkeley.edu/ringer.htm). The protein data files were retrieved from the PDB. We used functions and modules from the Biochemistry Algorithms Library (BALL) [18] to read and process the PDB files. The electron density maps for the proteins, which were required to run the Ringer program, were downloaded from the Electron Density Server [26]. All experiments were performed on the Sharcnet system (http://www.sharcnet.ca/).

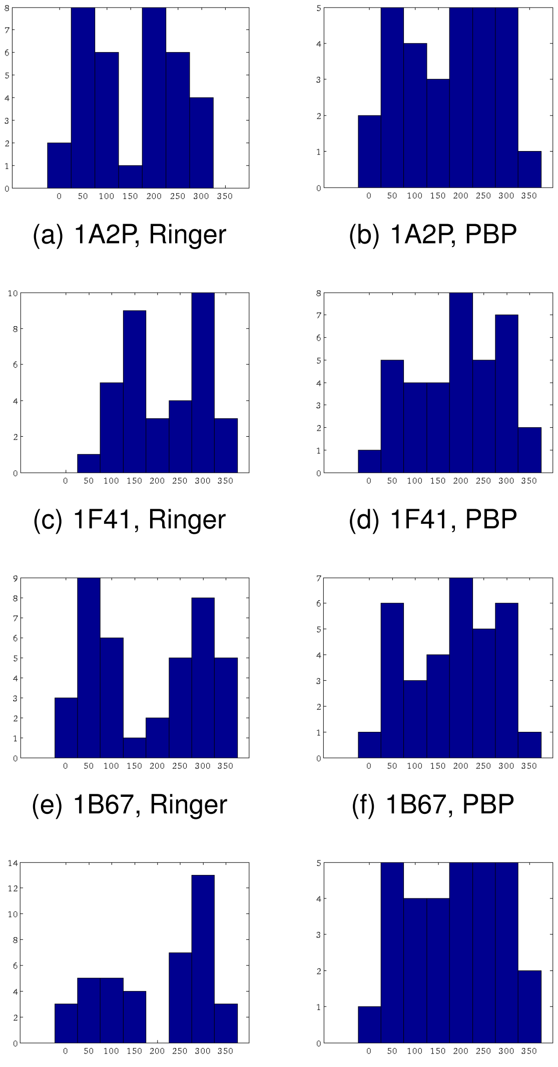
5.1 PBPMixVM

To reflect the fact that we used PBP for inference and mixtures of (multivariate) VM distributions for approximating the potential functions, from this point on we shall refer to our algorithm as PBPMixVM. We implemented it in C++, using the overall architecture provided by GraphLab (http://select.cs.cmu.edu/code/graphlab/) [33].

5.2 The Kolmogorov-Smirnov (KS) Test

For each residue, we used a two-sample Kolmogorov-Smirnov (KS) test [7] to compare the results from Ringer with those from PBPMixVM. The KS-test is a widely used nonparametric test for determining whether two distributions are significantly different from each other. The significance level for the KS-test was set to be 0.05.

Fig. 2 provides some visual illustrations of what the KS-test does. Based on the p-values from individual KS-tests, we selected four residues, whose corresponding p-values from the aforementioned KS-tests were 0.99, 0.77, 0.53, and 0.25, respectively. Such a selection is meant to illustrate varying levels of agreement between the PBPMixVM results and the Ringer results — larger p-values indicate higher levels of agreement.



(a) 1A2P, Ringer

(b) 1A2P, PBP

(c) 1F41, Ringer

(d) 1F41, PBP

(e) 1B67, Ringer

(f) 1B67, PBP

(g) 1F4P, Ringer

(h) 1F4P, PBP

Fig. 2. Illustration of results from Ringer (left) and those from PBPMixVM (right; simply labeled as "PBP" in the plots): four specific residues, whose polymorphisms in χ_1 differed to varying degrees as characterized by Ringer and predicted by PBPMixVM. (a) & (b) Residue ASN23, 1A2P, Chain C; KS-test p-value=0.99. (c) & (d) Residue HIS88, 1F41, Chain B; KS-test p-value=0.77. (e) & (f) Residue GLU134, 1B67; KS-test p-value=0.53. (g) & (h) Residue LYS87, 1F4P; KS-test p-value=0.25. Larger p-values indicate better agreement between Ringer and PBP.

