Partition function estimation in computational protein design with continuous-label Markov random fields

Aditya Mukund

A thesis submitted to the Department of Computer Science for honors
Duke University
Durham, North Carolina
2017

Contents

1 Abstract

Proteins perform a variety of biological tasks, and drive many of the dynamic processes that make life possible. Computational structure-based protein design (CSPD) involves computing optimal sequences of amino acids with respect to particular backbones, or folds, in order to produce proteins with novel functions. In particular, it is crucial to be able to accurately model protein-protein interfaces (PPIs) in order to realize desired functionalities. Accurate modeling of PPIs raises two significant considerations. First, incorporating continuous side-chain flexibility in the design process has been shown to significantly improve the quality of designs. Second, because proteins exist as ensembles of structures, many of the properties we wish to design, including binding affinity, require the computation of ensemble properties as opposed to features of particular conformations. The bottleneck in many design algorithms that attempt to handle the ensemble nature of protein structure, including the Donald Lab's K^* algorithm, is the computation of the partition function, which is the sum of the Boltzmann-weighted energies of all the conformational states of a protein or protein-ligand complex. Protein design can be formulated as an inference problem on Markov random fields (MRFs), where each residue to be designed is represented by a node in the MRF and an edge is placed between nodes corresponding to interacting residues. Label sets on each vertex correspond to allowed flexibility in the underlying design problem. The aim of this work is to extend message-passing algorithms that estimate the partition function for Markov random fields with discrete label sets to MRFs with continuous label sets in order to compute the partition function for PPIs with continuous flexibility and continuous entropy.

2 Computational Structure-Based Protein Design

The immense biomedical importance of proteins makes them attractive for use as therapeutic agents and as targets for novel therapies. However, the sheer number of possibly relevant protein structures in any biological situation renders exhaustive experimental analysis and validation of every candidate protein impractical in most scenarios. In order to comprehensively explore the space of possible drugs and drug targets, it is thus necessary to use computational methods to design and test biomedically relevant molecules *in silico*. This section provides some background on existing protein design algorithms in order to help explain the differences between algorithms using continuous-label MRFs and other design algorithms.

Protein design algorithms attempt to identify optimal amino acid sequences to produce target structures with specific and/or novel functionalities. Computational structure-based protein design (CSPD) seeks to search over the space of protein structures and/or sequences, and select optimal proteins according to parameters defined by a provided *input model* that not only defines permissible amino acid sequences and conformations, but also provides energy functions for analyzing and evaluating the optimality of particular protein structures. Because the size of the resulting conformational space is exponential in the number of residue positions on the protein being designed, exhaustive search of the space is impossible. This demands the development of novel algorithms to efficiently identify promising structures; these algorithms then find applications in various experimental designs, including predicting resistance mutations [18] and designing inhibitors to rescue lost protein functionality in diseases such as cystic fibrosis [20].

2.1 Conformation Optimization and the GMEC Model

The traditional formulation of the protein design problem takes as input a rigid protein backbone and places amino acids along that backbone in energetically optimal conformations

[6]. Thus, choosing amino acids for each position involves not only a choice of sequence but, critically, a choice of structure. Though amino acid side chain motion is continuous in solution, initial formulations of the protein design problem restrict possible conformations of each amino acid side chain to a discrete set of rigid conformations. Each amino acid has a set number of side chain dihedral angles, also called χ angles; rotameric assignments are thus often represented as assignments to each χ angle of the residue.

These rotational isomers, or rotamers, in turn define the *conformational space* over which protein design algorithms search [16]. Specifically, if there are n residues which are to be designed or redesigned, and each residue has on average q conformations, then the conformational space consists of n^q conformations; that is, the overall conformational space is just the Cartesian product of the conformational spaces (sets of rotamers) of each mutable amino acid.

Thus the protein design problem can be reduced to a side-chain placement problem that seeks to optimize an energy function that evaluates the energetic favorability of a conformation over a set of discrete conformations. This permits the use of efficient provable search techniques, including the combination of dead-end elimination (DEE) and A^* search [6]. This side-chain placement formulation saw early success in the full-sequence redesign of a Zinc finger protein [5].

Determining optimal amino acid conformations is a key problem in CSPD, and inadequate modeling of protein side-chain flexibility can cause algorithms to fail to identify biomedically relevant protein structures. While modeling amino acid conformations as rigid structures is computationally attractive, including continuous side-chain flexibility has proved to be crucial for accurate designs. For example, the iMinDEE algorithm allows amino acid side chains to minimize their energy within a square voxel with a length in each dimension of the amino acid's configuration space of 18 degrees; in turn, it computes lower bounds on protein energies to enumerate optimal structures and design peptides [9]. This is the first axis along

which we can characterize protein design algorithms: the inclusion of *continuous flexibility* when modeling amino acid side-chain movements.

2.2 Computing Ensemble Properties of Proteins

Both of these formulations of the protein design problems, side-chain placement and iMinDEE, seek to identify optimal structures by computing the global minimum energy conformation (GMEC). In this manner protein function optimization is treated as protein structure optimization, and the goal within the GMEC model is to identify the lowest-energy conformation or structure. However, this formulation is inaccurate in a biological sense when attempting to engineer proteins with novel functionality. Proteins exist as thermodynamic ensembles, and binding affinities depend on the free energy of these ensembles as opposed to internal energies of particular conformations. This is to say that protein design algorithms that attempt to optimize for binding need to take into condition entropic changes as well as enthalpic changes, and understand the ensemble properties of sequences being considered [7]. This is the second axis along which we may characterize protein design algorithms: the inclusion of *ensemble properties* when trying to design novel functionality.

Computing protein ensembles (and the associated partition function) is significantly harder than enumerating the lowest energy conformation; however, it is still possible to compute provably accurate approximations to the partition function, as the K^* algorithm does. The algorithm computes an approximation to the binding constant of two proteins by computing the ratio of partition functions between bound and unbound states. To approximate each partition function, K^* enumerates conformations in order of increasing energy using the A^* algorithm and stops once a provable ε-approximation to the partition function has been computed [21].

The probability of a protein occupying a particular conformational microstate is proportional to the negative exponential of the energy of that state; the probability p of a protein

existing in the conformation c is expressible as

$$p(c) = \frac{\exp\left(-E_c/RT\right)}{Z} \tag{1}$$

where E_c is the energy of the conformation c, R is the gas constant, T is the temperature (the quantity $\beta = 1/RT$ is also referred to as thermodynamic beta), and Z is a normalizing constant called the *partition function*. That is, if $\mathcal{C}$ is the conformational space of the protein, then Z is defined as:

$$Z = \int_{c \in \mathcal{C}} \exp\left(-E_c/RT\right). \tag{2}$$

Computation of the partition function is particularly significant in that being able to estimate the partition function allows for the estimation of several other important thermodynamic quantities, including binding constants, free energy, and entropy. Most immediately, the binding constant between a protein and ligand can be approximated as the ratio of partition functions in the bound and unbound state:

$$k_a = \frac{1}{k_d} = \frac{Z_{PL}}{Z_P Z_L} \tag{3}$$

where Z_{PL} is the partition function of the protein and ligand bound together, Z_P is the partition function of the unbound protein, and Z_L is the partition function of the unbound ligand.

The K^* algorithm computes an approximation q^* to Z based on discretization of the conformational space into a set of rigid conformations and enumeration of those conformations in order of increasing energy. This is possible through the construction of a *conformational tree*, with each level in the tree representing a particular residues and each node in the tree representing a particular assignment to the corresponding residue. Thus, traversing from the root of the tree to the leaves involves repeatedly assigning rotamers to each mutable residue, meaning that each leaf represents a fully defined conformation. Thus, through repeated use

of the A^* search algorithm, protein conformations (leaves in the tree) can be enumerated in order of increasing energy. Then, because the conformations to be enumerated will always have lower Boltzmann weights than the conformations that have already been enumerated, it is possible to bound the error between the approximation produced by the partial enumeration of the sequence and the "true" value of the sum of Boltzmann weights.

In this manner an ε-approximation to Z can be computed without enumerating every conformation. The iMinDEE/K^* algorithm performs this same process; however, instead of conformations being derived from rotameric assignments to each mutable residue, conformations are minimized prior to being included in the sum of Boltzmann weights. This allows for the inclusion of continuous flexibility in the computation of ensemble properties. A correspondingly modified ε-approximation to Z can be computed, and the ensuing upper/lower bounds, are presented in [10].

The iMinDEE/K^* algorithm incorporates continuous flexibility in that amino acid side chains in any enumerated conformation are allowed to minimize before the Boltzmann-weighted energy term is incorporated into the partition function approximation. However, the approximation still necessitates taking a sum of Boltzmann weights over a discrete set of conformations, resulting in the computation of a discrete representation of entropy.

In contrast, the algorithms involving MRFs with continuous label sets involve an energy landscape that is truly continuous; the partition function approximation requires integration of continuous functions over unary and pairwise conformational spaces (in this work that computation is facilitated through the use of reproducing kernel Hilbert space representations of functions). Thus the goal of using probabilistic graphical models is to allow the partition function approximation to more accurately reflect the energy landscape by integrating the exponential term over the conformational space instead of assuming a constant value for the term. In turn the value for the approximation, instead of being a sum of constant values, is computed as a sum of integrals; we say this approach includes *continuous entropy*, which is distinct from the *continuous flexibility* used in algorithms such as iMinDEE/K^*.

3 From Protein Design to Graphical Models

Translating protein design into the graphical model framework involves encoding each protein design problem as an undirected graph G. As explained in [13], this involves creating a vertex (or node) in the graph for each designable residue in the design problem. Additionally, edges are added between nodes corresponding to residues that interact. Oftentimes this means that the resulting graph is the complete graph, although sparse graphs in protein design have been the topic of significant consideration [12]. Last, a label set representing the possible amino acids and conformations at the corresponding residue in the protein is attached to each node in the graph.

3.1 Constructing the MRF

Formally, a Markov random field is a set of random variables $X = \{x_1, \ldots, x_n\}$ that satisfy a Markov property described by an undirected graph. That is, given a graph $G = (V, E)$ where each $v \in V$ corresponds to a particular $x \in X$, X forms an MRF with respect to G if it satisfies the global Markov property: any two subsets of variables are conditionally independent given a separating subset. By the Hammersley-Clifford theorem, a probability distribution with positive densities (such as a probability distribution over conformational states corresponding to a protein structural ensemble) satisfies the Markov property with respect to a graph G if and only if its density can be factorized over the cliques of the graph. This is equivalent to saying that the distribution is a Gibbs random field. In the field of protein design, when energy functions are often (including in this work) pairwise-decomposable, these conditions hold.

