Multi-Domain Sampling With Applications to Structural Inference of Bayesian Networks

Qing ZHOU

When a posterior distribution has multiple modes, unconditional expectations, such as the posterior mean, may not offer informative summaries of the distribution. Motivated by this problem, we propose to decompose the sample space of a multimodal distribution into domains of attraction of local modes. Domain-based representations are defined to summarize the probability masses of and conditional expectations on domains of attraction, which are much more informative than the mean and other unconditional expectations. A computational method, the multi-domain sampler, is developed to construct domain-based representations for an arbitrary multimodal distribution. The multi-domain sampler is applied to structural learning of protein-signaling networks from high-throughput single-cell data, where a signaling network is modeled as a causal Bayesian network. Not only does our method provide a detailed landscape of the posterior distribution but also improves the accuracy and the predictive power of estimated networks. This article has supplementary material online.

KEY WORDS: Domain-based representation; Monte Carlo; Multimodal distribution; Network structure; Protein-signaling network; Wang–Landau algorithm.

1. INTRODUCTION

In Bayesian inference the information on an unknown parameter $\boldsymbol{\theta}$ given an observed dataset $\mathbf{y}_{obs}$ is contained in the posterior distribution $p(\boldsymbol{\theta} \mid \mathbf{y}_{obs})$. When a posterior distribution does not belong to a well-characterized family of distributions, Markov chain Monte Carlo (MCMC) is a standard computational approach to Bayesian inference via sampling from the posterior distribution. Typical examples of MCMC include the Metropolis–Hastings (MH) algorithm (Metropolis et al. 1953; Hastings 1970) and the Gibbs sampler (Geman and Geman 1984; Tanner and Wong 1987; Gelfand and Smith 1990). Thorough reviews of recent developments on Monte Carlo methods and their applications in Bayesian computation can be found in the work of Chen, Shao, and Ibrahim (2001) and Liu (2008). The posterior mean $E(\boldsymbol{\theta} \mid \mathbf{y}_{obs})$ and other expectations are usually approximated from a Monte Carlo sample to summarize the posterior distribution. However, these unconditional expectations may not offer good summaries of the information for Bayesian inference when a posterior distribution has multiple local modes. One can easily construct a multimodal posterior distribution of which the mean is located in a low-density region and thus using it as an estimator for $\boldsymbol{\theta}$ lacks a conventional interpretation. To extract more information contained in a multimodal posterior distribution, it is desired to identify all major modes and calculate various statistics in appropriate neighborhoods of these modes.

To achieve these tasks, we propose to partition the sample space of $\boldsymbol{\theta}$ into a collection of domains such that the posterior distribution restricted to each domain is unimodal. The most parsimonious partition that minimizes the number of domains is to use the domains of attraction (to be defined rigorously later) of the local modes. Take the trimodal distribution $p(\boldsymbol{\theta})$ in Figure 1 as an illustration. The space is partitioned into three domains, denoted by Θ_1, Θ_2, and Θ_3: Each domain contains exactly one local mode; if we move any $\boldsymbol{\theta} \in \Theta_k$

$(k = 1, 2, 3)$ along the trajectory that always follows the gradient direction of $p(\boldsymbol{\theta})$, it will eventually reach the mode in the domain Θ_k. We may then calculate various conditional expectations on these domains, $E[h(\boldsymbol{\theta}) \mid \boldsymbol{\theta} \in \Theta_k]$ $(k = 1, 2, 3)$, for different functions h. Together with the probability masses of the domains, $P(\boldsymbol{\theta} \in \Theta_k)$, they provide more informative summaries of the distribution $p(\boldsymbol{\theta})$ than unconditional expectations. Such a summary is called a domain-based representation (DR) for $p(\boldsymbol{\theta})$.

Although desired, construction of DRs for an arbitrary distribution is very challenging in practice. Sufficient Monte Carlo samples from domains of all local modes are necessary for estimating DRs, but efficient sampling from a multimodal distribution has always been a hard problem. In this article, we develop a computational method that is able to construct domain-based representations for an arbitrary multimodal distribution. We partition the sample space into domains of attraction and utilize an iterative weighting scheme aiming at sampling from each domain with an equal frequency. The weighting scheme was proposed by Wang and Landau (2001), and further developed and generalized by Liang (2005), Liang, Liu, and Carroll (2007), and Atchadé and Liu (2010), among others. However, a direct application of these existing methods cannot provide accurate estimation of DRs, due to at least two reasons. First, sample space partition used in these methods is usually predetermined according to a set of selected density levels. But partitioning the space into domains of attraction, as employed in our method, cannot be completed beforehand because it is a nontrivial job to detect all local modes and their domains in real applications. Second, the above methods mostly rely on simple local moves and lack a coherent global move to jump between different local modes. To obtain accurate estimation of DRs, we propose a dynamic scheme to partition the sample space into domains of attraction and devise a global move that utilizes estimated DRs along sampling iterations to enable fast transitions between multiple domains. Since the main feature of
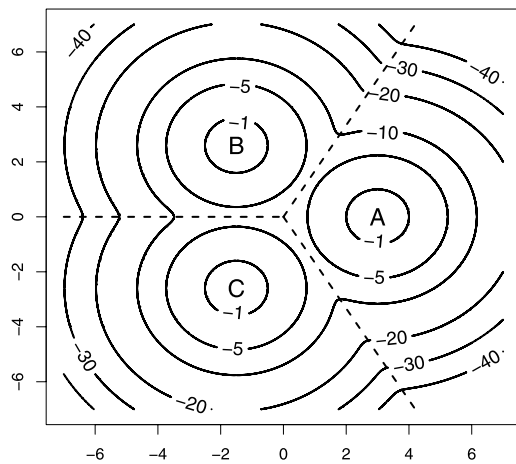
Figure 1. Contour plot of a two-dimensional density with three modes labeled A, B, and C. The numbers report log densities of contours and the dashed curves mark the boundaries between domains of attraction.

our method is to sample from multiple domains and construct DRs, we call it the multi-domain (MD) sampler.

The MD sampler can be applied to a wide range of Bayesian inference problems and it is particularly powerful in tackling problems with complicated posterior distributions. Although there are many such applications in different fields, this article mainly concerns structural learning of causal Bayesian networks from experimental data. Learning network structure via Monte Carlo sampling is very challenging as the multimodality of the posterior distribution is extremely severe (Friedman and Koller 2003; Ellis and Wong 2008; Liang and Zhang 2009). In this problem, a domain of attraction is defined by a set of network structures, each represented by a directed acyclic graph (DAG). In a sense, the goal of the MD sampler is to construct a detailed landscape of the posterior distribution, which can provide new insights into the structural learning problem. Application of our method to a scientific problem is illustrated by a study on constructing protein-signaling networks from single-cell experimental data. A living cell is highly responsive to its environment due to the existence of widespread and diverse signal transduction pathways. These pathways constitute a complex signaling network as cross-talks usually exist between them. Knowledge of the structure of this network is critical to the understanding of various cellular behaviors and human diseases. Recent advances in biotechnology allow the biologist to measure the states of a collection of molecules on a cell-by-cell basis. Such large-scale data contain rich information for statistical inference of signaling networks, but powerful computational methods are needed given the complexity in the likelihood function and the posterior distribution. With the MD sampler, not only can we build a signaling network from the posterior mean graph, but also we may discover new pathway connections revealed by different domains of the posterior distribution, which are not accessible by other approaches.

The remaining part of this article is organized as follows. Section 2 defines the domain of attraction and domain-based representation. In Section 3 we develop the MD sampler and its estimation of DRs, with convergence and ergodicity of the sampler established in the Appendix. The method is tested in

Section 4 on an example in Euclidean space and implemented in Section 5 for Bayesian inference of network structure with a simulation study. Section 6 is the main application to the construction of signaling networks in human T cells. The article concludes with a discussion on related and future works.

2. DOMAIN–BASED REPRESENTATION

Let $p(\mathbf{x})$, $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^m$, be the density of the target distribution. Suppose that $p(\mathbf{x})$ is differentiable and denote by $\nabla p(\mathbf{x})$ the gradient of p at $\mathbf{x}$. Define a differential equation

$$\frac{d\mathbf{x}(t)}{dt} = \nabla p(\mathbf{x}(t)) \tag{1}$$

and write a solution path of this equation as $\mathbf{x}(t)$, $t \in [0, \infty)$, where $\mathbf{x}(0)$ is a chosen initial point. Under some mild regularity conditions, $\mathbf{x}(\infty)$ converges to a local mode of $p(\mathbf{x})$, which is the basic intuition behind the gradient ascent algorithm to maximize $p(\mathbf{x})$. Denote by $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_K\}$ all the local modes, including the global mode, of $p(\mathbf{x})$. For $\mathbf{x} \in \mathcal{X}$, let $\mathbf{x}(0) = \mathbf{x}$ and define the domain partition index by

$$I(\mathbf{x}) = \begin{cases} k, & \text{if } \mathbf{x}(\infty) = \boldsymbol{v}_k, \text{ for } k = 1, \ldots, K \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

It maps $\mathbf{x}$ to the index of the local mode to which the solution path starting at $\mathbf{x}$ converges.