The four selected residues are: residue ASN23, 1A2P, Chain C [39], p-value=0.99; residue HIS88, 1F41, Chain B [20], p-value=0.77; residue GLU134, 1B67 [8], p-value=0.53; and residue LYS87, 1F4P [44], p-value=0.25. For these four residues, the polymorphisms in their respective χ_1-angles as characterized by Ringer and predicted by PBPMixVM are displayed

TABLE 1
Comparison of PBPMixVM and Ringer results: χ_1 and χ_2, average results over all residues, across all proteins in the data set.

Dihedral angle	Percent in agreement (%)	Mean p-value
χ_1	57	0.19
χ_2	56	0.17

TABLE 2
Comparison of PBPMixVM and Ringer results: χ_1 only, average results over all residues of the same amino acid type, across all proteins in the data set.

Amino acid	Total No.	Percent in agreement (%)	Mean p-value
ARG	4067	57	0.18
ASN	3613	67	0.24
ASP	5072	63	0.21
CYS	1342	62	0.21
GLN	3180	67	0.24
GLU	5466	71	0.26
HIS	2080	52	0.16
ILE	4894	52	0.17
LEU	8250	64	0.22
LYS	4890	53	0.16
MET	1348	69	0.23
PHE	3675	48	0.14
PRO	4097	42	0.09
SER	4776	77	0.29
THR	4749	77	0.29
TRP	1259	54	0.16
TYR	3044	50	0.15
VAL	6422	66	0.23

TABLE 3
Comparison of PBPMixVM and Ringer results: χ_1 only, average results over all residues of the same amino acid type *and* having the same secondary structure, across all proteins in the data set.

Amino acid	Secondary structure	Total No.	Percent in agreement (%)	Mean p-value
ARG	Helix	1774	60	0.19
	Strand	815	56	0.17
	Loop	1478	56	0.17
ASN	Helix	1070	64	0.21
	Strand	541	64	0.23
	Loop	2002	69	0.25
ASP	Helix	1764	62	0.21
	Strand	571	59	0.19
	Loop	2737	65	0.22
CYS	Helix	423	63	0.22
	Strand	409	57	0.20
	Loop	510	66	0.20
GLN	Helix	1573	71	0.26
	Strand	546	66	0.22
	Loop	1060	61	0.20
GLU	Helix	2854	73	0.27
	Strand	861	66	0.24
	Loop	1751	70	0.25
HIS	Helix	679	53	0.16
	Strand	456	52	0.17
	Loop	945	51	0.16
ILE	Helix	1804	51	0.16
	Strand	1885	52	0.17
	Loop	1205	52	0.18
LEU	Helix	3947	71	0.26
	Strand	2107	57	0.18
	Loop	2195	60	0.20
LYS	Helix	2203	54	0.17
	Strand	870	54	0.17
	Loop	1817	50	0.15
MET	Helix	581	68	0.24
	Strand	318	67	0.22
	Loop	449	70	0.24
PHE	Helix	1383	51	0.14
	Strand	1179	42	0.13
	Loop	1112	50	0.15
PRO	Helix	841	44	0.09
	Strand	363	44	0.10
	Loop	2893	42	0.09
SER	Helix	1462	78	0.30
	Strand	884	74	0.28
	Loop	2430	77	0.29
THR	Helix	1450	78	0.29
	Strand	1255	75	0.27
	Loop	2043	76	0.29
TRP	Helix	487	58	0.18
	Strand	392	48	0.14
	Loop	380	55	0.17
TYR	Helix	1122	51	0.15
	Strand	1009	46	0.13
	Loop	913	52	0.16
VAL	Helix	2087	72	0.26
	Strand	2736	63	0.21
	Loop	1599	65	0.24

next to each other in Fig. 2, ordered by p-values from top to bottom.