Suppose we want to compute the partition function, given the following:

- A set of n mutable residues $r_1, \ldots, r_n$
- A set of rotamers for each residue: q_i is the set of rotamers for the i-th residue.
- Intra-rotamer energy functions θ_i, $1 \leq i \leq n$

- Inter-rotamer energy functions θ_{ij}, $1 \leq i \leq n$, $1 \leq j \leq n$, $i \neq j$

Then, we can construct a pairwise MRF representing the protein design problem as follows:

- Construct an undirected graph G with n vertices $v_1, \ldots, v_n$
- Add an edge between v_i and v_j if r_i and r_j interact
- Let the label set L_i of v_i be a set of labels such that each label corresponds to a rotameric assignment to the residue r_i. That is, $|L_i| = |q_i|$.
- Define vertex potentials $\phi_i = \exp(-\theta_i/RT)$ and edge potentials $\phi_{ij} = \exp(-\theta_{ij}/RT)$

It is worth noting that these vertex potentials aren't normalized by the partition function – they are an non-normalized Gibbs measure.

These energy functions serve as a *factorization* of the distribution; if x_i is a rotameric assignment to the i-th residue the probability $p(x_1, \ldots, x_n)$ of the fully defined conformation $(x_1, \ldots, x_n)$ is proportional to a product of Boltzmann weights

$$p(x_1, \ldots, x_n) = \frac{1}{Z} \prod_{i,j} \exp\left(\frac{-\theta(x_i, x_j)}{RT}\right) \prod_i \exp\left(\frac{-\theta(x_i)}{RT}\right) \tag{4}$$

where Z is the partition function. Importantly, the value of the partition function of the Markov random field is also the value of the partition function of the protein. Thus, we may use algorithms to approximate or compute the partition function in Markov random fields to do the same for proteins.

3.2 A Toy Example

For example, consider an idealized protein design problem with two residues, the first of which has one continuous domain of continuous flexibility and the second of which has two such continuous domains, shown in Figure 1. Each of these residues only has one degree

of freedom, corresponding to only having one χ angle, but in the design problem we are considering subsets of the larger space of all possible χ angles.

Let $\phi_1 : [51, 69] \to \mathbb{R}$ be the potential function computed from the input model for amino acid 1, and $\phi_2 : [21, 39] \cup [81, 99] \to \mathbb{R}$ be the similarly computed potential function for amino acid 2. These domains correspond to voxels of continuous flexibility around rotamers at 60 degrees, 30 degrees, and 90 degrees, respectively. Last, let $\phi_{1,2}$ be the pairwise potential for interactions between amino acid 1 and amino acid 2.

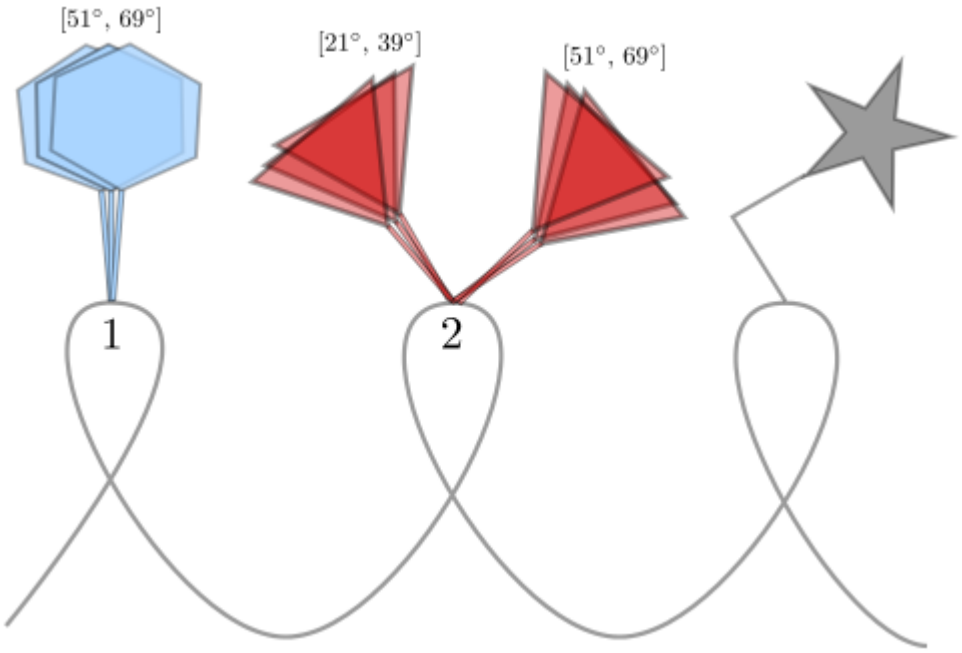

Figure 1: Idealized two-residue protein design problem.

So, in this case, we have two mutable residues, which we can call r_1 and r_2. We then have rotamer sets $q_1 = [51, 69]$ and $q_2 = [21, 39] \cup [81, 99]$, along with energy functions θ_1 and θ_2 and potential functions ϕ_1 and ϕ_2.

This will give us a two-node MRF, the graph of which is shown in Figure 2. This graph has two nodes, v_1 corresponding to r_1 and v_2 corresponding to r_2. Because r_1 and r_2 interact, we add an edge between v_1 and v_2. Next, we have $L_1 = [51, 69]$ and $L_2 = [21, 39] \cup [81, 99]$. The vertex potential on v_1 is ϕ_1, the vertex potential on v_2 is ϕ_2, and the edge potential is

$\phi_{1,2}$. Now, we have a graphical model representing our protein design problem, and can use MRF inference algorithms to compute properties of our protein.

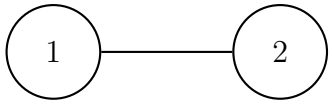

Figure 2: Markov random field corresponding to the idealized two-residue protein design problem in Figure 1.

In graphical models we generally have access only to a non-normalized Gibbs measure over the MRF, and we want to compute the normalized distribution (which is the same as computing Z). There are a number of well known algorithms to bound the log partition function ($\log Z$); the two that are of interest here are the mean field approximation, which provides a lower bound on $\log Z$, and tree-reweighted belief propagation, which provides an upper bound on $\log Z$. Both of these algorithms are message-passing algorithms, which means that they pass messages between vertices until convergence, and then use those messages to compute unary and pairwise marginals or pseudomarginals.

4 Mean-Field Approximations

The mean-field approximation to the log partition function computes a lower bound on $\log Z$ by considering only the space of fully factorizable distributions over the Markov random field. Because this set is a subset of the space of all possible distributions, any value of $\log Z$ derived from considering this smaller set is a lower bound on the maximum value of $\log Z$ over the entire distribution space. Thus, by only considering a smaller set of tractable distributions, we may compute without too much difficulty a lower bound on the log partition function.

Suppose we have a factorized distribution of the form

$$P_{\boldsymbol{\phi}}(\chi) = \frac{1}{Z} \prod_{\phi \in \boldsymbol{\phi}} \phi(U_\phi) \tag{5}$$

where χ represents the set of all MRF nodes, the factors $\phi \in \boldsymbol{\phi}$ comprise the distribution, and the variables $U_\phi \in \text{Scope}[\phi] \subseteq \chi$ are the scope (i.e. the domain) of each factor. Denote by $\widetilde{P}_{\boldsymbol{\phi}}$ the unnormalized distribution, that is, $\widetilde{P}_{\boldsymbol{\phi}}(\chi) = \prod_{\phi \in \boldsymbol{\phi}} \phi(U_\phi)$. Denote by ω_{X_i} the continuous domain or label space corresponding to the node X_i. Each factor corresponds to a clique of some size in the MRF; in our case, the only relevant factors will be factors over single vertices and pairs of vertices.

4.1 Variational Optimization

We are interested in answering questions about $P_{\boldsymbol{\phi}}$, including queries about the partition function Z. This can be done by searching over the space of distributions $\mathcal{D}$ that the MRF can represent to find the distribution Q that matches $P_{\boldsymbol{\phi}}$, specifically by minimizing the relative entropy (or K-L divergence) between Q and $P_{\boldsymbol{\phi}}$. The relative entropy between P_1 and P_2 is defined as

$$D(P_1||P_2) = \int_\chi P_1(x) \ln \frac{P_1(x)}{P_2(x)} \, \mathrm{d}x = \mathbb{E}_{P_1}[\ln P_1] - \mathbb{E}_{P_1}[\ln P_2] \tag{6}$$

where $\mathbb{E}_p[f]$ denotes the expectation of the function f with respect to the distribution p, that is, $\mathbb{E}_p[f] = \int_\chi p(x)f(x)\,\mathrm{d}x$.

Thus, we will attempt to solve the optimization problem

$$\min_{Q \in \mathcal{Q}} D(Q||P_{\boldsymbol{\phi}}), \tag{7}$$

which does not necessitate running any inference in $P_{\boldsymbol{\phi}}$. However, direct optimization of $D(Q||P_{\boldsymbol{\phi}})$ is unwieldy.

Theorem 1. *$D(Q||P_{\boldsymbol{\phi}}) = \ln Z - F(\widetilde{P}_{\boldsymbol{\phi}}, Q)$ where $F(\widetilde{P}_{\boldsymbol{\phi}}, Q)$ is the energy functional*

$$F(\widetilde{P}_{\boldsymbol{\phi}}, Q) = \mathbb{E}_Q\left[\ln \widetilde{P}(\chi)\right] + \mathbf{H}_Q(\chi) = \sum_{\phi \in \boldsymbol{\phi}} \mathbb{E}_Q[\ln \phi] + \mathbf{H}_Q(\chi) \tag{8}$$

and $\mathbf{H}_Q(\chi)$ is the entropy of the distribution Q over the domain χ.

Proof. First, note that

$$D(Q||P_{\boldsymbol{\phi}}) = \mathbb{E}_Q[\ln(Q(\chi))] - \mathbb{E}_Q[\ln(P_{\boldsymbol{\phi}}(\chi))]. \tag{9}$$

Taking the natural log of the factorized form of $P_{\boldsymbol{\phi}}$ gives us:

$$\ln(P_{\boldsymbol{\phi}}(\chi)) = \sum_{\phi \in \boldsymbol{\phi}} \ln \phi(U_\phi) - \ln Z. \tag{10}$$

Additionally, the entropy of a distribution is simply the expectation of its negative natural logarithm:

$$\mathbf{H}_Q(\chi) = \mathbb{E}_Q[\ln Q]. \tag{11}$$

Plugging these into the equation for $D(Q||P_{\boldsymbol{\phi}})$ we get:

$$D(Q||P_{\boldsymbol{\phi}}) = -\mathbf{H}_Q(\chi) - \mathbb{E}\left[\sum_{\phi \in \boldsymbol{\phi}} \ln \phi(U_\phi)\right] + \mathbb{E}_Q[\ln Z] \tag{12}$$

$$= -F(\widetilde{P}_{\boldsymbol{\phi}}, Q) + \ln Z. \tag{13}$$

□

The negative of the free energy functional $F(\widetilde{P}_{\boldsymbol{\phi}}, Q)$ is referred to in the statistical physics literature as the Helmholtz free energy. The functional is composed of two terms; the first is called the energy term and involves the expectation of the logarithms of factors in $\boldsymbol{\phi}$, while the second is called the entropy term and involves the entropy of Q. It is also worth noting that because $\ln Z$ does not depend on Q, we can minimize $D(Q||P_{\boldsymbol{\phi}})$ by maximizing the free energy functional.