Definition 1. The domain of attraction of $\boldsymbol{v}_k$ is $D_k = \{\mathbf{x} \in \mathcal{X} : I(\mathbf{x}) = k\}$ for $k = 1, \ldots, K$. For simplicity we may call D_k an attraction domain or a domain of p.

If the stationary points of $p(\mathbf{x})$ have zero probability mass, then $\{D_k : k = 1, \ldots, K\}$ form a partition of the sample space $\mathcal{X}$ except for a set of zero probability mass. This is the default setting for this article.

Let $h(\mathbf{x})$ be a p-integrable function of $\mathbf{x}$. Write the probability mass of D_k and the conditional expectation of $h(\mathbf{X})$ given $\mathbf{X} \in D_k$ as

$$\lambda_k = P(\mathbf{X} \in D_k) = \int_{D_k} p(\mathbf{x}) \, d\mathbf{x}, \tag{3}$$

$$\mu_{h,k} = E[h(\mathbf{X}) \mid \mathbf{X} \in D_k] = \frac{1}{\lambda_k} \int_{D_k} h(\mathbf{x}) p(\mathbf{x}) \, d\mathbf{x}, \tag{4}$$

respectively, for $k = 1, \ldots, K$.

Definition 2. The domain-based representation of h with respect to the distribution p is a $2 \times K$ array, $DR_p(h) = \{(\mu_{h,k}, \lambda_k) : k = 1, \ldots, K\}$.

The DR is equivalent to the probability mass function of $E[h(\mathbf{X}) \mid I(\mathbf{X})]$ that assigns probability λ_k to $\mu_{h,k}$ for $k = 1, \ldots, K$. It provides the expectation $E[h(\mathbf{X})] = \sum_k \lambda_k \mu_{h,k}$ and the decomposed contributions from the attraction domains of $p(\mathbf{x})$. Such a representation gives an informative low-dimensional summary of a multimodal distribution. For a complex distribution with many local modes, however, we cannot afford to estimate $(\mu_{h,k}, \lambda_k)$ for every domain when K is too large and are less interested in domains of negligible probability masses (λ_k very close to zero). Due to these reasons we define domain-based representations with respect to a set of local modes $\{\boldsymbol{v}_k : k = 1, \ldots, M\}$. Index all the local modes as

$\mathbf{v}_1, \ldots, \mathbf{v}_M, \ldots, \mathbf{v}_K$ ($M \leq K$). Define the domain partition index with respect to $\{\mathbf{v}_k\}_{k=1}^{M}$ by $I_M(\mathbf{x}) = I(\mathbf{x})$ if $1 \leq I(\mathbf{x}) \leq M$ and $I_M(\mathbf{x}) = 0$ otherwise. Then the sample space $\mathcal{X}$ can be partitioned into $D_k = \{\mathbf{x} \in \mathcal{X} : I_M(\mathbf{x}) = k\}$ for $k = 0, \ldots, M$, where D_k is the domain of $\mathbf{v}_k$ ($k \geq 1$) and $D_0 = \mathcal{X} - \bigcup_{k=1}^{M} D_k$. The DR of h with respect to $\{\mathbf{v}_k\}_{k=1}^{M}$ is defined by the array $\{(\mu_{h,k}, \lambda_k) : k = 0, \ldots, M\}$, where λ_0 and $\mu_{h,0}$ are defined for D_0 similarly as in Equations (3) and (4), respectively. Note that one can still obtain $E[h(\mathbf{X})] = \sum_{k=0}^{M} \lambda_k \mu_{h,k}$ after merging $D_{M+1}, \ldots, D_K$ into D_0.

There is a geometric interpretation for the attraction domains of a posterior distribution. Suppose that $\mathbf{Y} = (\mathbf{Y}_1, \ldots, \mathbf{Y}_n)$ is a sample from an unknown distribution $\psi(\mathbf{y})$. We assume a parametric family $f_{\boldsymbol{\theta}} = f(\cdot \mid \boldsymbol{\theta})$, $\boldsymbol{\theta} \in \Theta$, as the model for $\mathbf{Y}$ and put a prior $\pi(\boldsymbol{\theta})$ on the unknown parameter $\boldsymbol{\theta}$. The posterior distribution of $\boldsymbol{\theta}$ is given by

$$p(\boldsymbol{\theta} \mid \mathbf{Y}) \propto \pi(\boldsymbol{\theta}) \prod_{i=1}^{n} f(\mathbf{Y}_i \mid \boldsymbol{\theta}) \approx e^{nE[\log f(\mathbf{Y}_1 \mid \boldsymbol{\theta})]}$$

when the sample size n is large, where $E[\log f(\mathbf{Y}_1 \mid \boldsymbol{\theta})] = \int \log[f(\mathbf{y} \mid \boldsymbol{\theta})] \psi(\mathbf{y}) \, d\mathbf{y}$. Denote the Kullback–Leibler (KL) divergence between ψ and $f_{\boldsymbol{\theta}}$ by

$$d_{KL}(\psi \| f_{\boldsymbol{\theta}}) = \int \log\left[\frac{\psi(\mathbf{y})}{f(\mathbf{y} \mid \boldsymbol{\theta})}\right] \psi(\mathbf{y}) \, d\mathbf{y}.$$

Then, $p(\boldsymbol{\theta} \mid \mathbf{Y}) \propto \exp[-n d_{KL}(\psi \| f_{\boldsymbol{\theta}})]$ when n is large. In the space of density functions, we can regard $d_{KL}(\psi \| f_{\boldsymbol{\theta}})$ as the "distance" to the point ψ from a point in the manifold $\mathcal{M} = \{f_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \Theta\}$. Note that ψ is not necessarily in $\mathcal{M}$ if our model assumption on $\mathbf{Y}$ is incorrect. Then $p(\boldsymbol{\theta} \mid \mathbf{Y})$ may be interpreted as a Boltzmann distribution on the manifold $\mathcal{M}$ under a potential field $n d_{KL}(\psi \| f_{\boldsymbol{\theta}})$. This potential pushes every $f_{\boldsymbol{\theta}}$, indexed by $\boldsymbol{\theta}$, toward ψ, and the collection of $\boldsymbol{\theta}$ which will be driven to an identical stationary point in $\mathcal{M}$ forms an attraction domain of $p(\boldsymbol{\theta} \mid \mathbf{Y})$.

3. THE MULTI–DOMAIN SAMPLER

To develop an algorithm that is able to construct domain-based representations with respect to the target distribution p, it is necessary to identify the attraction domain of any $\mathbf{x} \in \mathcal{X}$. When $p(\mathbf{x})$ is differentiable, this can be achieved by application of the gradient ascent (GA) algorithm that finds local modes of $p(\mathbf{x})$, or $\log p(\mathbf{x})$ for computational convenience. Generalization to the space of network structures will be discussed later.

A naive two-step approach to the construction of DRs is quite obvious. We may first apply a Monte Carlo algorithm to simulate a sample $\{\mathbf{X}^t\}_{t=1}^{n}$ from $p(\mathbf{x})$ or from a diffuse version of $p(\mathbf{x})$, for example, $[p(\mathbf{x})]^{1/\tau}$ for $\tau > 1$ as used in parallel tempering (Geyer 1991). Then, for every t we determine $I(\mathbf{X}^t)$ by a GA search initiated at $\mathbf{X}^t$ to find to which domain it belongs. This approach partitions the sample into attraction domains so that we can estimate the probability masses and conditional expectations for all identified domains. Although simple to implement, this two-step approach has a few drawbacks in terms of efficiency. Without a careful and specific design the Monte Carlo algorithm, even targeting at a diffuse version of $p(\mathbf{x})$, may not generate enough samples from all major domains or may completely miss some modes. As a result, estimation on some

attraction domains may be inaccurate or unavailable. In addition, this approach does not utilize the information on the target distribution provided by the constructed DRs. To overcome these drawbacks, we develop the MD sampler that may achieve simultaneously an efficient simulation from a multimodal distribution and an accurate construction of domain-based representations, with comparable computational complexity as the naive two-step approach.

3.1 The Main Algorithm

We wish to sample sufficiently from the majority of attraction domains. However, the density at the boundary between two neighboring domains is often exponentially low (e.g., Figure 1), which makes it difficult for an MH algorithm or a Gibbs sampler to jump across multiple domains. Thus, we need to allow the sampler to generate enough samples from such low-density regions that connect different domains. These considerations motivate the following double-partitioning design in the MD sampler.