5.3 Comparative Results

From the individual KS-tests, we computed two summary statistics to evaluate the overall agreement between our results from PBPMixVM and those given by Ringer: (i) *percent in agreement* — the fraction of residues for which the KS-test failed to show a statistically significant difference; and (ii) *mean p-value* — the average p-value from individual KS-tests.

Table 1 shows the overall results for the first two dihedral angles, χ_1 and χ_2, averaged over all residues across all proteins in our data set. Table 2 shows results for χ_1 only, averaged over residues of the same amino acid type across all proteins in our data set. Based on the KS-tests, these proteins' side-chain polymorphisms as predicted by PBPMixVM and as described by Ringer agreed for well over 50% of all the residues, and the average p-value from these KS-tests was about 0.20, much higher than the typical cutoff value of 0.05.

We also can observe from Table 2 that, for some residue types, including not only those having just one χ-angle, e.g., Serine (SER), Threonine (THR), Valine (VAL), but also those having relatively large structures and hence, multiple χ-angles, e.g., Asparagine (ASN), Glutamine (GLN), Glutamic Acid (GLU), Methionine (MET), the agreement between PBPMixVM and Ringer can be noticeably higher than the overall average (Table 1) — the *percent in agreement* for some residue types was close to 70% and 80%, and the corresponding *mean p-value* was close to 0.25 and 0.30. To the extent that the Ringer program can discover alternate side-chain conformations, PBPMixVM can be seen to have the ability to predict alternate side-chain conformations for these residues as well. The agreement with Ringer for large residues having multiple χ-angles, in particular, are indications that using mixtures of *locally independent* VM distributions in our approximation of the potential functions (see Section 4.4) has not had a significant impact on our algorithm.

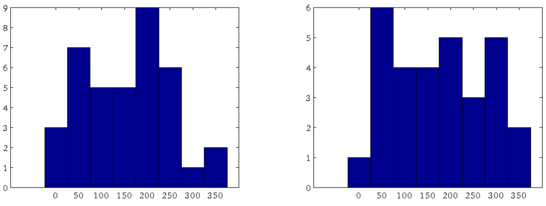
Table 3 further groups the results by the secondary structures of the residues, which we obtained using the DSSP software [25], [24]. Here we easily can see that the agreement between PBPMixVM and Ringer was generally better for residues whose secondary structures are helices. This is not surprising, since helices typically are more stable.

Earlier, we mentioned in Section 1 that, using Ringer, Lang et al. [29] had uncovered interesting polymorphisms in the protein, Calmodulin (CaM) [50]. A curious residue in that protein is SER38. The currently modeled χ_1-angle for SER38 changes conformation from $80°$ in the unbound form of CaM (PDB ID: 1EXR) to $295°$ in the complex form (PDB ID: 2O5G). By analyzing crystallography data for 1EXR (the unbound form of CaM), Ringer successfully recognized the $295°$ conformation — often detectable only from 2O5G (the complex form of CaM) — as a secondary peak. This result was scientifically significant because, previously, such conformational changes could not have been easily identifiable without a complete structural refinement of both the bound and the unbound proteins, but Ringer was able to detect this conformational polymorphism from the unbound molecule alone. We also analyzed 1EXR with PBPMixVM. Our predicted polymorphism for the χ_1-angle of SER38 agreed well with the result from Ringer (Fig. 3; KS-test p-value = 0.64). In particular, PBPMixVM also predicted the secondary conformation near $295°$ from the unbound form of CaM (1EXR).

5.4 Some Computational Details

The PBPMixVM algorithm was deemed to have converged when the Kullback-Leibler (KL) divergence between $W_i^{(t)}(r_i)$ and $W_i^{(t-1)}(r_i)$ fell below 10^{-8} for each residue i, where $W_i^{(t)}$ denotes the belief function of node i at iteration t, given previously in (9). For all proteins in our data set, PBPMixVM converged in < 50 iterations.

At the moment, PBPMixVM is relatively slow, compared with some other side-chain prediction algorithms such as SCWRL. Whereas the running time of SCWRL is on the order of seconds or minutes, that of PBPMixVM is on the order of hours. This is mainly because, within *each* PBP iteration, we must run a *separate* MCMC to update the particles for *each* residue! To speed up the computation, we used a relatively small number of particles and relatively short MCMC chains to analyze all the proteins.