4.2 The Mean Field Algorithm

Thus, we will focus ultimately on maximizing the free energy functional $F(\widetilde{P}_{\boldsymbol{\phi}}, Q)$. Additionally, because the K-L divergence is always nonnegative, $\ln Z \geq F(\widetilde{P}_{\boldsymbol{\phi}}, Q)$, which means that maximizing the free energy functional gives us a lower bound on the log partition function. We will use the mean field approximation, a method in the family of structured variational approaches, which aim to optimize the energy functional over a family $\mathcal{Q}$ of coherent or tractable distributions. In general this family is not expressive enough to capture all of the information in $P_{\boldsymbol{\phi}}$, but it will have the useful benefit of being simple enough to permit inference.

The mean field algorithm attempts to minimize $D(Q||P_{\boldsymbol{\phi}})$ over the space of distributions $\mathcal{Q}$ representable as the product of independent marginals:

$$Q(\chi) = \prod_i Q(X_i). \tag{14}$$

These distributions are clearly not particularly expressive; notably, they assume that the distributions over the different factors in the network are all independent. However, they are particularly tractable in that computing properties such as enthalpy and entropy are not exponentially costly. In addition, we may write down the set of fixed-point equations that characterize the stationary points of the mean-field optimization problem:

Theorem 2. *The distribution $Q(X_i)$ is a local maximum of the mean-field approximation problem given $\{Q(X_j)\}_{j\neq i}$ if and only if*

$$Q(x_i) = \frac{1}{Z} \exp\left(\sum_{\phi\in\boldsymbol{\phi}} \mathbb{E}_{\chi\sim Q}[\ln\phi|x_i]\right). \tag{15}$$

Proof. See [14], Section 11.5. □

Optimization via Lagrange multipliers leads to the message-passing mean-field approximation algorithm presented in Algorithm 1.

Algorithm 1 Mean-field approximation algorithm

procedure MEAN-FIELD($\boldsymbol{\phi}$, Q_0)
  $Q \leftarrow Q_0$
  Unprocessed$\leftarrow \chi$
  while *Unprocessed*$\neq \emptyset$ **do**
    Choose X_i from *Unprocessed*
    $Q_{old}(X_i) \leftarrow Q(X_i)$
    for $x_i \in \omega_{X_i}$ **do**
      $$Q(x_i) \leftarrow \exp\left\{\sum_{\phi:X_i\in U_\phi} \mathbb{E}_{(U_\phi-\{X_i\})\sim Q}[\ln\phi[U_\phi, x_i]]\right\}$$
    Normalize $Q(X_i)$ to sum to one
    if $Q_{old}(X_i) \neq Q(X_i)$ **then**
      $$\textit{Unprocessed}\leftarrow \textit{Unprocessed}\cup\left(\bigcup_{\phi:X_i\in U_\phi} U_\phi\right)$$
    Unprocessed$\leftarrow$ *Unprocessed*$-\{X_i\}$
  return Q

Note that the update procedure of $Q(X_i)$ in Algorithm 1 is to be done for each element in the domain of X_i. The energy functional consists of two terms, an energy term and an

entropy term. The energy term consists of a sum of terms of the form $\mathbb{E}_{U_\phi \sim Q}[\ln \phi]$ where we need to evaluate:

$$E_{U_\phi \sim Q}[\ln \phi] = \int_{U_\phi} Q(u_\phi) \ln \phi(u_\phi) \tag{16}$$

$$= \int_{u_\phi} \left(\prod_{X_i \in U_\phi} Q(x_i) \ln \phi(u_\phi) \right) \tag{17}$$

where u_ϕ is a particular assignment to members of U_ϕ, i.e., an element in the Cartesian product of each of their domains. Note that because we can compute $Q(u_\phi)$ as the product of marginals, the cost of evaluating the energy term is linear in the number of factors of $P_{\boldsymbol{\phi}}$. In our case, each factor corresponds to a single node (i.e. a clique of size one), and the product term in the integral is a product involving only one term.

Similarly, we can decompose the entropy term as:

$$\mathbf{H}_Q(\chi) = \sum_i \mathbf{H}_Q(X_i) \tag{18}$$

$$= \sum_i \mathbb{E}_{X_i \sim Q}[-\ln(Q(X_i))] \tag{19}$$

$$= \sum_i \left(\int_{\omega_{X_i}} -Q(x) \ln(Q(x)) \, \mathrm{d}x \right) \tag{20}$$

where ω_{X_i} is the continuous label space for the node X_i.

As a result, the energy functional for a fully factored distribution Q can be computed as a sum of expectations over small numbers of variables, and the complexity of this expression depends on the size of the factors in $P_{\boldsymbol{\phi}}$ instead of the topology of the network. So, even in networks that would require exponential time for computation of the partition function, the energy functional can be manipulated and computed efficiently.

Thus, upon termination of the mean-field approximation algorithm, computation of the free energy functional immediately gives us a lower bound on the log partition function.

5 Tree-Reweighted Belief Propagation

Similar to the mean-field approximation, the basic idea of TRBP is to approximate the original distribution (the "true" distribution) using a collection of tractable distributions where the partition functions of each of the tractable distributions can be feasibly computed. While SCMF simply picked one tractable distribution to use as a lower bound, TRBP instead considers these distributions as a collection, and uses a distribution over this collection to derive an upper bound on $\log Z$ using Jensen's inequality.

The collection of MRFs associated with a given graph G with vertex set V and edge set E constitutes an exponential family, any member of which is specified by an exponential parameter comprised of elements which are weights for potential functions defined on the graph cliques. Given a target distribution, the idea is to decompose that target distribution's exponential parameter as a convex combination of exponential parameters of a set of tractable distributions, and exploit the convexity of that combination to obtain an upper bound on the log partition function. Of course, choosing these this set of exponential parameters is a nontrivial task; fortunately, there is a variational problem that can be solved to obtain these parameters.

5.1 The Marginal Polytope

Suppose we have an indexing $\mathcal{I}$ of all the cliques in the graph; in this case, we denote the factor associated with the α-th clique as ϕ_α, and the set $\boldsymbol{\phi}$ of all such factors defines a vector-valued mapping from the total label domain of the MRF to $\mathbb{R}^d$, where d is the number of factors associated with the MRF. Associated with $\boldsymbol{\phi}$ is a vector $\theta \in \mathbb{R}^d$, known as the exponential parameter. The *exponential family* associated with $\boldsymbol{\phi}$ consists of the following family of MRFs parameterized by θ:

$$p(x;\theta) = \exp\{\langle \theta, \phi(x) \rangle - \Phi(\theta)\} \tag{21}$$

$$\Phi(\theta) = \log\left(\int \exp\{\langle\theta, \phi(x)\rangle\}\right) \tag{22}$$

where $\langle x, y\rangle$ denotes the standard inner product.

Brute-force computation of $\Phi(\theta)$ requires an integration over an exponentially large number of terms, which is impractical. Thankfully, the log partition function is convex as a function of the exponential parameters. Another important quantity is the conjugate dual function of Φ, defined by the optimization problem

$$\Phi^*(\mu) = \sup_{\theta\in\mathbb{R}^d} \{\langle\theta, \mu\rangle - \Phi(\theta)\} \tag{23}$$

where $\mu \in \mathbb{R}^d$ is a vector of dual variables. It can be shown that for a given dual vector μ^* this supremum is either equal to $+\infty$ or is realized at a vector θ^* such that for each α the following holds:

$$\mu^*_\alpha = \mathbb{E}_{\theta^*}[\phi_\alpha(x)] = \int p(x;\theta^*)\phi_\alpha(x) \tag{24}$$

Since these conditions involve taking an expectation, the components of μ^* must be realizable mean parameters; specifically, μ^* must belong to the relative interior of the marginal polytope, defined as:

$$\textsc{Marg}(\phi) = \{\mu \in \mathbb{R}^d \mid \exists\ p \text{ s.t. } \mathbb{E}_p[\phi(x)] = \mu\}. \tag{25}$$

This is to say we can think of μ^* as the expectation of θ^* under some distribution p.

It is worth taking a moment to discuss the importance of the marginal polytope. The duality between Φ and Φ^* defines a bijection between exponential parameters and mean parameters; thus, optimization over the space of exponential parameters amounts to optimization over the space of mean parameters. This provides us another way of interpreting SCMF and TRBP. The mean-field approximation performs an optimization over a subset of the marginal polytope (fully factorizable distributions), thus providing a lower bound on $\log Z$. In contrast, tree-reweighted belief propagation performs an optimization over a super-

set of the marginal polytope, as the pseudomarginals derived upon convergence only satisfy local marginalization properties, thereby providing an upper bound on $\log Z$.

5.2 Jensen's Inequality

We will principally be concerned with pairwise MRFs, in which the collection $\boldsymbol{\phi}$ consists solely of functions associated with individual nodes and single edges. The upper bounds on the log partition function are convex combinations of tree-structured distributions; specifically, we will only consider spanning trees of the graph.

Let $\mathcal{T}$ be the set of all spanning trees in the graph corresponding to our pairwise MRF; for each spanning tree $T \in \mathcal{T}$, let $\theta(T)$ be an exponential parameter that respects the structure of the tree T. In order for the distribution $p(x; \theta(T))$ to be tree-structured, $\theta(T)$ must belong to the constraint set

$$\mathcal{E}(T) = \{\theta(T) \mid \theta_\alpha(T) = 0 \ \forall \ \alpha \in \mathcal{I} \setminus \mathcal{I}(T)\} \tag{26}$$

where θ_α refers to the component of θ that weights the α-th clique. Let Θ be the full collection of tree-structured exponential parameter vectors; Θ is required to belong to the constraint set

$$\mathcal{E} = \{\Theta \mid \theta(T) \in \mathcal{E}(T) \ \forall \ T \in \mathcal{T}\} = \bigcup_{T \in \mathcal{T}} \mathcal{E}(T). \tag{27}$$

In order to produce the convex combinations we wish, we use a probability distribution $\vec{\rho}$ over the set of spanning trees

$$\vec{\rho} = \left\{ \rho(T), T \in \mathcal{T} \mid \rho(T) \geq 0, \sum_{T \in \mathcal{T}} \rho(T) = 1 \right\}. \tag{28}$$

We define the support of such a distribution $\text{supp}(\vec{\rho})$ as the set of trees to which $\vec{\rho}$ assigns strictly positive probability. A convex combination of exponential parameter vectors is given

by taking an expectation with respect to $\vec{\rho}$:

$$\mathbb{E}_{\vec{\rho}}[\theta(T)] = \sum_{T \in \mathcal{T}} \rho(T)\theta(T). \tag{29}$$

Recall that we are attempting to decompose a target distribution $\overline{\theta}$ into a convex combination; thus we are interested in collections of exponential parameters in which we can find convex combinations equal to $\overline{\theta}$. We define the feasible parameter set

$$\mathcal{A}(\overline{\theta}) = \left\{ (\theta, \vec{\rho}) \mid \mathbb{E}_{\vec{\rho}}[\theta(T)] = \overline{\theta} \right\}. \tag{30}$$

The convexity of Φ lets us apply Jensen's inequality to any convex combination $(\theta, \vec{\rho}) \in \mathcal{A}(\overline{\theta})$ in order to derive an upper bound on Φ:

$$\begin{aligned} \Phi(\overline{\theta}) &= \Phi(\mathbb{E}_{\vec{\rho}}[\Phi(\theta(T))]) && (31) \\ &\leq \mathbb{E}_{\vec{\rho}}[\Phi(\theta(T))] && (32) \\ &\leq \sum_{T \in \mathcal{T}} p(T)\Phi(\theta(T)). && (33) \end{aligned}$$

Note that this bound is a function both of the distribution $\vec{\rho}$ and the collection θ of tree-structured exponential parameters.