Suppose that we are given a set of local modes of $p(\mathbf{x})$, $\{\mathbf{v}_1, \ldots, \mathbf{v}_M\}$, which may include the global mode. Given $\infty = H_0 > H_1 > \cdots > H_L = -\infty$, define the density partition index $J(\mathbf{x}) = j$ if $\log p(\mathbf{x}) \in [H_j, H_{j-1})$ for $j = 1, \ldots, L$. We partition the space $\mathcal{X}$ into $(M+1) \times L$ subregions,

$$D_{kj} = \{\mathbf{x} \in \mathcal{X} : I_M(\mathbf{x}) = k, J(\mathbf{x}) = j\},$$
$$k = 0, \ldots, M, j = 1, \ldots, L, \quad (5)$$

where $I_M(\mathbf{x})$ is the domain partition index with respect to $\{\mathbf{v}_k\}_{k=1}^{M}$. Then, the attraction domain of the local mode $\mathbf{v}_k$ is $D_k = \bigcup_j D_{kj}$ ($1 \leq k \leq M$). Note that some D_{kj} may be empty; if $\log p(\mathbf{v}_k) < H_i$, then all D_{kj} for $j \leq i$ are empty. In what follows, we only consider nonempty subregions. For a given matrix $\mathbf{W} = (w_{kj})_{(M+1) \times L}$, define a working density

$$p(\mathbf{x}; \mathbf{W}) \propto \sum_{k=0}^{M} \sum_{j=1}^{L} \frac{p(\mathbf{x}) \mathbf{1}(\mathbf{x} \in D_{kj})}{\exp(w_{kj})}, \quad (6)$$

where $\mathbf{1}(\cdot)$ is an indicator function. Let $\mathbf{W}^* = (w_{kj}^*)$ such that $\exp(w_{kj}^*) = \int_{D_{kj}} p(\mathbf{x}) \, d\mathbf{x}$. Then, the probability masses of D_{kj} are identical under $p(\mathbf{x}; \mathbf{W}^*)$. Sampling from $p(\mathbf{x}; \mathbf{W}^*)$ has two immediate implications. First, the sample sizes on the attraction domains, $\{D_k\}_{k=1}^{M}$, will be comparable, and thus, domain-based representations can be constructed with a high accuracy. Note that commonly used MCMC strategies for multimodal distributions, such as tempering, cannot generate samples of comparable sizes from different domains. Second, the sampler will stay in low-density regions (e.g., D_{kL}) for a substantial fraction of time, which makes it practically possible to jump between domains. Conversely, domain-based representations may be utilized to design efficient local and global moves for sampling from $p(\mathbf{x}; \mathbf{W}^*)$. We may construct online estimate of the covariance matrix on the domain of a local mode, which can be used for tuning the step size of a local move in this domain. For a multimodal distribution, tuning step size for each domain is more useful than tuning the overall step size (Haario, Saksman, and Tamminen 2001). Once we have identified sufficient local modes and estimated covariances of their respective domains, we can use them to design global moves that may jump from

one domain to another. As one can see, these proposals can be implemented only if we have partitioned samples into attraction domains.

For $\mathbf{x}, \mathbf{y} \in \mathcal{X}$, let $q(\mathbf{x}, \mathbf{y})$ be a proposal from $\mathbf{x}$ to $\mathbf{y}$ and $r(\mathbf{x}; \boldsymbol{\theta}, \mathbf{V})$ a distribution with parameters $\boldsymbol{\theta}$ and $\mathbf{V} \in \mathcal{V}$. We first develop the main algorithm of the MD sampler, assuming that the density ladder $\{H_j\}$ is fixed and M local modes $\{\boldsymbol{v}_k\}_{k=1}^{M}$ have been identified. Dynamic update of these parameters will be discussed in Section 3.2 as the burn-in algorithm. Let $\mathbf{g}_k(\mathbf{x})$ be a map from $\mathcal{X}$ to $\mathcal{V}$ for $k = 1, \ldots, M$.

Algorithm 1 (The main algorithm). Initialize $\mathbf{W}^1 = (w_{kj}^1)$, parameters $\mathbf{V}_k^1 \in \mathcal{V}$ ($k = 1, \ldots, M$), $p_{mx} \in [0, 1)$, and $\mathbf{X}^1 \in \mathcal{X}$. Set $\gamma_1 \leq 1$. For $t = 1, \ldots, n$:

1. Draw $\mathbf{Y}$ from $q(\mathbf{X}^t, \mathbf{y})$ with probability $(1 - p_{mx})$ or from the mixture distribution $\frac{1}{M} \sum_{k=1}^{M} r(\mathbf{y}; \boldsymbol{v}_k, \mathbf{V}_k^t)$ with probability p_{mx}.
2. Determine the density partition index $J(\mathbf{Y})$ and perform a GA search initiated at $\mathbf{Y}$ to determine the domain partition index $I_M(\mathbf{Y})$.
3. Accept or reject $\mathbf{Y}$ according to the MH ratio targeting at $p(\mathbf{x}; \mathbf{W}^t)$ to obtain $\mathbf{X}^{t+1}$.
4. For $k = 0, \ldots, M$ and $j = 1, \ldots, L$ update

$$w_{kj}^{t+1} = w_{kj}^t + \gamma_t \mathbf{1}\big(I_M(\mathbf{X}^{t+1}) = k, J(\mathbf{X}^{t+1}) = j\big), \quad (7)$$

for $k = 1, \ldots, M$ update

$$\mathbf{V}_k^{t+1} = \mathbf{V}_k^t + \frac{\gamma_t}{2}[\mathbf{g}_k(\mathbf{X}^{t+1}) - \mathbf{V}_k^t]\mathbf{1}(I_M(\mathbf{X}^{t+1}) = k), \quad (8)$$

and determine γ_{t+1}.

We may regard w_{kj}^t as a weight for the subregion D_{kj}. After each visit to D_{kj}, the weight w_{kj}^{t+1} increases by γ_t unit (7), which decreases the probability mass of D_{kj} under the working density $p(\mathbf{x}; \mathbf{W}^{t+1})$ (6). Such a weighting scheme aims to visit every D_{kj} uniformly. There are two typical choices of $\{\gamma_t\}$. The first choice follows the standard design in stochastic approximation which employs a predetermined sequence such that $\sum_{t=1}^{\infty} \gamma_t = \infty$ and $\sum_{t=1}^{\infty} \gamma_t^\zeta < \infty$ for $\zeta \in (1, 2)$ (Andrieu, Moulines, and Priouret 2005; Andrieu and Moulines 2006; Liang, Liu, and Carroll 2007). The second design, originally proposed by Wang and Landau (2001), adjusts γ_t in an adaptive way. However, there is difficulty in establishing its convergence (Atchadé and Liu 2010), and thus we adopt a modified Wang–Landau (MWL) design to update $\{\gamma_t\}$ in the MD sampler. Initialize $c_{kj}^1 = 0$ for all k and j in Algorithm 1. The MWL update at iteration t ($t = 1, \ldots, n$) is given below.

Routine 1 (MWL update). If $\gamma_t < \epsilon_\gamma$, set $\gamma_{t+1} = \gamma_t(\gamma_t + 1)^{-1}$; otherwise:

- Set $c_{kj}^{t+1} = c_{kj}^t + \mathbf{1}(\mathbf{X}^{t+1} \in D_{kj})$ for all k, j and calculate $\Delta c_{\max}^{t+1} = \max_{k,j} |c_{kj}^{t+1} - \bar{c}^{t+1}|$, where $\bar{c}^{t+1}$ is the average of all c_{kj}^{t+1}.
- If $\Delta c_{\max}^{t+1} \geq \eta \bar{c}^{t+1}$, set $\gamma_{t+1} = \gamma_t$; otherwise, set $\gamma_{t+1} = \rho \gamma_t$ and $c_{kj}^{t+1} = 0$ for all k, j.

That is, if $\gamma_t \geq \epsilon_\gamma$ we decrease γ_t ($\rho < 1$) only when the sampler has visited every subregion D_{kj} with a roughly equal frequency since our last modification of γ_t. Let $t_c = \min\{t : \gamma_t <$

$\epsilon_\gamma\}$. For $t > t_c$ the update becomes deterministic with $\gamma_t = 1/(t + \xi)$, where $\xi = \gamma_{t_c}^{-1} - t_c$. The default setting for all the results in this article is given by $\rho = 0.5$, $\eta = 0.25$, and $\epsilon_\gamma = 10^{-4}$.

Under some regularity conditions and the MWL update of $\{\gamma_t\}$,

$$\exp(w_{kj}^n) \xrightarrow{\text{a.s.}} \int_{D_{kj}} p(\mathbf{x}) \, d\mathbf{x} = \exp(w_{kj}^*), \quad (9)$$

after being normalized to sum up to one,

$$\mathbf{V}_k^n \xrightarrow{\text{a.s.}} \frac{\int_{D_k} \mathbf{g}_k(\mathbf{x}) p(\mathbf{x}; \mathbf{W}^*) \, d\mathbf{x}}{\int_{D_k} p(\mathbf{x}; \mathbf{W}^*) \, d\mathbf{x}} \triangleq \mathbf{V}_k^*, \quad (10)$$

$$\frac{1}{n} \sum_{t=1}^{n} h(\mathbf{X}^t) \xrightarrow{\text{a.s.}} \int_{\mathcal{X}} h(\mathbf{x}) p(\mathbf{x}; \mathbf{W}^*) \, d\mathbf{x}, \quad (11)$$

as $n \to \infty$. See Theorem A.1 in the Appendix for more details. If $\mathcal{X} \subseteq \mathbb{R}^m$, we often choose $\mathbf{g}_k(\mathbf{x}) = (\mathbf{x} - \boldsymbol{v}_k)(\mathbf{x} - \boldsymbol{v}_k)^{\mathsf{T}}$ so that $\mathbf{V}_k^*$ is close to the covariance matrix of the conditional distribution $[\mathbf{X} \mid \mathbf{X} \in D_k]$, where $\mathbf{X} \sim p(\mathbf{x}; \mathbf{W}^*)$. We use the mode $\boldsymbol{v}_k$ instead of the mean because the mode can be obtained accurately via a GA algorithm. The use of $\gamma_t/2 (< 1)$ in Equation (8) ensures that $\mathbf{V}_k^{t+1}$ is positive definite if $\mathbf{V}_k^t$ is positive definite.