(a) Ringer

(b) PBP

Fig. 3. Calmodulin (1EXR), residue SER38: polymorphism in χ_1 as extracted by Ringer from crystallography data (left) and predicted by PBPMixVM (right).

However, we did examine, using a small subset of 20 proteins, how much the performance of PBPMixVM could be affected by these computational parameters. On this small subset at least, increasing the length of the MCMC chains and the number of particles per residue had little effect on the overall level of agreement between PBPMixVM and Ringer.

6 Conclusion and Future Work

Proteins in crystals undergo a lot of large- and small-scale motions. Hence, studying a protein molecule with a single conformational model is not adequate. We have developed a computational approach capable of predicting residue-specific conformational polymorphisms. We modeled side-chain dihedral angles as continuous random variables in an MRF, and used PBP as our main inference technique. To speed up the computation, we approximated the (continuous) node- and edge-potential functions by mixtures of VM distributions. For each node in the MRF, a set of particles were sampled at each iteration to represent its distribution. After convergence, these node-specific marginal distributions could be seen to encode information about alternate side-chain conformations. To the best of our knowledge, this work is the first one to address the prediction of side-chain polymorphisms from a purely computational point of view, without relying on additional experimental inputs such as electron density data.

A distinct feature of our method is the treatment of side-chain dihedral angles as continuous variables. We believe it constitutes an important (and necessary) step toward being able to provide an accurate description of side-chain ensembles, and to discover low-occupancy conformers.

As mentioned earlier (Section 5.4), PBPMixVM is relatively slow at the moment, due to the need to update the particles for *each* residue by a *separate* MCMC within *each* PBP iteration. Although we haven't yet done so, the running time of our algorithm could be improved significantly by parallelizing some of these updates. For local message passing, we are currently using a synchronous schedule, but there have been suggestions that using an asynchronous schedule could further accelerate BP-types of algorithms [12].

We are also considering some other refinements to our algorithm, for example, improving our approximation of the potential functions by modeling the local correlations among the dihedral angles. We also believe that combining the results from PBPMixVM with those from state-of-the-art side-chain prediction algorithms, such as SCWRL [28] and TreePack [51], can further enhance the accuracy and reliability of the predicted side-chain polymorphisms.

References

[1] T. Akutsu, "NP-hardness Results for Protein Side-chain Packing," in Miyano, S., Takagi, T., eds., *Genome Informatics*, vol. 8, pp. 180-186, 1997.

[2] 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, vol. 28, pp. 235-242, 2000.

[3] D. D. Boehr, R. Nussinov, and P. E. Wright, "The Role of Dynamic Conformational Ensembles in Biomolecular Recognition," Nat. Chem. Biol., vol. 5, no. 11, pp. 789-796, 2009.

[4] M. Bower, F. Cohen, and R. Dunbrack, "Prediction of Protein Side-Chain Rotamers from a Backbone-Dependent Rotamer Library: A New Homology Modeling Tool," J. Mol. Biol., vol. 267, pp. 1268-1282, 1997.

[5] R. Bruschweiler, "New Approaches to the Dynamic Interpretation and Prediction of NMR Relaxation Data from Proteins," Curr. Opin. Struct. Biol., vol. 13, pp. 175-183, 2003.

[6] A. Canutescu, A. Shelenkov, and R. Dunbrack, "A Graph-Theory Algorithm for Rapid Protein Side-Chain Prediction," Protein Sci., vol. 12, no. 9, pp. 2001-2014, 2003.

[7] W. J. Conover, *Practical Nonparametric Statistics*, New York: John Wiley & Sons, 1971.

[8] K. Decanniere, A. M. Babu, K. Sandman, J. N. Reeve, and U. Heinemann, "Crystal Structures of Recombinant Histones HMfA and HMfB from the Hyperthermophilic Archaeon Methanothermus Fervidus," J. Mol. Biol., vol. 303, no. 1, pp. 35-47, 2000.