5.3 A Message-Passing Algorithm

First, we will consider how to optimize this upper bound with a fixed $\vec{\rho}$. Consider the constrained optimization problem

$$\begin{cases} \min\limits_{\theta \in \mathcal{E}} \mathbb{E}_{\vec{\rho}}[\Phi(\theta(T))] \\ \text{such that } \mathbb{E}_{\vec{\rho}}[\theta(T)] = \overline{\theta}. \end{cases} \tag{34}$$

A notable obstacle to directly attempting to solve this problem is that the exponential parameter θ has dimension equal to the number of spanning trees in the MRF, which can be intractably large.

Fortunately, the Lagrangian of this problem gives rise to a set of dual variables interpretable as *pseudomarginals* on the nodes and edges of the graph, so called because they need only satisfy local marginalization and normalization constraints. We use τ to refer to the set of pseudomarginals, which are density functions over unary and pairwise label spaces of the graph. We denote by $\tau_s(j)$ the value of the pseudomarginal at the point j, and by ω_s the label space of the node s. The single node entropy H_s is defined as:

$$H_s(\tau_s) = -\int_{\omega_s} \tau_s(j) \log \tau_s(j) \tag{35}$$

and the joint pseudomarginal entropy I_{st} is defined as:

$$I_{st}(\tau_{st}) = \int_{\omega_s \times \omega_t} \tau_{st}(j,k) \log \frac{\tau_{st}(j,k)}{\int_{k \in \omega_t} \tau_{st}(j,k) \int_{j \in \omega_s} \tau_{st}(j,k)}. \tag{36}$$

Then, we define the function $Q(\tau, \vec{\rho}_e)$:

$$Q(\tau, \vec{\rho}_e) = -\sum_{s \in V} H_s(\tau_s) + \sum_{(s,t) \in E} \rho_{st} I_{st}(\tau_{st}) \tag{37}$$

where $\vec{\rho}_e$ is a vector of edge appearance probabilities such that $\rho_{st} \in \vec{\rho}_e$ corresponds to the probability that the edge (s,t) appears in a spanning tree randomly selected according to the distribution $\vec{\rho}$. The spanning tree polytope $\mathbb{T}(G)$ consists of the set of all edge appearance probability vectors that represent permissible distributions of spanning trees over G. In this way we can avoid dealing explicitly with individual spanning trees and solely consider the probability any particular edge appears in a spanning tree chosen according to the distribution $\vec{\rho}$.

We also write down the inner product between τ and $\overline{\theta}$ as:

$$\langle t, \overline{\theta} \rangle = \sum_{s \in V} \int_{\omega_s} \tau_s(j) \theta_s(j) \, \mathrm{d}j + \sum_{(s,t) \in E} \int_{\omega_s} \int_{\omega_t} \tau_{s,tj,k} \theta_{st}(j,k) \, \mathrm{d}j \, \mathrm{d}k. \tag{38}$$

In turn, the following theorem provides an upper bound on Φ [24]:

Theorem 3. *For each fixed $\vec{\rho}$, the value of the best upper bound on the log partition function can be found by solving the variational problem:*

$$\Phi(\overline{\theta}) \leq \max_{\tau} \langle \tau, \overline{\theta} \rangle - Q(\tau; \vec{\rho}_e). \tag{39}$$

The TRBP algorithm consists of an iterative message-passing framework in which messages between nodes in the graph are repeatedly updated until convergence is achieved. We define M_{ts}^k as the message from node t to node s at the k-th iteration of the algorithm; the initial messages $M^0 = \{M_{ts}^0\}$ can be initialized to arbitrary positive real numbers. During each iteration, the messages are updated as follows:

$$M_{ts}^{n+1}(x_s) = \alpha \int_{\omega_t} \exp\left(\frac{\theta_{st}(x_s, X_t)}{\rho_{st}} + \theta_t(X_t) \right) \frac{\prod_{v \in \Gamma_t - \{s\}} (M_{vt}^n(X_t))^{\rho_{vt}}}{(M_{st}(X_t))^{1-\rho_{ts}}} \, \mathrm{d}X_t \tag{40}$$

where Γ_t is the set of neighbors of the node t and α is a normalization constant chosen such that $\int_{\omega_s} M_{ts}^{n+1}(X_s) \, \mathrm{d}X_s = 1$.

These messages can be used to define a set of unary and joint message-derived pseudomarginals. Denoting the probability of the edge $(i,j) \in E$ as ρ_{ij}, the singleton pseudomarginal τ_s can be computed as:

$$\tau_s(x_s) \propto \exp(\theta_s(x_s)) \prod_{v \in \Gamma_s} (M_{vs}(x_s))^{\rho_{vs}} \tag{41}$$

and the joint pseudomarginal τ_{st} can be written as:

$$\tau_{st}(x_s x_t) \propto \psi_{st}(x_s, x_t; \theta) \frac{\prod_{v \in \Gamma_s - \{t\}} (M_{vs}(x_s))^{\rho_{vs}}}{(M_{ts}(x_s))^{1-\rho_{st}}} \frac{\prod_{v \in \Gamma_t - \{s\}} (M_{vt}(x_t))^{\rho_{vt}}}{(M_{st}(x_t))^{1-\rho_{ts}}} \tag{42}$$

where

$$\psi_{st}(x_s, x_t; \theta) = \exp\left(\frac{\theta_{st}(x_s, x_t)}{\rho_{st}} + \theta_s(x_s) + \theta_t(x_t)\right). \tag{43}$$

The utility of this construction is provided by the following theorem:

Theorem 4. *For any valid $\vec{\rho}$, the message-derived pseudomarginals attain the global optimum of the variational problem posed in Theorem 3 when derived from a fixed point M^* of the message update procedure.*

Proof. See [24]. □

Thus, given a probability distribution $\vec{\rho}$ over the set of spanning trees, repeated iteration of the message update procedure until convergence allows us to compute an optimal upper bound on the log partition function.

Now, we consider the problem of constructing and optimizing a distribution over the spanning trees of G. We can represent any such distribution as a set of edge appearance probabilities through the vector $\vec{\rho}_e$. In particular, if $\vec{\rho}$ is the uniform distribution over all spanning trees of G, then the edge appearance probability of the edge $(i, j) \in E$ is equivalent to the resistance distance r_{ij} between the vertices i and j, which can be computed as:

$$r_{ij} = r_{ji} = M_{ii} + M_{ij} - M_{ij} - M_{ji} \tag{44}$$

where M is the Moore-Penrose inverse (pseudoinverse) of the graph Laplacian L [3].

These edge appearance probabilities can in turn be optimized in an iterative update scheme as follows. First, compute the minimum spanning tree S over G where the weight on the edge (i, j) is the negative mutual information N_{ij} between the nodes i and j, computable

as:

$$N_{ij} = \int_{\omega_s} \int_{\omega_t} \theta_{st}(x_s, x_t) \log \frac{\theta_{st}(x_s, x_t)}{\theta_s(x_s)\theta_t(x_t)} \, \mathrm{d}x_t \, \mathrm{d}x_s. \tag{45}$$

We use this tree S to define a descent direction vector $\vec{s}$, consisting of weights on each edge such that the weight on any edge is 1 if and only if that edge appears in S. Note that $|\vec{s}| = |\vec{\rho}_e| = |E|$. Then, if $(\vec{\rho}_e)^n$ is the edge probability vector at the nth iteration, set $(\vec{\rho}_e)^{n+1} = \alpha\vec{s} + (1-\alpha)(\vec{\rho}_e)^n$ where α is a step size parameter. For further details see [24].

Thus the optimization of the upper bound on the log partition function ultimately consists of two steps: optimization of the distribution over spanning trees, followed by optimization of the pseudomarginals via message-passing.

6 Reproducing Kernel Hilbert Spaces

In MRFs with discrete label spaces (dMRFs), distributions over unary or pairwise conformational spaces are easily represented as vectors with each component corresponding to a particular label in the label space. However, in MRFs with continuous label spaces (cMRFs), these distributions cannot be represented as vectors. To put it another way, the space of distributions over a discrete label space with dimension n is isomorphic to the unit ball in $\mathbb{R}^n$ with the ℓ_1 norm. In contrast, the space of distributions over a continuous label space is an infinite-dimensional space of functions. Thus, in order to perform message-passing algorithms that involve manipulations of distributions over the MRF label spaces, we need a mechanism of manipulating distributions over a continuous space that is as expressive as possible without being computationally cumbersome. In order to do this we will be using reproducing kernel Hilbert space representations of distributions.

6.1 Constructing the RKHS

First, we provide some background as to how reproducing kernel Hilbert spaces are constructed. Given some continuous domain Ω, a kernel k is a symmetric function $\Omega \times \Omega \to \mathbb{R}$ and is called a positive definite kernel if, for any square-integrable functions f_1 and f_2 the following holds:

$$\int_\Omega \int_\Omega f_1(x) k(x, y) f_2(y) \, \mathrm{d}x \, \mathrm{d}y \geq 0. \tag{46}$$

One example of a positive-definite kernel is the Gaussian kernel with variance σ:

$$k(x, y) = \exp\left(-\frac{||x - y||^2}{\sigma^2}\right). \tag{47}$$

For any $x \in \Omega$, we denote by ϕ_x the function $\Omega \to \mathbb{R}$ such that $\phi_x(y) = k(x, y)$; we call this the *feature map* at the point x.

A *reproducing* kernel is a kernel that satisfies the reproducing kernel property:

$$k(x, x') = \langle k(x, \cdot), k(\cdot, x') \rangle = \langle \phi_x, \phi_{x'} \rangle \tag{48}$$

where ϕ_x is the feature map at the point x. The Gaussian kernel, for example, is a reproducing kernel.