There are two types of proposals in step 1 of the algorithm, a local proposal $q(\mathbf{X}^t, \mathbf{y})$ and a mixture distribution proposal. One advantage of partitioning samples into attraction domains is embodied in the mixture distribution proposal, in which we randomly draw a domain partition index $k \in \{1, \ldots, M\}$ and then propose a sample $\mathbf{Y}$ from $r(\mathbf{y}; \boldsymbol{v}_k, \mathbf{V}_k^t)$. Equal mixture proportions $(1/M)$ are used because a uniform sampling across domains is preferred. The default choice of the distribution $r(\mathbf{y}; \boldsymbol{v}_k, \mathbf{V}_k^t)$ in $\mathbb{R}^m$ is $\mathcal{N}(\boldsymbol{v}_k, \mathbf{V}_k^t)$ for $k = 1, \ldots, M$, which gives a mixture normal proposal that matches the mode and the covariance on each domain of the working target $p(\mathbf{x}; \mathbf{W}^t)$. This proposal uses a mixture distribution to approximate the multimodal target. It can generate efficient global jumps from one domain to another if $p(\mathbf{x}; \mathbf{W}^t)$ on the domain D_k can be well approximated by $r(\mathbf{x}; \boldsymbol{v}_k, \mathbf{V}_k^t)$ with the identified mode $\boldsymbol{v}_k$ and the estimated $\mathbf{V}_k^t$. For simplicity we call this proposal the *mixed jump*. The typical design for $q(\mathbf{X}^t, \mathbf{y})$ in $\mathbb{R}^m$ is to proposal $\mathbf{Y} \sim \mathcal{N}(\mathbf{X}^t, \sigma^2 \mathbf{I})$, where σ^2 is a scalar and $\mathbf{I}$ is the identity matrix. However, when the covariances are very different between domains, using a single local proposal may cause high autocorrelation, since the step size might be either too big for domains with small covariances or too small for those with large covariances, or both. In this case, we may incorporate an adaptive local proposal, $\mathbf{Y} \sim \mathcal{N}(\mathbf{X}^t, \sigma^2 \mathbf{V}_{I_M(\mathbf{X}^t)}^t)$, in addition to $q(\mathbf{X}^t, \mathbf{y})$, such that the learned covariance structure of a domain is utilized to guide the local proposal. This shows another advantage of the domain-partitioning design.

Remark 1. We summarize the unique features of the main algorithm. First, domain partitioning is incorporated in the framework of the Wang–Landau (WL) algorithm. This allows a more uniform sampling from different domains, which facilitates construction of DRs. At each iteration, a GA search is employed to determine $I_M(\mathbf{Y})$ and thus the computational complexity of this algorithm is comparable to the naive two-step approach. Second, an adaptive global move, the mixed jump, is proposed

given DRs constructed along the iteration, which utilizes identified modes and learned covariances to achieve between-domain moves.

Remark 2. Verification of the regularity conditions for convergence of the algorithm (see the Appendix) is recommended before application. Furthermore, we suggest a few convergence diagnostics that can be conveniently used in practice. First, γ_n should be small enough at the final iteration and the frequency of visiting different D_{kj} should be roughly identical. Second, $\mathbf{W}^n$ and $\mathbf{V}_k^n$ should have converged with an acceptable accuracy. Violations of these two criteria indicate that more iterations may be necessary. Third, the adaptive parameters used in the mixed jump ($\mathbf{V}_k^t$) should always stay in a reasonable range. For example, if $\mathbf{V}_k^t$ is a covariance matrix, one may check whether its eigenvalues are close to zero or unreasonably large, which may indicate divergence of the current run. If the last criterion is not satisfied, it is suggested to reinitialize the MD sampler with a smaller γ_1.

3.2 The Burn-in Algorithm

In practical applications of the MD sampler, the density ladder $\{H_j\}$ and the local modes $\{\mathbf{v}_k\}$ are updated dynamically in a burn-in period before the main algorithm (Algorithm 1). The dynamic updating schemes are crucial steps for constructing domain-based representations in real applications, as one cannot partition the sample space into domains of attraction beforehand. We set $H_1, \ldots, H_{L-1}$ as an evenly spaced sequence so that $\Delta H = H_j - H_{j+1}$ is a constant for $j = 1, \ldots, L-2$. Let $\{H_j^t\}$ be the density ladder and $\Lambda^t = \{\mathbf{v}_k^t : k = 1, \ldots, M^t\}$ be the set of M^t identified modes at iteration t. Let K^* be the maximum number of modes to be recorded and denote by $\mathbf{v}_{(\mathbf{x})}$ the mode of the domain that $\mathbf{x}$ belongs to. Let $\mathbf{0}$ be the zero matrix with dimension determined by the context. The following routine is used to update Λ^t when a new sample $\mathbf{Y}$ is proposed.

Routine 2. Let $s^t = \arg\min_{1 \le k \le M^t} p(\mathbf{v}_k^t)$.

- If $\mathbf{v}_{(\mathbf{Y})} \notin \Lambda^t$ and $M^t < K^*$, set $\Lambda^{t+1} = \Lambda^t \cup \{\mathbf{v}_{(\mathbf{Y})}\}$, $M^{t+1} = M^t + 1$, and initialize $w_{M^{t+1}j}^t = 0$ for all j;
- if $\mathbf{v}_{(\mathbf{Y})} \notin \Lambda^t$, $M^t = K^*$, and $p(\mathbf{v}_{(\mathbf{Y})}) > p(\mathbf{v}_{s^t}^t)$, set $\mathbf{v}_{s^t}^{t+1} = \mathbf{v}_{(\mathbf{Y})}$, $\mathbf{v}_k^{t+1} = \mathbf{v}_k^t$ for $k \ne s^t$, $M^{t+1} = M^t$, and assign $w_{0j}^t \Leftarrow w_{0j}^t + w_{s^t j}^t$ and $w_{s^t j}^t \Leftarrow 0$ for all j.
- otherwise set $\Lambda^{t+1} = \Lambda^t$ and $M^{t+1} = M^t$.

According to this routine, we record at most the K^* highest modes identified during the burn-in period. If there are more than K^* modes, Algorithm 1 will construct DRs with respect to the recorded modes. The weights (w_{kj}^t) are updated when a new mode replaces an old one in Λ^t, for which the assignment operator "$\Leftarrow$" is used to distinguish from equality.

The density ladder $\{H_j^t\}$ is adjusted such that H_1^t, the lower bound of the highest density partition interval, is close to $\log u^*$, where u^* is the density of the highest mode identified so far. If $\log u^* > H_1^t + \Delta H$, we move upward the density ladder by ΔH unit and update the weights (w_{kj}^t) accordingly, with details provided in Routine 3. This strategy helps the sampler to explore the high-density part, which is important for statistical estimation and finding the global mode.

Routine 3. Given Λ^{t+1}, find $u^{t+1} = \max\{p(\mathbf{v}_k^{t+1}) : k = 1, \ldots, M^{t+1}\}$.

- If $\log u^{t+1} > H_1^t + \Delta H$, set $H_j^{t+1} = H_j^t + \Delta H$ for $j = 1, \ldots, L-1$; for $k = 0, \ldots, M^{t+1}$, assign $w_{kL}^t \Leftarrow w_{k(L-1)}^t + w_{kL}^t$, $w_{kj}^t \Leftarrow w_{k(j-1)}^t$ for $j = L-1, \ldots, 2$, and $w_{k1}^t \Leftarrow 0$;
- otherwise set $\{H_j^{t+1}\} = \{H_j^t\}$.

Algorithm 2 (The burn-in algorithm). Input L, ΔH, and K^*. Set $\gamma_0 = 1$. Initialize $\mathbf{X}^1 \in \mathcal{X}$, $\Lambda^1 = \{\mathbf{v}_{(\mathbf{X}^1)}\}$, $M^1 = 1$, $\mathbf{W}^1 = (w_{kj}^1)_{2 \times L} = \mathbf{0}$, and $\mathbf{V}_1^1$. Set $H_1^1 = \log p(\mathbf{v}_{(\mathbf{X}^1)})$ and $H_j^1 = H_{j-1}^1 - \Delta H$ for $j = 2, \ldots, L-1$. For $t = 1, \ldots, B$:

1. Draw $\mathbf{Y} \sim q(\mathbf{X}^t, \mathbf{y})$ and find $\mathbf{v}_{(\mathbf{Y})}$ by a GA search.
2. Given $\mathbf{v}_{(\mathbf{Y})}$, update Λ^{t+1} and M^{t+1} by Routine 2; if $\mathbf{v}_\ell^{t+1} = \mathbf{v}_{(\mathbf{Y})}$ is a new mode in Λ^{t+1}, initialize $\mathbf{V}_\ell^t$. Given Λ^{t+1}, update $\{H_j^{t+1}\}$ by Routine 3.
3. Given Λ^{t+1} and $\{H_j^{t+1}\}$, accept or reject $\mathbf{Y}$ with the MH ratio targeting at $p(\mathbf{x}; \mathbf{W}^t)$ to obtain $\mathbf{X}^{t+1}$.
4. Execute step 4 of Algorithm 1 with $\gamma_t = \gamma_0$.