[9] J. Desmet, M. De Maeyer, and I. Lasters, "The Dead-End Elimination Theorem and Its Use In Protein Side-Chain Positioning," Nature, vol. 356, pp. 539-542, 1992.

[10] A. Doucet, N. de Freitas, and N. J. Gordon, *Sequential Monte Carlo Methods in Practice*, New York: Springer-Verlag, 2001.

[11] R. Dunbrack and M. Kurplus, "Backbone-Dependent Rotamer Library for Proteins: Application to Side-Chain Prediction," J. Mol. Biol., vol. 230, pp. 543-574, 1993.

[12] G. Elidan, I. McGraw, and D. Koller, "Residual Belief Propagation: Informed Scheduling for Asynchronous Message Passing," Proc. 22nd Conf. Uncertainty in Artificial Intelligence (UAI '06), 2006.

[13] H. Frauenfelder, G. A. Petsko, and D. Tsernoglou, "Temperature-Dependent X-Ray-Diffraction as a Probe of Protein Structural Dynamics," Nature, vol. 280, pp. 558-563, 1979.

[14] K. K. Frederick, M. S. Marlow, K. G. Valentine, and A. J. Wand, "Conformational Entropy in Molecular Recognition by Proteins," Nature, vol. 448, pp. 325-329, 2007.

[15] W. Freeman and E. Pasztor, "Learning to Estimate Scenes from Images," Advances in Neural Information Processing Systems 11 (NIPS '99), MIT Press, 1999.

[16] B. Frey, R. Koetter, and N. Petrovic, "Very Loopy Belief Propagation for Unwrapping Phase Images," Advances in Neural Information Processing Systems 14 (NIPS '04), MIT Press, 2004.

[17] R. F. Goldstein, "Efficient Rotamer Elimination Applied to Protein Side Chains and Related Spin Glasses," Biophys. J., vol. 66, pp. 1335-1340, 1994.

[18] A. Hildebrandt, A. K. Dehof, A. Rurainski, A. Bertsch, M. Schumann, N. C. Toussaint, A. Moll, D. Stöckel, S. Nickels, S. C. Mueller, H.-P. Lenhof, and O. Kohlbacher, "BALL - Biochemical Algorithms Library 1.3," BMC Bioinformatics, vol. 11, article 531, 2010.

[19] L. Holm and C. Sander, "Fast and Simple Monte Carlo Algorithm for Side-Chain Optimization in Proteins: Application to Model," Proteins: Structure, Function, and Genetics, vol. 14, pp. 213-223, 1992.

[20] A. Hörnberg, T. Eneqvist, A. Olofsson, E. Lundgren, and A. E. Sauer-Eriksson, "A Comparative Analysis of 23 Structures of the Amyloidogenic Protein Transthyretin," J. Mol. Biol., vol. 302, no. 3, pp. 649-669, 2000.

[21] J. Hwang and W. Liao, "Side-Chain Prediction by Neural Networks and Simulated Annealing Optimization," Protein Eng., vol. 8, no. 4, pp. 363-370, 1995.

[22] A. Ihler and D. McAllester, "Particle Belief Propagation," Proc. 12th Int. Conf. on Artificial Intelligence and Statistics (AISTATS '09), pp. 256-263, 2007.

[23] R. Ishima and D. A. Torchia, "Protein Dynamics from NMR," Nat. Struct. Biol., vol. 7, pp. 740-743, 2000.

[24] R. P. Joosten, T. A. H. te Beek, E. Krieger, M. L. Hekkelman, R. W. W. Hooft, R. Schneider, C. Sander, and G. Vriend, "A Series of PDB Related Databases for Everyday Needs," Nucleic Acids Research, vol. 39 (database issue), pp. D411-D419, 2010, doi: 10.1093/nar/gkq1105.

[25] W. Kabsch and C. Sander, "Dictionary of Protein Secondary Structure: Pattern Recognition of Hydrogen-Bonded and Geometrical Features," Biopolymers, vol. 22, pp. 2577-2637, 1983.