In order to construct the RKHS associated with a kernel k, we consider the set $\mathcal{F}_0$ of all linear combinations of feature maps of points in Ω. That is, let

$$\mathcal{F}_0 = \left\{ f(y) = \sum_{i=1}^{n} \alpha_i \phi_{x_i}(y) \mid n \in \mathbb{N}, \alpha_i \in \mathbb{R}, x_i \in \Omega \right\}. \tag{49}$$

Endowing $\mathcal{F}_0$ with the operations of addition

$$(f + g)(x) = f(x) + g(x) \tag{50}$$

and scalar multiplication

$$(\lambda f)(x) = \lambda f(x), \lambda \in \mathbb{R}, \tag{51}$$

and introducing the inner product between two elements of the space $f = \sum_{i=1}^{n} \alpha_i \phi_{x_i}$ and $g = \sum_{i=1}^{m} \beta_i \phi_{y_i}$ $(\alpha_i, \beta_i \in \mathbb{R} \ \forall \ i)$ as:

$$\langle f, g \rangle = \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \beta_j k(x_i, y_j) \tag{52}$$

makes $\mathcal{F}_0$ an inner product space.

We can construct the corresponding RKHS $\mathcal{F}$ via the set of equivalence classes of Cauchy sequences in $\mathcal{F}_0$. Let $C[\Omega]$ be the collection of all Cauchy sequences in $\mathcal{F}_0$. Define an equivalence relation $\sim$ via

$$(x_n) \sim (y_n) \Leftrightarrow \lim_{n \to \infty} d(x_n, y_n) = 0 \tag{53}$$

where $d(x_n, y_n) = \sqrt{\langle x_n, y_n \rangle^2}$. Then, $\mathcal{F}$ is the set of all equivalence classes of $C[\Omega]$ with respect to $\sim$ [11].

This is to say that if we have a reproducing kernel k over some domain Ω then the corresponding RKHS $\mathcal{F}$ is the linear span of the set of all feature maps of k. By the Moore-Aronszajn theorem every reproducing kernel uniquely specifies a corresponding RKHS and vice-versa [2]. Thus, it is worth considering the relationship between the linear span of a particular kernel and the space of all functions over its domain.

We first consider the *mean map* μ_f of a function f:

$$\mu_f = \int_\Omega \phi_x f(x) \, \mathrm{d}x. \tag{54}$$

where ϕ_x is the feature map of the kernel k at the point x. If this map is injective, i.e. $\mu_f \neq \mu_g \Rightarrow f \neq g$, then we say that the kernel k is *characteristic*. If $\mathcal{C}(\Omega)$ is the space of all continuous bounded functions on Ω, if ϕ_x is continuous for all x, and if the corresponding RKHS $\mathcal{F}$ is dense in $\mathcal{C}(\Omega)$, then the kernel k is considered *universal*. The Gaussian kernel, for example, is both characteristic and universal. One of the benefits of characteristic kernels is that the linear span of feature maps of that kernel is dense in the space of real-valued continuous functions over Ω with the supremum norm, meaning that the linear span of feature maps is a reasonably expressive approximation to the broader function space we are interested in representing [23].

6.2 RKHS Representations of Functions

We now consider how to construct RKHS representations of functions. We consider the problem of approximating a function f over a domain Ω as a linear combination of n feature maps, where n is a chosen parameter. Let k be a kernel on Ω, and denote by ϕ_x the feature map at the point $x \in \Omega$. We wish to compute a linear combination g of feature maps that allows us to best approximate the values of f on Ω.

First, if we are given a set of n feature maps $\phi_{x_1}, \ldots, \phi_{x_n}$ such that $g = \sum_{i=1}^{n} \alpha_i \phi_{x_i}$ where $\alpha_i \in \mathbb{R}$, then we can compute the optimal α_i values to approximate f as the unique solution to the linear system

$$\sum_{i=1}^{n} K(x_i, x_j)\alpha_i = f(x_j), 1 \leq j \leq n[25]. \tag{55}$$

So, in order to optimally represent f, we need to identify the n feature maps that will allow us to best approximate f over Ω, and then compute the associated linear coefficients. While there is a method of computing an optimal set of feature maps (these correspond to eigenfunctions of a particular integral operator) [25], in practice we simply perform a uniform gridding of the conformational space we are interested in. This gridding is feasible in large part because the message-passing algorithms we will be using only consider unary and pairwise conformational spaces, as opposed to the exponentially large overall conformational space of the protein.

6.3 The Mean Map

Given an RKHS representation of a function, we now consider how we manipulate the function representation; specifically, we will mainly be concerned with either taking expectations of functions or computing integrals over volumes of space. Both of these operations can be done using the mean map mentioned earlier.

Computing the mean map via the integral in Equation 54 is often excessively costly. Thus, a sampling-based approximation to the mean map, called the *empirical mean map*, is often used. Let p be a probability distribution over a domain Ω. Suppose we sample $x_1, \ldots, x_m$ i.i.d. from Ω according to p. We define the empirical mean map $\widehat{\mu_p}$ as:

$$\widehat{\mu_p} = \sum_{i=1}^{m} \frac{1}{m} \phi_{x_i}. \tag{56}$$

The i.i.d. sampling can be performed as follows. Suppose $\Omega = [0, a]^n$ where $a < \infty$. Let $S = [0, b]^n$ where $b \leq a$ be a subset of Ω. If X is a random variable distributed according

to f, we can compute $P(X \in S)$. First, let ν be the uniform distribution on Ω with mean embedding μ_ν, and let σ be the uniform distribution on S with mean embedding μ_σ. Then,

$$P(X \in S) \approx \frac{b^n \langle \mu_\sigma, f \rangle}{a^n \langle \mu_\nu, f \rangle}. \tag{57}$$

Assuming we have a RKHS representation of f, taking the inner products is relatively easy. Then, we can sample from f by uniformly sampling a point q from $[0,1]^n$, and then searching Ω to find the point x such that $|f(x) - q| \leq \varepsilon$ for some sufficiently small ε. Because CDFs are monotonically increasing functions, we can, for example, perform a binary search over Ω to compute the relevant x. This search isn't done over the entire conformational space, but just the domain of the function we are trying to fit, which will typically involve only single residue conformational spaces.

The mean map has the useful property that the expectation of any function $f \in \mathcal{F}$ can be calculated by taking the inner product in $\mathcal{F}$: $\langle \mu_p, f \rangle = \mathcal{E}_p[f] \; \forall \; f \in \mathcal{F}$.

It is worth noting that if p is just the uniform distribution on Ω, then μ_p can be estimated by uniformly sampling Ω. Then, for any function f on Ω we have

$$\langle \mu_p f \rangle = \mathbb{E}_p[f] = \int_\Omega f(X) p(X) dX = \frac{1}{|\Omega|} \int_\Omega f(X) dX. \tag{58}$$

Note here that the value of the inner product is the integral of the function over the domain divided by the size of the domain, which can be easily calculated. This is to say that the reproducing kernel Hilbert space structure, specifically the nature of the mean map, lets us efficiently compute integrals of functions over the domain. This is particularly useful for message-passing algorithms such as TRBP, which involve an integration during each message update.

7 Results and Discussion

Both SCMF and TRBP were implemented in the Donald lab's open-source protein design software suite OSPREY, and were used to calculate bounds on $\log Z$ for synthetic examples as well as protein systems found in the Protein Data Bank (PDB). In general partition function computations are not performed on the entire protein, but rather on a subset of amino acid residues deemed to be important for design considerations; computing the partition function while including every single residue, even in the rigid case, is computationally unfeasible. Data from running the SCMF and TRBP algorithms on a test set of 30 proteins is presented, along with an examination in greater detail of the computational results from one of those proteins.

7.1 Bounds on $\log Z$

Partition function computations were performed on test set of 30 protein structures, which included the protein structures with PDB IDs 1A0R, 1AMU, 1B6C, 1B74, 1GWC, 1TP5, 2HNU, 2HNV, 2P49, 2P4A, 2Q1E, 2Q2A, 2RF9, 2RFD, 2RFE, 2RL0, 2WZP, 2XGY, 2XQQ, 2XXM, 3BU8, 3BUA, 3CAL, 3EB6, 3GXU, 3K3Q, 3MA2, 3RJQ, 3U7Y, and 4LAJ. Partition function bounds were computed for each of these proteins, with six residues in the protein chosen as flexible. Thus the conformational space corresponding to the computed partition function for any given protein is the space of all possible conformations of the six chosen amino acids. A list of all protein structures, PDB IDs, and mutable residues is shown in Table 1. Six-node MRFs were then set up for each partition function problem, and both SCMF and TRBP were used to compute upper and lower bounds (respectively) on $\log Z$.

Lower bounds on $\log Z$, upper bounds on $\log Z$, and the value of $\log Z$ according to iMinDEE/K^* are shown in Figure 3; values are listed in Table 2. The iMinDEE/K^* algorithm computes an ε-approximation to the partition function, where ε is a parameter set by the user prior to the beginning of the computation. For the purposes of these computations

ε was set to 0.68, in accordance with earlier work by the Donald lab using this same test set of proteins [17]. Calculation of the $\log Z$ upper bound for proteins with PDB IDs 1A0R, 1GWC, 2P4A, 3EB6, and 3RJQ failed to converge; thus, the upper bound for these proteins is not graphed in Figure 3. In general the lower bounds produced by SCMF ranged from approximately 7 to 15, while the upper bounds ranged from approximately 25 to 30.

Table 1: List of PDB IDs, protein structure names, and mutable residues for every test design performed.

PDB ID	Protein Structure Name	Mutable Residues
1A0R	Heterotrimeric complex of posducin/transducin beta-gamma	ASN-313, TRP-332, LEU-605, GLU-696
1AMU	Phenylalanine-activating domain of gramicidin synthetase 1 in a complex with amp and phenylalanine	GLU-374, ASP-413, ILE-429, GLU-441
1B6C	Crystal structure of the cytoplasmic domain of the type I TGF-Beta receptor in complex with FKBP12	ASP-37, TRP-59, LEU-195, LEU-196
1B74	Glutamate racemase from Aquifex pyrophilus	PHE-57, LEU-220, LEU-222, PHE-224
1GWC	The structure of a tau class glutathione s-transferase from wheat, active in herbicide detoxification	HID-53, CYS-67, PHE-100, TYR-104
1TP5	Crystal structure of PDZ3 domain of PSD-95 protein complexed with a peptide ligand KKETWV	PHE-325, ASN-326, GLU-422, THR-423
2HNU	Crystal Structure of a Dipeptide Complex of Bovine Neurophysin-I	THR-194, LEU-269, ILE-309, HIE-317
2HNV	Crystal Structure of a Dipeptide Complex of the Q58V Mutant of Bovine Neurophysin-I	LYS-255, ASP-267, LYS-336, GLU-349
2P49	Complex of a camelid single-domain vhh antibody fragment with RNASE A at 1.4A resolution: native mono_1 crystal form	ASN-66, TYR-68, TYR-140, ILE-153
2P4A	X-ray structure of a camelid affinity matured single-domain vhh antibody fragment in complex with RNASE A	LEU-135, SER-141, VAL-381, SER-386
2Q1E	Altered dimer interface decreases stability in an amyloidogenic kappa1 Bence Jones protein.	LEU-266, GLN-309, LEU-423, THR-426