Remark 3. Note that $\gamma_t = 1$ for every iteration in the burn-in algorithm. This pushes the sampler to explore different regions in the sample space so that more local modes can be identified. In this case, the weight w_{kj}^t records the number of visits to D_{kj} before the tth iteration, which is the reason for our updating schemes on $\{w_{kj}^t\}$ in Routine 2 when a mode is updated in Λ^{t+1} and in Routine 3 when the density ladder changes.

Remark 4. The burn-in algorithm can be used as an optimization method that searches for up to K^* local modes of the highest densities. As demonstrated in the Bayesian network applications, this algorithm is very powerful in finding global modes.

The MD sampler requires only a few input parameters, L, ΔH, p_{mx}, and K^*. A practical rule is to choose L and ΔH such that the range of the density partition intervals, $L\Delta H$ in log scale, is wide enough to cover important regions. In this article, we set $L\Delta H$ around 20 for the low-dimensional test example in Section 4 and around 200 for learning Bayesian networks in Sections 5 and 6. By default the probability of proposing a mixed jump $p_{mx} = 0.1$. The effect of keeping only K^* modes will be studied later with the examples.

3.3 Statistical Estimation

The domain-based representation of h is constructed by estimating λ_k (3) and $\mu_{h,k}$ (4) for $k = 0, \ldots, M$ with post burn-in samples, denoted by $\{\mathbf{X}^{t+1}\}_{t=1}^n$. Let $k^t = I_M(\mathbf{X}^{t+1})$, $j^t = J(\mathbf{X}^{t+1})$, $a_t = \sum_{k,j} \exp(w_{kj}^t)$, and $\exp(\tilde{w}_{kj}^t) = \exp(w_{kj}^t)/a_t$ such that $\sum_{k,j} \exp(\tilde{w}_{kj}^t) = 1$. The key identity for our estimation is

$$\frac{\sum_{t=1}^\infty h(\mathbf{X}^{t+1}) \exp(\tilde{w}_{k^t j^t}^t)}{\sum_{t=1}^\infty \exp(\tilde{w}_{k^t j^t}^t)} \xrightarrow{\text{a.s.}} \int_{\mathcal{X}} h(\mathbf{x}) p(\mathbf{x}) \, d\mathbf{x}, \quad (12)$$

which follows from (11) as $\exp(\tilde{w}_{k^t j^t}^t) \xrightarrow{\text{a.s.}} \exp(w_{k^t j^t}^*) \propto p(\mathbf{X}^{t+1})/p(\mathbf{X}^{t+1}; \mathbf{W}^*)$ asymptotically (9). See the articles by Liang (2009) and Atchadé and Liu (2010) for similar results. However, $\tilde{\mathbf{W}}^t = (\tilde{w}_{kj}^t)$ may be far from $\mathbf{W}^*$ even for post burn-in iterations. Thus, it is desired to use a weighted version of (12)

so that $\mathbf{X}^{t+1}$ will carry a higher weight if $\tilde{\mathbf{W}}^t$ is closer to $\mathbf{W}^*$. Since decrease in γ_t indicates convergence of the MD sampler and $a_{t+1}/a_t \xrightarrow{\text{a.s.}} e^{c\gamma_t}$ for $c \in (0, 1)$ (supplementary document), a reasonable choice is to weight $\mathbf{X}^{t+1}$ by a_t so that unnormalized (w_{kj}^t) will be used in (12). Consequently, $DR_p(h)$ is constructed with

$$\hat{\lambda}_k = \frac{\sum_{t=1}^n \mathbf{1}(\mathbf{X}^{t+1} \in D_k)\exp(w_{k^t j^t}^t)}{\sum_{t=1}^n \exp(w_{k^t j^t}^t)},$$

$$\hat{\mu}_{h,k} = \frac{\sum_{t=1}^n h(\mathbf{X}^{t+1})\mathbf{1}(\mathbf{X}^{t+1} \in D_k)\exp(w_{k^t j^t}^t)}{\sum_{t=1}^n \mathbf{1}(\mathbf{X}^{t+1} \in D_k)\exp(w_{k^t j^t}^t)},$$

for $k = 0, \ldots, M$. Then, $\mu_h = E[h(\mathbf{X})]$ is estimated by $\hat{\mu}_h = \sum_k \hat{\lambda}_k \hat{\mu}_{h,k}$. Please see supplementary document for more discussion on this weighted estimation.

In the next three sections we demonstrate the effectiveness of the MD sampler in statistical estimation, especially estimation of DRs, compared to the naive two-step approach. For all examples, we employ the WL algorithm with the MWL update (Routine 1) as the Monte Carlo method in the two-step approach. To minimize hidden artifacts in a comparison due to coding differences, we implement the WL algorithm with the same burn-in and main algorithms of the MD sampler. In the main algorithm (Algorithm 1) we replace the updating scheme in Equation (7) with

$$w_{kj}^{t+1} = w_{kj}^t + \gamma_t \mathbf{1}(J(\mathbf{X}^{t+1}) = j) \qquad (13)$$

for $k = 0, \ldots, M$ and $j = 1, \ldots, L$ and modify the burn-in algorithm accordingly, so that $w_{0j}^t = \cdots = w_{Mj}^t \overset{\Delta}{=} w_j^t$ for every iteration. Consequently, the working density is effectively

$$p(\mathbf{x}; \mathbf{W}^t) \propto \sum_{j=1}^L \frac{p(\mathbf{x})\mathbf{1}(J(\mathbf{x}) = j)}{\exp(w_j^t)} \qquad (14)$$

as used in the WL algorithm, which is a diffuse version of $p(\mathbf{x})$ such that each density partition interval will be equally sampled after convergence. Note that the same GA search is applied at each iteration to partition samples into attraction domains for estimating DRs. Our comparison aims to highlight the effect of domain partitioning and the mixed jump in the MD sampler which are the key differences from the WL algorithm.

4. A TEST EXAMPLE

We test the MD sampler with an example in $\mathbb{R}^m$. For this example, domain-based representations can be obtained via one-dimensional numerical integration with a high accuracy, which provides the basis to evaluate our estimation. We choose $K^* =$ 100, which is greater than the total number of local modes, to construct complete DRs.

Let $\mathbf{x} = (x_1, \ldots, x_m)$. The Rastrigin function (Gordon and Whitley 1993) is defined as

$$R(\mathbf{x}) = \sum_{i=1}^m x_i^2 + A \left[m - \sum_{i=1}^m \cos(\pi x_i) \right], \qquad (15)$$

where A is a positive constant. We set $A = 2$ and $m = 4$ in (15) to obtain our target distribution $p(\mathbf{x}) \propto \exp[-R(\mathbf{x})]$, which has $3^4 = 81$ local modes formed by all the elements of the product set $\{-1.805, 0, 1.805\}^4$. These local modes have five distinct log density values, 0, -3.62, -7.24, -10.87, and -14.49, dependent on the combinations of their coordinates. They are grouped accordingly into five layers so that the number of zeros and the number of ± 1.805 in the coordinates of a local mode at the kth layer are $(5 - k)$ and $(k - 1)$, respectively, for $k = 1, \ldots, 5$. The attraction domains of local modes at the same layer have identical probability masses and identical conditional means up to a permutation and change of signs of the coordinates.

We applied the MD sampler 100 times independently, each run with $L = 10$ density partition intervals, $\Delta H = 2$, $B = 50K$ burn-in iterations, and a total of 5 million (M) iterations (including the burn-in iterations). The local proposal was simply $\mathcal{N}(\mathbf{X}^t, \mathbf{I})$. The average acceptance rate was 0.26 for the local move and was 0.56 for the mixed jump. Let $\mathbf{X} = (X_1, \ldots, X_4)$ and $S = \sum_i X_i$. We estimated $E(\mathbf{X})$, $E(e^{2S})$, $E(\prod_i X_i)$, $E(\sum_i X_i^5)$, and $E(\sum_i X_i^6)$, all via domain-based representations. Since the target density of this example is a product of one-dimensional marginal densities, the above expectations can be calculated accurately through one-dimensional numerical integration. We compared our estimates from MD sampling with the results from numerical integration by computing mean squared errors (MSEs). We report the average MSE of the estimated log probability masses ($\log \lambda_k$) and the average MSE of the estimated conditional means ($\boldsymbol{\mu}_{\mathbf{X},k}$) over all the local modes at the same layer ($k = 1, \ldots, 5$), and for other functions we only report the MSEs of the estimated expectations to save space (Table 1).