[26] G. J. Kleywegt, M. R. Harris, J. Y. Zou, T. C. Taylor, A. Wählby, and T. A. Jones, "The Uppsala Electron-Density Server", Acta Cryst., vol. D60, pp. 2240-2249, 2004.

[27] R. Kothapa, J. Pacheco, and E. Sudderth, "Max-Product Particle Belief Propagation," Technical Report, Department of Computer Science, Brown University, 2011.

[28] G. G. Krivov, M. V. Shapovalov, and R. Dunbrack, "Improved Prediction of Protein Side-Chain Conformations with SCWRL4," Proteins, vol. 77, no. 4, pp. 778-795, 2009.

[29] P. T. Lang, H.-L. Ng, J. S. Fraser, J. E. Corn, N. Echols, M. Sales, J. M. Holton, and T. Alber, "Automated Electron-Density Sampling Reveals Widespread Conformational Polymorphism in Proteins," Protein Sci., vol. 19, pp. 1420-1431, 2010.

[30] C. Lee and S. Subbiah, "Prediction of Protein Side-Chain Conformation by Packing Optimization" J. Mol. Biol., vol. 213, pp. 373-388, 1991.

[31] S. C. Li, D. Bu, and M. Li, "Residues with Similar Hexagon Neighborhoods Share Similar Side-chain Conformations," IEEE/ACM Trans. Comput. Biology Bioinform., vol. 9, no. 1, pp. 240-248, 2012.

[32] J. S. Liu, *Monte Carlo Strategies in Scientific Computing*, New York: Springer-Verlag, 2001.

[33] Y. Low, J. Gonzalez, A. Kyrola, D. Bickson, C. Guestrin, and J. Hellerstein, "Graphlab: A new framework for parallel machine learning," Proc. 26th Conf. Uncertainty in Artificial Intelligence (UAI '10), 2010.

[34] B. Ma, S. Kumar, C.-J. Tsai, and R. Nussinov, "Folding Funnels and Binding Mechanisms," Protein Eng., vol. 12, pp. 713-720, 1999.

[35] B. Ma, S. Kumar, C.-J. Tsai, H. Wolfson, N. Sinha, and R. Nussinov, "Protein-Ligand Interactions: Induced Fit," in *Encyclopedia of Life Sciences*, Chichester: John Wiley, 2002, doi:10.1038/npg.els.0003140.

[36] K. V. Mardia, C. C. Taylor, and G.K. Subramaniam, "Protein Bioinformatics and Mixtures of Bivariate von-Mises Distributions for Angular Data," Biometrics, vol. 63, pp. 505-512, 2007.

[37] K. V. Mardia, G. Hughes, C. C. Taylor, and H. Singh, "A Multivariate von-Mises Distribution with Applications to Bioinformatics," Can. J. Stat., vol. 36, no. 1, pp. 99-109, 2008.

[38] K. V. Mardia, J. T. Kent, Z. Zhang, C. C. Taylor, and T. Hamelryck, "Mixtures of Concentrated Multivariate Sine Distributions

with Application to Bioinformatics," J. Appl. Stat., vol. 39, no. 11, pp. 2475-2492, 2012.

[39] C. Martin, V. Richard, M. Salem, R. Hartley, and Y. Mauguen, "Refinement and Structural Analysis of Barnase at 1.5Å Resolution," Acta Crystallogr. D. Biol. Crystallogr., vol. 55, pt. 2, pp. 386-398, 1999.

[40] B. W. Matthews, "Peripatetic Proteins," Protein Sci., vol. 19, pp. 1279-1280, 2010.

[41] A. Mittermaier and L. E. Kay, "New Tools Provide New Insights in NMR Studies of Protein Dynamics," Science, vol. 312, pp. 224-228, 2006.

[42] K. Murphy, Y. Weiss, and M. Jordan, "Loopy Belief Propagation for Approximate Inference: An Empirical Study," Proc. 15th Conf. Uncertainty in Artificial Intelligence (UAI '99), pp. 467-475, 1999.

[43] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, San Francisco: Morgan Kaufmann, 1988.