2Q2A	Crystal structures of the arginine-, lysine-, histidine-binding protein ArtJ from the thermophilic bacterium Geobacillus stearothermophilus	PHE-144, GLU-145, GLU-394, ASN-397
2RF9	Crystal structure of the complex between the EGFR kinase domain and a Mig6 peptide	SER-337, GLN-350, SER-888, GLN-911
2RFD	Crystal structure of the complex between the EGFR kinase domain and a Mig6 peptide	PHE-352, TYR-358, TYR-920, ILE-929
2RFE	Crystal structure of the complex between the EGFR kinase domain and a Mig6 peptide	PHE-352, TYR-358, LYS-925, MET-928
2RL0	Crystal structure of the fourth and fifth fibronectin F1 modules in complex with a fragment of staphylococcus aureus fnbpa-5	PHE-156, THR-193, PHE-649, GLU-651
2WZP	Structures of lactococcal phage p2 baseplate shed light on a novel mechanism of host attachment and activation in siphoviridae	TRP-207, THR-242, SER-630, ASN-614
2XGY	Complex of rabbit endogenous lentivirus (relik)capsid with cyclophilin a	HIE-76, LEU-89, PHE-189, HIE-255
2XQQ	Human dynein light chain (DYNLL2) in complex with an in vitro evolved peptide (Ac-SRGTQTE).	THR-4, THR-6, ASN-61, PHE-62
2XXM	Crystal structure of the hiv-1 capsid protein c-terminal domain in complex with a camelid vhh and the cai peptide.	THR-2, PHE-3, TYR-169, ARG-173
3BU8	Crystal Structure of TRF2 TRFH domain and TIN2 peptide complex	GLN-105, ASP-117, SER-257, PHE-258
3BUA	Crystal Structure of TRF2 TRFH domain and APOLLO peptide complex	GLN-84, SER-119, LEU-506, THR-507
3CAL	Crystal structure of the second and third fibronectin F1 modules in complex with a fragment of staphylococcus aureus fnbpa-5	THR-3, ILE-40, THR-100, THR-101
3EB6	Structure of the cIAP2 RING domain bound to UbcH5b	VAL-559, MET-561, GLU-1109, ASP-1112
3GXU	Crystal structure of Eph receptor and ephrin complex	GLN-43, ILE-135, GLN-618, TRP-625
3K3Q	Crystal Structure of a Llama Antibody complexed with the C. Botulinum Neurotoxin Serotype A Catalytic Domain	SER-100, ASP-102, TYR-349, PHE-357
3MA2	Complex membrane type-1 matrix metalloproteinase (MT1-MMP) with tissue inhibitor of metalloproteinase-1 (TIMP-1)	ASP-274, GLN-281, TYR-338, GLU-367
3RJQ	Crystal structure of anti-HIV llama VHH antibody A12 in complex with C186 gp120	HIE-105, ASP-113, SER-629, LYS-695

3U7Y	Structure of NIH45-46 Fab in complex with gp120 of 93TH057 HIV	ASN-280, THR-467, ARG-561, TRP-602
4LAJ	Crystal structure of HIV-1 YU2 envelope gp120 glycoprotein in complex with CD4-mimetic miniprotein, M48U1, and llama single-domain, broadly neutralizing, co-receptor binding site antibody, JM4	LYS-421, MET-434, TYR-632, THR-702

In general it was not possible to get convergence of the $\log Z$ upper bounds past a certain level of precision; initial testing seemed to indicate that the bound would not converge to a particular value but would rather oscillate indefinitely within a window of approximately 0.3 in either direction, but would never settle. Thus, the convergence criterion was set to 0.5, so if the bounds in any two consecutive iterations were within 0.5 of each other the algorithm was terminated. This phenomenon was initially surprising, but after some consideration we found a possible explanation. Belief propagation and other variational methods are traversals of manifolds representing spaces of distributions [1]. If these distributions are over discrete spaces, the manifolds have finite dimension. TRBP and SCMF, in turn, are performing gradient descent and ascent, respectively, with respect to the partition function over a manifold of distributions. In our case, this manifold has infinite dimension, but because our representations of continuous functions are necessarily finite, we are in a sense only able to approximate the gradient at each iteration. In turn, near the optimum of the function at which convergence will be expected, the algorithm may be unable to properly step along the gradient without stepping over the optimum and worsening the bound on $\log Z$. The algorithm will then attempt to step back, but imprecisely, and the process will repeat.

Table 2: $\log Z$ lower and upper bounds and K^* scores for all test proteins. In cases where TRBP failed to converge, no upper bound is listed.

Protein	$\log Z$ lower bound	$\log Z$ upper bound	$\log Z$ from K^*
1A0R	10.61	-	11.37

1AMU	11.89	29.84	13.23
1B6C	9.34	25.86	11.97
1B74	10.04	27.36	20.63
1GWC	9.77	-	13.15
1TP5	10.93	27.58	13.61
2HNU	9.43	29.45	11.88
2HNV	11.91	29.61	6.96
2P49	11.99	24.19	16.74
2P4A	8.66	-	12.24
2Q1E	10.61	27.10	13.32
2Q2A	11.97	24.81	12.81
2RF9	11.77	28.39	10.19
2RFD	11.79	26.80	17.77
2RFE	11.56	28.43	15.85
2RL0	10.62	26.63	12.18
2WZP	10.01	25.93	13.59
2XGY	10.01	25.86	15.80
2XQQ	9.24	29.32	10.18
2XXM	11.72	27.94	17.14
3BU8	10.94	30.80	6.65
3BUA	11.01	30.54	13.54
3CAL	10.04	29.44	11.11
3EB6	10.17	-	16.98
3GXU	11.95	28.15	17.28
3K3Q	10.73	28.07	15.18
3MA2	12.93	27.48	11.74
3RJQ	10.11	-	10.88
3U7Y	10.80	26.13	8.90
4LAJ	12.94	29.25	17.05

The bounds produced by algorithms on cMRFs are frequently quite loose; however, it is difficult to determine the cause of this. While the bounds produced by SCMF and TRBP are valid bounds upon convergence of the messages in that the value of the partition function is guaranteed to be no less than the lower bound and no greater than the upper bound, there are no guarantees with regards to the quality of those bounds; additionally, TRBP is not guaranteed to converge, but rather only provides valid upper bounds upon convergence. In the case of protein design, this is quite possibly due to the nature of the inference algorithms. Specifically, the mean-field approximation solely considers fully-factorizable distributions,

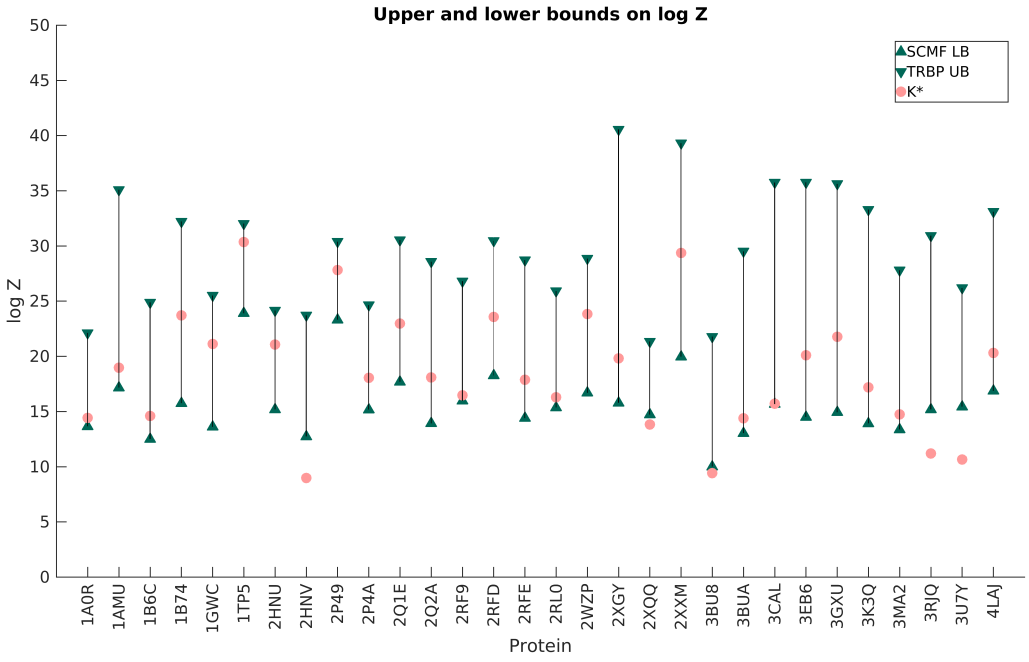

Figure 3: Bounds on $\log Z$ for protein structures computed by SCMF and TRBP, along with calculated iMinDEE/K^*.

and tree-reweighted belief propagation passes messages over pairwise Markov random fields. However, the iMinDEE/K^* algorithm performs full n-body minimization over the entire protein structure, and in this way explicitly includes higher-order terms in the partition function calculation process. It is possible that incorporating higher-order interactions into design algorithms will enable more accurate bounds on $\log Z$; such a trend has been demonstrated in previous work by the Donald lab [19].

In general, the SCMF lower bound was closer to the $\log Z$ value calculated by iMinDEE/K^* than the TRBP upper bound; the lone exception to this has PDB ID 1B74. Since SCMF bounds $\log Z$ by solely considering distributions that can be factorized into a product of marginals, this would seem to suggest that for this protein the biophysical interactions are such that the overall Boltzmann distribution is not easily expressible as a product of single-residue distributions. However, in general, the closer accuracy of SCMF seems to indicate that protein design problems can be well-approximated by "reducing" complex high-order

distributions to products of marginal distributions or lower-order factors. This strategy has been employed in the past for optimization of the rigid-rotamer GMEC model of protein design [22].

Interestingly, for 5 proteins (2HNV, 2RF9, 3BU8, 3MA2, and 3U7Y) the lower bound produced by SCMF is greater than the value calculated via iMinDEE/K^*. This is possibly due to the manner in which pairwise energy functions used by iMinDEE/K^* are calculated. The computed pairwise energy function, used both by iMinDEE/K^* and SCMF/TRBP, is a lower bound on the interaction energy between the two amino acids [19]. This is done so that iMinDEE/K^* can use that energy as a lower bound during the conformation enumeration process [10].

The actual inter-residue interaction energy function is only known after the process of full energy minimization is performed; in fact, computing the true function requires integrating out the energetic contributions of all other amino acids in the protein. Thus, the energy functions used by SCMF/TRBP are lower bounds on the energy; in turn, this means that the exponential of the negative energy is upper bounded over the entire conformational domain. Thus, the value of the partition function is overestimated as well; this overestimation phenomenon also provides a possible explanation of why the SCMF lower bounds were generally closer to the value of $\log Z$ computed by iMinDEE/K^* than the TRBP upper bounds.