As a comparison, we also applied the WL algorithm (as in the naive two-step approach) to this problem with the same parameter setting. The ratio (RMSE) of the MSE of the WL algorithm over that of the MD sampler for each estimate is given in Table 1. The WL algorithm showed larger MSEs than the MD sampler for almost all the estimates, especially for those on domains at layers 3, 4, and 5. For example, the MD sampler was at

Table 1. MSE comparison on the Rastrigin function

| | MSE | RMSE | | | MSE | RMSE | | | MSE | RMSE | |
	MD	WL	MD$_0$		MD	WL	MD$_0$		MD	WL	MD$_0$
$\log \lambda_1$	1.1e−5	1.93	2.24	$\boldsymbol{\mu}_{\mathbf{X},1}$	2.3e−4	0.83	2.87	$\mathbf{X}$	1.7e−4	1.22	3.19
$\log \lambda_2$	3.6e−3	2.35	2.47	$\boldsymbol{\mu}_{\mathbf{X},2}$	2.5e−4	2.88	2.96	e^{2S}	0.59	3.25	3.09
$\log \lambda_3$	3.5e−3	4.64	3.22	$\boldsymbol{\mu}_{\mathbf{X},3}$	2.8e−4	6.16	3.34	$\prod_i X_i$	1.6e−9	2.84	2.70
$\log \lambda_4$	3.3e−3	8.63	3.92	$\boldsymbol{\mu}_{\mathbf{X},4}$	2.9e−4	12.0	4.09	$\sum_i X_i^5$	6.1e−3	2.71	2.06
$\log \lambda_5$	3.2e−3	16.8	4.66	$\boldsymbol{\mu}_{\mathbf{X},5}$	3.3e−4	21.1	5.06	$\sum_i X_i^6$	0.11	1.95	2.26

NOTE: RMSE is the ratio of the MSE of the alternative method (WL or MD$_0$) to the MSE of the MD method.

least 16 times more efficient than the WL algorithm for estimating $\log \lambda_5$ and $\mu_{\mathbf{X},5}$. The WL algorithm did not simulate sufficient samples from these domains, although it visited uniformly different density partition intervals. On the contrary, the double-partitioning design facilitated the MD sampler to explore every domain in a uniform manner, which led to a substantial improvement in estimation for these layers. This shows the critical role of domain partitioning in estimating DRs. To study the effect of the mixed jump, we re-applied the MD sampler with $p_{mx} = 0$, and calculated the ratio of the resulting MSE (MD$_0$ in Table 1) over that of the MD sampler with $p_{mx} = 0.1$, the default setting. One sees an increase of two folds or more in MSEs without the mixed jump. The convergence of the MD sampler without the mixed jump became slower, reflected by a five-fold increase in γ_n after the same number of iterations, averaging over 100 independent runs. These observations demonstrate that the mixed jump served as an efficient global move which accelerated convergence of the MD sampler and improved estimation accuracy.

5. LEARNING BAYESIAN NETWORKS

A Bayesian network (BN) factorizes the joint distribution of m variables $Z = \{Z_1, \ldots, Z_m\}$ into

$$P(Z) = \prod_{i=1}^{m} P(Z_i \mid \Pi_i^G), \qquad (16)$$

where $\Pi_i^G \subset Z$ is the parent set of Z_i. A graph G is constructed to code the structure of a BN by connecting each variable (node) to its child variables via directed edges. For (16) to be a well-defined joint distribution, the graph G must be a DAG. We consider the use of Bayesian networks in causal inference (Spirtes, Glymour, and Scheines 1993; Pearl 2000), which is tightly connected to many areas in statistics, such as structural equations, potential outcomes, and randomization (Rubin 1978; Robins 1986; Holland 1988; Neyman 1990). Here we follow Pearl's formulation of causal networks by modeling experimental intervention. If Z_j is a parent of Z_i in a causal Bayesian network, then experimental interventions that change the value of Z_j may affect the distribution of Z_i, but not conversely. Once all the parents of Z_i are fixed by intervention, the distribution of Z_i will not be affected by interventions on any variables in the set $Z \setminus (\Pi_i^G \cup \{Z_i\})$. In the example causal network of Figure 2, if we fix Z_1 and Z_3 by experimental intervention, then the distribution of Z_4 will not be affected by perturbations on Z_2, Z_5, or Z_6.
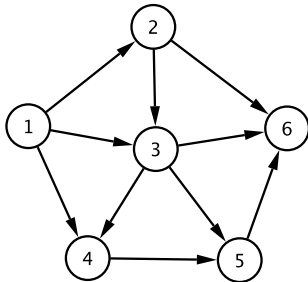

Figure 2. An example Bayesian network of six variables.

5.1 Posterior Distribution

We focus on the discrete case where each Z_i takes r_i states indexed by $1, \ldots, r_i$ and the parents of Z_i take $q_i = \prod_{Z_j \in \Pi_i^G} r_j$ joint states. Let θ_{ijk} be the causal probability for $Z_i = j$ given the kth joint state of its parent set. A causal BN with a given structure G is parameterized by $\boldsymbol{\Theta} = \{\theta_{ijk} : \sum_j \theta_{ijk} = 1, \theta_{ijk} \geq 0\}$.

We infer network structure from two types of data jointly, experimental data and observational data. For experimental data, a subset of variables are known to be fixed by intervention. Inferring causality with intervention has been extensively studied in various contexts (e.g., Robins 1986, 1987; Pearl 1993). We adopt the work of Cooper and Yoo (1999) for calculating the posterior probability of a network structure given a mix of experimental and observational data. Suppose that N_{ijk} is the number of data points for which Z_i is not fixed by intervention and is found in state j with its parent set in joint state k. Then, the collection of counts $\mathbf{N} = \{N_{ijk}\}$ is the sufficient statistic for $\boldsymbol{\Theta}$ (Ellis and Wong 2008). Let $|\Pi_i^G|$ be the size of the parent set of Z_i. The prior distribution over network structures is specified as $\pi(G) \propto \beta^{\sum_i |\Pi_i^G|}$, $\beta \in (0, 1)$, which penalizes graphs with a large number of edges. With a product-Dirichlet prior for $\boldsymbol{\Theta}$, the posterior distribution $[G \mid \mathbf{N}]$ (Cooper and Herskovits 1992) is

$$P(G \mid \mathbf{N}) \propto \prod_{i=1}^{m} \left\{ \beta^{|\Pi_i^G|} \prod_{k=1}^{q_i} \left[\frac{\Gamma(\alpha_{i \cdot k})}{\Gamma(\alpha_{i \cdot k} + N_{i \cdot k})} \right. \right.$$
$$\left. \left. \times \prod_{j=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})} \right] \right\}, \quad (17)$$

where $\alpha_{ijk} = \alpha/(r_i q_i)$ is the pseudo count for the causal probability θ_{ijk} in the product-Dirichlet prior and $N_{i \cdot k} = \sum_j N_{ijk}$ (similarly for $\alpha_{i \cdot k}$). The hyperparameters in the prior distributions are chosen as $\beta = 0.1$ and $\alpha = 1$.

5.2 MD Sampling Over DAGs

The space of DAGs is discrete in nature. We define domains of attraction for $P(G \mid \mathbf{N})$ (17) with a move set composed of addition, deletion, and reversal of an edge. Given a DAG G_a, we say that another DAG G_b is a neighbor of G_a if G_b can be obtained via a single move starting from G_a, that is, by adding, deleting, or reversing an edge of G_a. Denote by $ngb(G_a)$ all the neighbors of G_a and let $\overline{ngb}(G_a) = ngb(G_a) \cup \{G_a\}$. A DAG G^* is defined as a local mode of a probability density (mass) function $p(G)$ if $p(G^*) > p(G')$ for every $G' \in ngb(G^*)$. Let G^0 be a DAG and define recursively

$$G^{t+1} = \underset{G \in \overline{ngb}(G^t)}{\arg \max} \, p(G), \quad \text{for } t = 0, 1, \ldots. \qquad (18)$$

That is, we recursively find the single move that leads to the greatest increase in p until a local mode is reached, which can be viewed as a discrete counterpart of the gradient ascent algorithm. If there is more than one maximum in (18) with an identical function value, we take the first maximum according to a fixed ordering of the neighbors. We call this recursion the steepest neighbor ascent (SNA). Based on SNA, we define the domain partition index $I(G)$ and the attraction domains of p in

the same manner as for a differentiable density [Equation (2) and Definition 1].