[44] R. A. Reynolds, W. Watt, and K. D. Watenpaugh, "Structures and Comparison of the Y98H (2.0 Å) and Y98W (1.5 Å) Mutants of Flavodoxin (Desulfovibrio Vulgaris)," Acta Crystallogr. D. Biol. Crystallogr., vol. 57, pt. 4, pp. 527-535, 2001.

[45] H. Singh, V. Hnizdo, and E. Demchuk, "Probabilistic Model For Two Dependent Circular Variables," Biometrika, vol. 89, pp. 719-723, 2002.

[46] E. Sudderth, A. Ihler, W. Freeman, and A. Wilsky, "Nonparametric Belief Propagation," Proc. 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '03), pp. 605-612, 2003.

[47] H. van den Bedem, A. Dhanik, J. C. Latombe, and A. M. Deacon, "Modeling Discrete Heterogeneity in X-Ray Diffraction Data by Fitting Multi-Conformers," Acta Cryst. vol. D65, pp. 1107-1117, 2009.

[48] M. Vendruscolo, "Determination of Conformationally Heterogeneous States of Proteins," Curr. Opin. Struct. Biol., vol. 17, pp. 15-20, 2007.

[49] J. Wang, M. Dauter, R. Alkire, A. Joachimiak, and Z. Dauter, "Triclinic Lysozyme at 0.65 Angstrom Resolution," Acta Crystallogr. D. Biol. Crystallogr., vol. 63, pt. 12, pp. 1254-1268, 2007.

[50] M. A. Wilson, A. T. Brunger, "The 1.0Å Crystal Structure of Ca^{2+}-bound Calmodulin: An Analysis of Disorder and Implications for Functionally Relevant Plasticity," J. Mol. Biol., vol. 301, no. 5, pp. 1237-1256, 2000.

[51] J. Xu and B. Berger, "Fast and Accurate Algorithms for Protein Side-Chain Packing," J. ACM., vol. 53, no. 4, pp. 533-557, 2006.

[52] C. Yanover and Y. Weiss, "Approximate Inference and Protein Folding," Advances in Neural Information Processing Systems (NIPS '02), MIT Press, 2002.

[53] C. Yanover, T. Meltzer, and Y. Weiss, "Linear Programming Relaxations and Belief Propagation: An Empirical Study," J. Mach. Learn. Res., vol. 7, pp. 1887-1907, 2006.

[54] C. Yanover and Y. Weiss, "Approximate Inference and Side-Chain Prediction," Technical Report, School of Computer Science and Engineering, Hebrew University of Jerusalen, 2007.

Laleh Soltan Ghoraie received her MSc degree in Computer Science from the University of Windsor, Ontario, Canada, in 2009. She is currently a PhD student at the University of Waterloo, Ontario, Canada. Her research interests include probabilistic graphical models, protein structure prediction and conformational ensembles.

Forbes Burkowski is an Associate Professor in the David R. Cheriton School of Computer Science, University of Waterloo, Ontario, Canada. Over the past eight years, he has served as the Director of Bioinformatics and participated in the design of the undergraduate program in Bioinformatics at the University of Waterloo. He has taught the fourth year Structural Bioinformatics course since 2003.

Shuai Cheng Li received his BSc (Hons) and the MSc degrees from the National University of Singapore (NUS) in 2001 and 2002, respectively. Between 2002 and 2004, he was a full-time research associate in the database research group at the NUS. He was a doctoral student with Ming Li at the University of Waterloo, Ontario, Canada, between 2004 and 2009. His doctoral dissertation addressed the protein structure prediction problem from various aspects. He was a postdoctoral fellow with Prof. Richard M. Karp at the University of California, Berkeley between 2009 and 2011. His current research interests include bioinformatics and algorithms, with a focus on problems related to protein structures.

Mu Zhu is Associate Professor of Statistics at the University of Waterloo, Ontario, Canada. He received his AB (magna cum laude) degree in applied mathematics from Harvard University, Cambridge, MA, USA, in 1995, and his PhD degree in statistics from Stanford University, Stanford, CA, USA, in 2001. His primary research interests are machine learning and multivariate analysis.