Additionally, there is a second possible explanation for this phenomenon. Consider a protein design problem with a protein and a ligand, each with one residue allowed to be flexible. Suppose additionally that the energy of the bound state is always -2, while the energy of the unbound state of both the protein and the ligand is always -1. Last, suppose that the flexible residue in both the protein and ligand has one χ-angle, which can vary between 0° and 20°. Let Z_{AB} be the partition function for the bound complex, Z_A be the partition function for the unbound protein, and Z_B be the partition function for the unbound ligand. From the perspective of iMinDEE/K^*, the binding constant would be approximated

as follows:

$$k_a = \frac{1}{k_d} \tag{59}$$
$$= \frac{Z_{AB}}{Z_A Z_B} \tag{60}$$
$$\approx \frac{e^2}{(e)(e)} \tag{61}$$
$$\approx 1. \tag{62}$$

Integrating over the conformational domain, in turn, gives us:

$$k_a = \frac{1}{k_d} \tag{63}$$
$$= \frac{Z_{AB}}{Z_A Z_B} \tag{64}$$
$$= \frac{\int_0^{20}\int_0^{20} e^2}{\int_0^{20} e \int_2^{20} e} \tag{65}$$
$$= \frac{400e^2}{(20e)(20e)} \tag{66}$$
$$= 1. \tag{67}$$

The binding constants here agree with each other, and yet while iMinDEE/K^* approximates Z_{AB} as e^2, integrating over the conformational domain gives us $Z_{AB} = 400e^2$. This is to say that integrating over a continuous domain gives us a different answer than taking the sum of the minima over each domain. While it would be useful to test this out on our proteins, the bounds on the partition function generated by SCMF and TRBP are too loose to allow us to compute the binding constant to any reasonable degree of precision.

7.2 HIV Envelope Glycoprotein gp120

In general we use RKHS representations of functions throughout the message passing algorithm process in order to represent messages, energy functions, and probability density

functions. For example, consider the results for a partition function computation of the structure of the HIV envelope glycoprotein gp120 in complex with a llama antibody fragment (PDB ID: 4LAJ). The conformational domain of four residues was considered: Lys-421, Met-434, Tyr-632, and Thr-702. A view of the interface, with these residues highlighted, is shown in Figure 4. For this particular system, the lower bound on $\log Z$ was 12.94, while the upper bound on $\log Z$ was 29.25; the value of $\log Z$ computed by iMinDEE/K^* was 17.05.

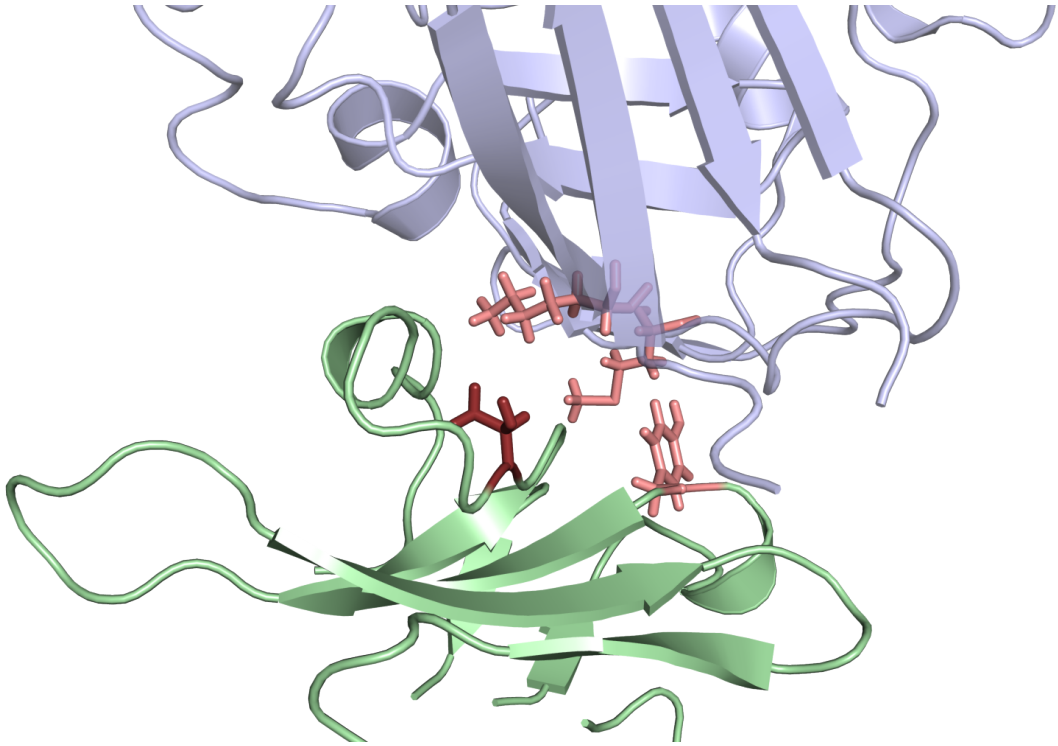

Figure 4: View of the interface between HIV glycoprotein gp120 and an antibody fragment. The Threonine residue at position 702 is shown in dark red; other residues whose conformational spaces were considered are shown in light red. The rest of the HIV glycoprotein is shown in purple, and the rest of the antibody fragment is shown in green.

The marginal density computed by SCMF for one region of the unary conformational space of the Threonine residue at position 702 is shown in Figure 5, and the corresponding pseudomarginal computed by TRBP is shown in Figure 6. In general the probability at each point in the conformational domain is quite low, most directly because the conformational space has total area 364 square degrees, and even a uniform distribution over the space would

have low probability densities at each point in the domain. The SCMF marginal exhibits significant nonconvexity, with peaks at roughly $(-180°, -170°)$, $(-175°, -175°)$, $(-180°, -155°)$, and $(-155°, -170°)$. In contrast, the TRBP marginal features one extremely prominent peak at $(-180°, -155°)$. Notably this peak is present in both marginals, although much less prominently in the SCMF marginal density. There is a secondary TRBP peak at $(-175°, -170°)$, which is itself reasonably close to one of the peaks in the SCMF density. While it is not immediately clear how the difference in these two marginals contributes to the upper- and lower- bounding properties, it is interesting to note that the SCMF marginal seems to spread out the probability mass relative to the TRBP marginal.

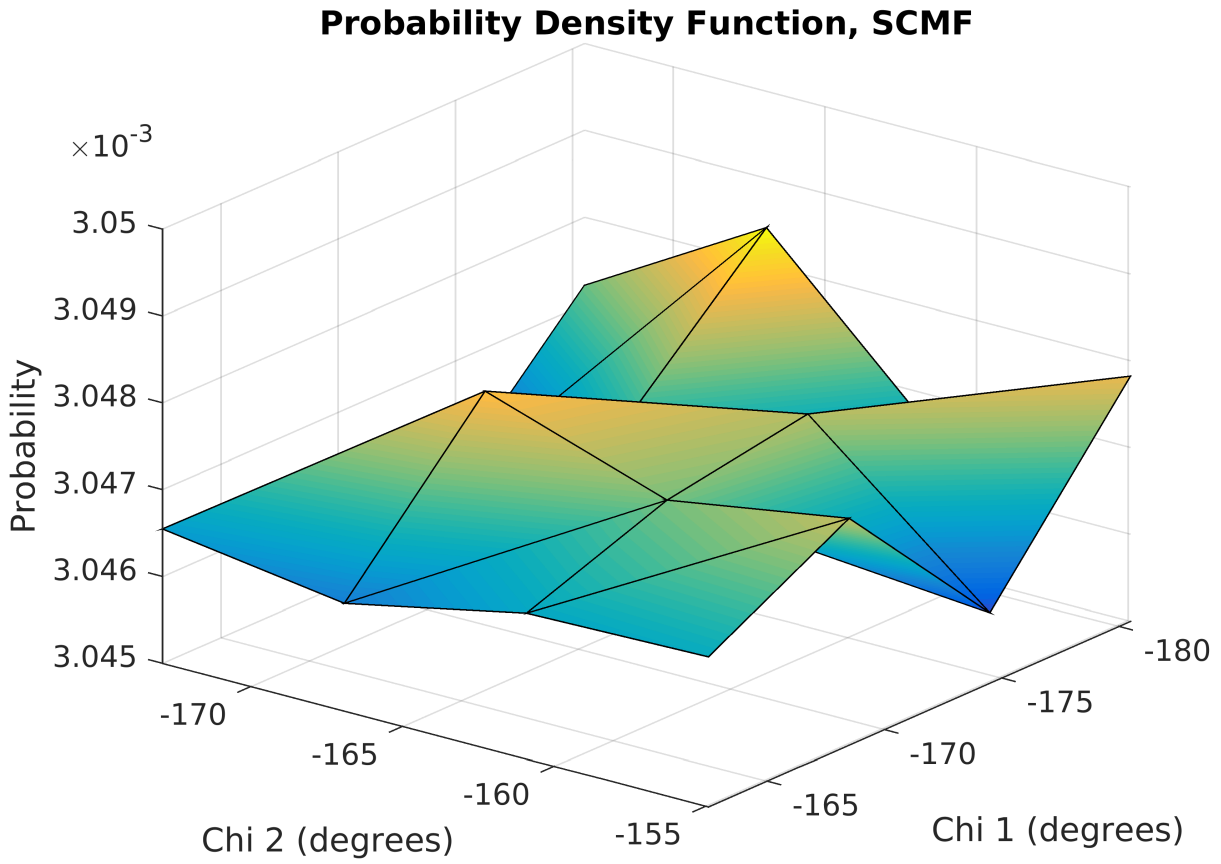

Figure 5: Marginal density computed for one region of the conformational space of the Threonine residue at position 702, HIV envelope protein gp120 in complex with an antibody fragment (PDB ID: 4LAJ).