The target distribution in this application is the posterior distribution $P(G \mid \mathbf{N})$ and we define working density $P(G \mid \mathbf{N}; \mathbf{W})$ similarly as in (6). To implement the MD sampler for DAGs, we employ the move set as the local proposal and develop the following mixed jump. For a DAG G, we define an edge variable E_{ij}^G for every pair of nodes Z_i and Z_j $(i < j)$ such that $E_{ij}^G = 1$ if Z_i is a parent of Z_j, $E_{ij}^G = -1$ if Z_j is a parent of Z_i, and $E_{ij}^G = 0$ otherwise. Given a DAG ν, let $\mathbf{C}(G; \nu) = (C_a(G; \nu), C_d(G; \nu), C_r(G; \nu))$ be a map of G, where

$$C_a(G; \nu) = \sum_{i<j} \mathbf{1}(E_{ij}^G \neq 0, E_{ij}^\nu = 0),$$

$$C_d(G; \nu) = \sum_{i<j} \mathbf{1}(E_{ij}^G = 0, E_{ij}^\nu \neq 0),$$

$$C_r(G; \nu) = \sum_{i<j} \mathbf{1}(E_{ij}^G \cdot E_{ij}^\nu = -1).$$

In words, $\mathbf{C}(G; \nu)$ gives the numbers of additions, deletions, and reversals needed to obtain G from ν. Let $T = m(m-1)/2$ be the total number of node pairs and $|E_\nu|$ be the number of edges in ν. Then, the number of common edges between G and ν is $|E_\nu| - [C_r(G; \nu) + C_d(G; \nu)]$ and the number of node pairs with no edge in either DAG is $T - |E_\nu| - C_a(G; \nu)$. Given a set of local modes of $P(G \mid \mathbf{N})$, $\{\nu_k\}_{k=1}^M$, let $\mathbf{g}_k(G) = \mathbf{C}(G; \nu_k)$ in the update of $\mathbf{V}_k^t$ (8) of Algorithm 1, where $\mathbf{V}_k^t = (v_{k,a}^t, v_{k,d}^t, v_{k,r}^t)$ is a vector. As $n \to \infty$, $\mathbf{V}_k^n \xrightarrow{\text{a.s.}} E[\mathbf{C}(G; \nu_k) \mid G \in D_k]$, where the expectation is taken with respect to the limiting working density $P(G \mid \mathbf{N}; \mathbf{W}^*)$ (10). In the mixed jump, after a local mode ν_k is randomly chosen, we sequentially modify the edge variables of ν_k to propose a new DAG Y. The proposal is designed according to $\mathbf{V}_k^t$, the current estimate of the expected numbers of additions, deletions, and reversals of DAGs in the domain D_k relative to the mode ν_k. Let $|E_k|$ be the number of edges in ν_k. If $E_{ij}^{\nu_k} \neq 0$, we propose to reverse, delete, or retain the edge $E_{ij}^{\nu_k}$ (i.e., $E_{ij}^Y = -E_{ij}^{\nu_k}, 0$, or $E_{ij}^{\nu_k}$) with probabilities proportional to the vector $(v_{k,r}^t, v_{k,d}^t, |E_k| - (v_{k,r}^t + v_{k,d}^t)) + b$, where $b > 0$ is a small prior count added to each category. Analogously, if $E_{ij}^{\nu_k} = 0$ we propose $E_{ij}^Y = 0, 1$, or -1 with probabilities proportional to $(T - |E_k| - v_{k,a}^t, v_{k,a}^t/2, v_{k,a}^t/2) + b$. Last, to ensure a proposed graph is acyclic, a check for cycles is performed when we propose to add or reverse an edge in either the local proposal or the mixed jump. If the resulting graph is cyclic, we suppress the probability for the corresponding move.

Following the common practice in structural learning of discrete BNs, we set an upper bound for the number of parents (indegree) of a node. In all the following examples and applications, this upper bound is chosen to be four. We are interested in the posterior expected adjacency matrix $\mathbf{A} = (a_{ij})_{m \times m}$ and its domain-based representation, where a_{ij} $(1 \leq i, j \leq m)$ is the posterior probability for a directed edge from Z_i to Z_j. For each identified local mode ν_k, we estimate the probability mass λ_k of its attraction domain D_k and the conditional expected adjacency matrix $\mathbf{A}_k$ on the domain. Then, $\mathbf{A}$ is estimated by $\hat{\mathbf{A}} = \sum_k \hat{\lambda}_k \hat{\mathbf{A}}_k$.

5.3 Simulation

We simulated data from two BNs, each of six binary variables ($m = 6$, $r_i = 2$, $\forall i$). This is the maximum number of nodes for which we can enumerate all DAGs, numbering about four million, to obtain true posterior distributions and domain-based representations as the ground truth for testing a computational method. The first network has a chain structure in which Z_i is the only parent of Z_{i+1} for $i = 1, \ldots, 5$ and Z_1 has no parent. The second network has a more complex structure shown in Figure 2. We simulated 50 datasets independently from each network. In each dataset, 20% of the data points were generated with interventions. Please see supplementary document for data simulation details.

The MD sampler was applied to the 100 datasets with $L = 15$, $\Delta H = 10$, $p_{mx} = 0.1$, $K^* = 100$, and a total of 5M iterations with the first 50K as burn-in iterations. To verify its performance, we compared identified local modes, estimated probability masses $\{\log \hat{\lambda}_k\}$, conditional expected adjacency matrices $\{\hat{\mathbf{A}}_k\}$, and expected adjacency matrix $\hat{\mathbf{A}}$ to their respective true values obtained via enumerating all DAGs. Our enumeration confirms that the posterior distributions indeed have multiple local modes. The chain and the graph (Figure 2) networks have on average 3.57 and 7.06 modes over the simulated datasets, respectively, and the maximum number of modes is 29 for the chain and 34 for the graph. As reported in Table 2, the MD sampler did not miss a single local mode for any dataset, which demonstrates its global search ability. Recall that all recorded modes are detected in the burn-in algorithm. In fact, all modes, including the global mode, were identified within 10K iterations for every dataset. This observation confirms the notion that the burn-in algorithm alone may serve as a powerful optimization algorithm (Remark 4). We calculated the MSE of the vector $(\log \hat{\lambda}_1, \ldots, \log \hat{\lambda}_K)$, where K is the number of local modes, and the average MSE of $\hat{\mathbf{A}}_1, \ldots, \hat{\mathbf{A}}_K$. When calculating the MSE of the log probability vector, we ignored those tiny domains with a probability mass $< 10^{-4}$. These estimates are seen to be very accurate as reported in Table 2.

We also applied the MD sampler with $p_{mx} = 0$ (MD$_0$) and the WL algorithm with the same parameter setting to these datasets (Table 2). The degraded performance of MD$_0$ demonstrates the effectiveness of the mixed jump for sampling DAGs.

Table 2. Comparison on simulated data from two BNs

	MD	MD$_0$	WL	$K^* = 10$
	MSE		RMSE	
# of missed modes	0	0	0	0.51
$\log \hat{\lambda}_k$	0.028	1.48	5.06	0.48
$\hat{\mathbf{A}}_k$	1.3e−4	1.65	11.0	0.99
$\hat{\mathbf{A}}$	1.3e−4	1.76	1.67	1.03
# of missed modes	0	0	0.12	2.06
$\log \hat{\lambda}_k$	0.029	401	1368	0.84
$\hat{\mathbf{A}}_k$	1.7e−4	2.96	13.6	1.01
$\hat{\mathbf{A}}$	1.5e−4	7.17	5.09	0.98

NOTE: The top and bottom panels report the results for the chain and the graph networks, respectively. For each estimate, reported are the MSE of the MD sampler and the RMSEs (ratios) of the other methods relative to the MD sampler.

The WL algorithm missed 0.12 modes on average for the second network, and its estimation of the DR $\{(\hat{\mathbf{A}}_k, \hat{\lambda}_k)\}$ was much less accurate compared to that of the MD sampler. The average MSE of $\hat{\mathbf{A}}_1, \ldots, \hat{\mathbf{A}}_K$ and the MSE of $(\log \hat{\lambda}_k)_{1:K}$ were more than 10 and 1000 times greater than those of the MD sampler, respectively. The huge MSE of the $(\log \hat{\lambda}_k)$ constructed by the WL algorithm was often due to severe underestimation of the probability masses of domains sampled insufficiently. This result implies that without domain partitioning, the WL algorithm is unable to estimate domain-based representations for a BN of a moderately complicated structure. Since the number of local modes often increases very fast with the complexity of a problem, we re-applied the MD sampler with $K^* = 10$ to investigate the effect of keeping only a subset of local modes. Obviously, the algorithm missed a few local modes when the total number of modes exceeded K^*. But in terms of estimating $\mathbf{A}$ and $\mathbf{A}_k$, the performance of the MD sampler with $K^* = 10$ was very comparable to its performance when all the local modes were kept (Table 2). The probability mass outside the domains of recorded modes, $\lambda_0 = 1 - \sum_{k=1}^{K^*} \lambda_k$, is less than 0.007 averaging over the 15 datasets where the posterior distributions have more than 10 local modes. This confirms that the MD sampler indeed captured major modes in the burn-in period.