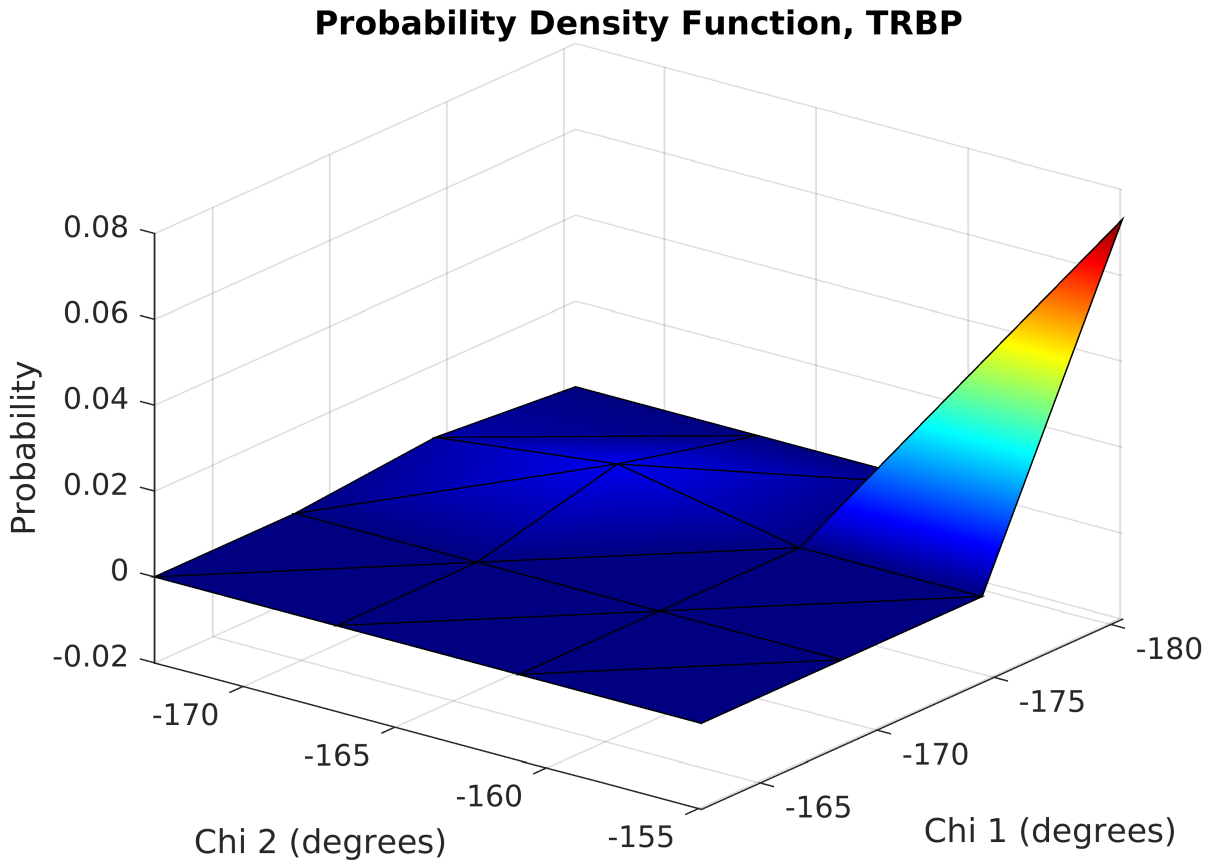

Figure 6: Pseudomarginal computed for one region of the conformational space of the Threonine residue at position 702, HIV envelope protein gp120 in complex with an antibody fragment (PDB ID: 4LAJ).

It is tempting to view the plots in Figures 5 and 6 as plots of Boltzmann-weighted probability distributions; however, strictly speaking, this would be inaccurate, as the TRBP pseudomarginal only needs to satisfy local marginalization constraints. There is evident non-convexity within the SCMF marginal, suggesting that the energy landscapes involved in protein design problems are rugged even within the individual voxel spaces defined by the iMinDEE algorithm. Thus, treating all points within a voxel as contribution uniformly to the partition function, as minimization-based algorithms do, may not be accurate. This non-convexity also highlights the important of effective conformation minimization processes in design algorithms. Given the difficulty and slow speed of accurate n-body minimization

in the conformation enumeration process, this only serves to further highlight the potential of variational approaches for computing ensemble properties of protein structure.

One significant opportunity for further work is the absence of a multi-sequence bound, or any clear method of introducing sub-linearity into the computation process. This is to say that if one is considering k possible sequences and wishes to identify the sequence with the largest partition function, with cMRFs it is necessary to perform all the relevant computations for each different sequence. In contrast, algorithms that are sub-linear in the number of sequences often use a multi-sequence bound in order to enable the use of A* search over a sequence space to reduce computation time [17]. Ultimately this means that the algorithms as presented in this thesis will likely scale poorly to scenarios where a protein designer is attempting to choose the best of several hundred or thousand sequences; in such cases the development of a sub-linear version of TRBP and SCMF for the continuous-domain case will likely be needed. In particular, this would necessitate the computation of a lower bound on the partition function for any protein sequence. For example, suppose a designer is considering 3 residue positions and the first two are defined. We would wish to compute a lower bound on the partition function for any sequence with those first two residues no matter how the third is chosen.

7.3 Discussion

Computation of ensemble properties and inclusion of continuous flexibility have been shown to significantly improve the accuracy of protein design predictions. However, the cost of both of these features is significantly increased computation time: ensemble properties involve calculations and optimizations over significantly larger spaces than single-conformation computations, while continuous flexibility involves an expensive process of conformation enumeration and minimization. Here we have attempted to solve these problems by using inference algorithms on Markov random fields with continuous label spaces in order to avoid enumeration or consideration of individual conformations, while simultaneously using

message-passing algorithms to bound the partition function while minimizing computation time. The algorithms as implemented often produced bounds that were quite loose, although they were consistent with each other; it is not entirely clear if tightening these bounds is a matter of significant software engineering, or if the kinds of graphical models produced by protein design problems are not amenable to variational inference of the kind discussed in this work. Additionally, the algorithms in this work are not intrinsically sub-linear, meaning that they will scale poorly to scenarios where large numbers of possible protein sequences are being considered. However, this work does make two contributions that lay the groundwork for future projects: (1) a demonstration of the feasibility of performing inference on Markov random fields with continuous label spaces, and (2) computations of ensemble properties with both continuous flexibility and continuous entropy.

8 Acknowledgements

There are a number of people to thank; for starters, the list includes my brother and parents.

I would like to thank my advisor, Prof. Bruce Donald, as well as my thesis committee members Prof. Raluca Gordân and Prof. Cynthia Rudin. I would also like to thank Goke Ojewole, whom I worked closely with towards the end of this project and whose help was instrumental in completing this work.

Certainly this would have been impossible without the help of all others who worked in the Donald lab while I was there: Hunter, Pablo, Marcel, Mark, François, JJ, Anna, Yang, Jeff, Chanelle, Kyle, David, Graham, Divya, and Rachel.

This work was funded in part by a summer undergraduate research fellowship from the Duke Undergraduate Research Support Office, and in part by the National Institute of General Medical Sciences (NIGMS), National Institutes of Health (NIH) under award number R01GM78031.

Last, I would like to acknowledge those others who helped in ways small and large: Gordon, Michael, Ish, Lenny, Brad, Jaleelah, Nate, James, Kira, Elias, Rich, Charles, Ed, Syd, Caroline, Collin, Madhu, John, Kshithij, Mark, PJ, and Jorge.

References

[1] S. Amari and H. Nagaoka. *Methods of Information Geometry*, volume 191 of *Translations of Mathematical monographs*. Oxford University Press, 2000.

[2] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404, 1950.

[3] R. B. Bapat, I. Gutmana, and W. Xiao. A Simple Method for Computing Resistance Distance. *Zeitschrift Naturforschung Teil A*, 58:494–498, October 2003.

[4] Cheng-Yu Chen, Ivelin Georgiev, Amy C. Anderson, and Bruce R. Donald. Computational structure-based redesign of enzyme activity. *Proceedings of the National Academy of Sciences*, 106(10):3764–3769, 2009.

[5] Bassil I. Dahiyat and Stephen L. Mayo. De novo protein design: Fully automated sequence selection. *Science*, 278(5335):82–87, 1997.

[6] Bruce R. Donald. *Algorithms in Structural Molecular Biology*. The MIT Press, 2011.

[7] Pablo Gainza, Hunter M Nisonoff, and Bruce R Donald. Algorithms for protein design. *Current Opinion in Structural Biology*, 39:16 – 26, 2016. Engineering and design Membranes.

[8] Pablo Gainza, Hunter M Nisonoff, and Bruce R Donald. Algorithms for protein design. *Current Opinion in Structural Biology*, 39:16 – 26, 2016. Engineering and design Membranes.

[9] Pablo Gainza, Kyle E. Roberts, and Bruce R. Donald. Protein design using continuous rotamers. *PLoS Comput Biol*, 8(1):1–15, 01 2012.

[10] I. Georgiev, R. H. Lilien, and B. R. Donald. The minimized dead-end elimination criterion and its application to protein redesign in a hybrid scoring and search algorithm for computing partition functions over molecular ensembles. *J Comput Chem*, 29(10):1527–1542, Jul 2008.

[11] Hal Daum III. From zero to reproducing kernel hilbert spaces in twelve pages or less. February 2004.

[12] J. D. Jou, S. Jain, I. S. Georgiev, and B. R. Donald. BWM*: A Novel, Provable, Ensemble-based Dynamic Programming Algorithm for Sparse Approximations of Computational Protein Design. *J. Comput. Biol.*, 23(6):413–424, Jun 2016.

[13] Hetunandan Kamisetty, Arvind Ramanathan, Chris Bailey-Kellogg, and Christopher James Langmead. Accounting for conformational entropy in predicting binding free energies of protein-protein interactions. *Proteins: Structure, Function, and Bioinformatics*, 79(2):444–462, 2011.

[14] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009.

[15] Frank R. Kschischang, Brendan J. Frey, and Hans-Andrea Loeliger. Factor graphs and the sum-product algorithm. *IEEE TRANSACTIONS ON INFORMATION THEORY*, 47:498–519, 1998.

[16] Simon C. Lovell, J. Michael Word, Jane S. Richardson, and David C. Richardson. The penultimate rotamer library. *Proteins: Structure, Function, and Bioinformatics*, 40(3):389–408, 2000.

[17] Adegoke A. Ojewole, Jonathan D. Jou, Vance G. Fowler, and Bruce R. Donald. $BBK*$ (branch and bound over K^*: A provable and efficient ensemble-based algorithm to optimize stability and binding affinity over large sequence spaces. In *Lecture Notes in Computer Science*, pages 157–172. Springer International Publishing, 2017.

[18] Stephanie M. Reeve, Pablo Gainza, Kathleen M. Frey, Ivelin Georgiev, Bruce R. Donald, and Amy C. Anderson. Protein design algorithms predict viable resistance to an experimental antifolate. *Proceedings of the National Academy of Sciences*, 112(3):749–754, 2015.

[19] K. E. Roberts and B. R. Donald. Improved energy bound accuracy enhances the efficiency of continuous protein design. *Proteins*, 83(6):1151–1164, Jun 2015.

[20] Kyle E. Roberts, Patrick R. Cushing, Prisca Boisguerin, Dean R. Madden, and Bruce R. Donald. Computational design of a pdz domain peptide inhibitor that rescues cftr activity. *PLoS Comput Biol*, 8(4):1–12, 04 2012.

[21] Kyle E. Roberts, Patrick R. Cushing, Prisca Boisguerin, Dean R. Madden, and Bruce R. Donald. Computational design of a pdz domain peptide inhibitor that rescues cftr activity. *PLOS Computational Biology*, 8(4):1–12, 04 2012.

[22] David Simoncini, David Allouche, Simon de Givry, Cline Delmas, Sophie Barbe, and Thomas Schiex. Guaranteed discrete energy optimization on large protein design problems. *Journal of Chemical Theory and Computation*, 11(12):5980–5989, 2015. PMID: 26610100.

[23] Bharath K. Sriperumbudur, Kenji Fukumizu, and Gert R. G. Lanckriet. Universality, characteristic kernels and rkhs embedding of measures. *J. Mach. Learn. Res.*, 12:2389–2410, July 2011.

[24] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. A new class of upper bounds on the log partition function. *IEEE Transactions on Information Theory*, 51(7):2313–2335, July 2005.

[25] Rui Wang and Haizhang Zhang. Optimal sampling points in reproducing kernel hilbert spaces. *CoRR*, abs/1207.5871, 2012.