6. PROTEIN–SIGNALING NETWORKS

6.1 Background and Data

The ability of cells to properly respond to environment is the basis of development, tissue repair, and immunity. Such response is established via information flow along signaling pathways mediated by a series of signaling proteins. Cross-talks and interplay between pathways reflect the network nature of the interaction among these signaling molecules. Construction of signaling networks is an important step toward a global understanding of normal cellular responses to various environmental stimuli and more effective treatment of complex diseases caused by malfunction of components in a pathway. Causal Bayesian networks may be used for modeling signaling networks as the relation among pathway components has a natural causal interpretation. That is, the activation or inhibition of a set of upstream molecules in a network causes the state change of downstream molecules. An edge from molecule A to molecule B in a signaling network implies that a change in the state of A causes a change in the state of B via a direct biochemical reaction. Here, a state change refers to a chemical, physical, or locational modification of a molecule. However, as there may exist mutual regulation between two signaling molecules, the use of DAGs for modeling signaling networks is only a first-step approximation.

In this study, we construct protein-signaling networks from flow cytometry data. Polychromatic flow cytometry is a high-throughput technique for probing simultaneously the (phosphorylation) states of multiple proteins in a single cell. Since measurements are collected on a cell-by-cell basis, huge amounts of data can be produced in one experiment. Sachs et al. (2005) made flow cytometry measurements of 11 proteins and phospholipids in the signaling network of human primary naive $CD4^+$ T cells under nine different experimental perturbations that either activate or inhibit a particular molecule or activate
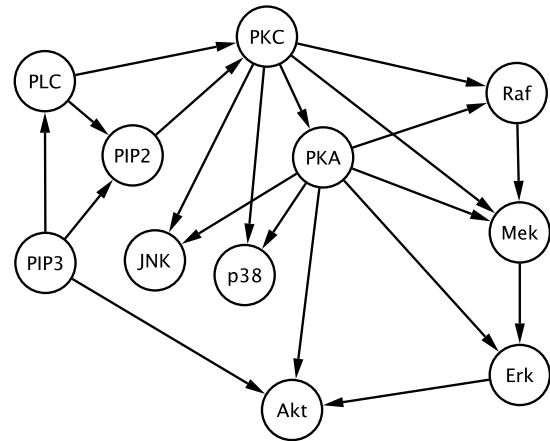

Figure 3. An annotated protein-signaling network in naive $CD4^+$ T cells.

the entire pathway. Note that a perturbation that activates or inhibits a particular molecule is essentially an intervention on the molecule, so that causal structures of the underlying network may be inferred. Under each perturbation, 600 cells were collected with 11 measurements for each. The measurements in the data were discretized into three levels, high, medium, and low by Sachs et al. In summary, this dataset contains 5400 data points for 11 ternary variables. Since naive T cells are essential for the immune system to continuously respond to unfamiliar pathogens, extensive studies have been conducted to establish the signaling pathways. An annotated signaling network among the 11 molecules, provided by Sachs et al., is depicted as a causal Bayesian network in Figure 3. This network contains 18 edges that are well-established in the literature and two edges (PKC → PKA and Erk → Akt) reported from recent experiments independent of the flow cytometry data.

6.2 Predicted Networks

The MD sampler was applied to this dataset with $L = 20$, $\Delta H = 10$, $p_{mx} = 0.1$, and $K^* = 10$. The total number of iterations was 5M, of which the first 50K were used for burn-in. We estimated the posterior expected adjacency matrix $\mathbf{A}$ and its domain-based representation. Three predicted networks were constructed by thresholding posterior edge probabilities at

Table 3. Results on the flow cytometry data

	MD	MD$_0$	WL
Global max (SD)	−31,757.9 (2.7)	−31,787.3 (82)	−31,937.8 (199)
TP/FP ($c = 0.5$)	15.50/10.35	15.55/10.35	13.40/12.35
TP/FP ($c = 0.7$)	15.50/10.35	15.55/10.35	13.35/12.25
TP/FP ($c = 0.9$)	15.50/10.35	15.55/10.35	13.35/11.95
TP/FP ($c = 0.5$)	15.2/10.3	14.6/10.6	11.6/13.6
TP/FP ($c = 0.7$)	15.2/10.2	14.6/10.6	11.6/13.2
TP/FP ($c = 0.9$)	15.2/9.9	14.6/10.2	11.6/13.1
Log pred (mean)	0	−1.4	−62.1
Log pred (DR)	17.5	16.2	−51.7

NOTE: The top and bottom panels report the average results over 20 independent runs on the full dataset and the average results over ten test datasets in cross-validation, respectively. Predictive probabilities (Log pred) are reported as log ratios over the predictive probability given the mean network of the MD sampler.

$c = 0.5, 0.7, 0.9$, that is, an edge from Z_i to Z_j was predicted if the edge probability $\hat{a}_{ij} \geq c$. For simplicity we call such a predicted network a mean network (with a threshold c). Table 3 (top panel) reports the number of true positive edges (TP) that are both predicted by the MD sampler and annotated in Figure 3 and the number of false positive edges (FP) that are predicted but not annotated, together with the (unnormalized) log posterior probability of the identified global maximum DAG. To compare the results, we re-applied the MD sampler with the same parameters except that $p_{mx} = 0$ (MD$_0$ in Table 3) and applied the WL algorithm with the same parameters as used in the MD sampler to the same data. The average result over 20 independent runs of each method is summarized in Table 3. In terms of finding the global mode, the MD sampler was much more effective and robust than the other two algorithms, reflected by a much higher average log probability and a much smaller standard deviation across multiple runs. The MD sampler with or without the mixed jump showed comparable results in predicting network structures, and both predicted more true positives and fewer false positives than the WL algorithm did for all the three thresholds. We noticed that the mean networks constructed with different thresholds ($c = 0.5, 0.7, 0.9$) were almost identical. This was due to the fact that the posterior edge probabilities were close to either 1 or 0 because of the large data size. The network constructed from the same data by the order-graph sampler, reported in figure 11 of the article by Ellis and Wong (2008), has 9 true positive and 11 false positive edges, which misses many more true edges and includes more false positives than the networks predicted by the MD sampler. These results demonstrate that the MD sampler is very powerful in learning underlying network structures from experimental data compared to other advanced Monte Carlo techniques.

Next, we focus on the estimation of the DR, $\{(\hat{\mathbf{A}}_k, \hat{\lambda}_k) : k = 0, \ldots, K^*\}$, and its scientific implications. A network $\hat{G}_k$ can be constructed for an attraction domain by thresholding $\hat{\mathbf{A}}_k$, the conditional expected adjacency matrix on the domain D_k, for $k = 1, \ldots, K^*$. To distinguish it from the mean network, we call $\hat{G}_k$ a local network. We take the result of a representative run of the MD sampler ($p_{mx} = 0.1$) to demonstrate local networks with the threshold $c = 0.9$. The parent sets of eight nodes are identical across the $K^* = 10$ local networks. We report in Table 4 the parents of the other three nodes, PLC, PIP3, and Erk, which

are distinct among the local networks, together with the probability masses ($\log \hat{\lambda}_k$) of the 10 domains and the probabilities of the local modes [$\log P(\nu_k \mid \mathbf{N})$]. The local networks may predict meaningful alternative edges not included in the mean network, as illustrated by the result on a particular pathway, Raf $\rightarrow$ Mek $\rightarrow$ Erk (Figure 3). This expected pathway was predicted by all the local networks and the mean network. However, some local networks also contained a direct link from Raf to Erk (Table 4). As Mek was inhibited in one of the experimental conditions, this finding suggests that the cells may have another pathway that passes the signal from Raf to Erk via some indirect regulation or via molecules not included in this analysis, when Mek is not functioning properly. Such compensational mechanisms exist widely in many biological networks. Indeed, Raf has been reported to enhance the kinase activity of PKCθ, an isoform of PKC, although PKCθ is unlikely a direct phosphorylation target of Raf (Hindley and Kolch 2007). As indicated by Figure 3, Erk is a downstream node of PKC and thus may be regulated indirectly by Raf via the enhanced kinase activity of PKCθ. Such novel hypotheses could not be proposed if we did not construct the DR for the posterior distribution. Clearly, the DR of network structures not only gives a detailed landscape of various local domains but also provides new insights into the underlying scientific problem.

From Table 4 we find that the probability mass is dominated by the domain of the identified global mode with a log probability of $-31,757$. Consistent with the summary in Table 3, the MD sampler almost always reached this global mode for different runs. On the contrary, the highest mode detected by the WL algorithm with an average log probability of $-31,938$ is even much lower than the lowest mode in Table 4. In other words, the WL algorithm was inevitably trapped to some local modes with negligible probability masses. This again demonstrates the advantage of the MD sampler, particularly the burn-in algorithm, in finding global modes. Even when the probability mass of the global mode is dominant and other domains occupy only a small fraction of the sample space, without the domain-partitioning design the WL algorithm may be trapped to a local mode of a tiny probability mass and produce severely biased estimates.

6.3 Cross-Validation

To check the statistical variability and the predictive power of our method, we conducted ten-fold cross-validation on this

Table 4. Local networks constructed from domain-based representation

		Parents of		
$\log \hat{\lambda}_k$	$\log P(\nu_k \mid \mathbf{N})$	PLC	PIP3	Erk
-0.00274	$-31,756.81$	PKC p38 JNK	PLC PKC	Mek PKA PKC
-6.39777	$-31,762.75$	PKC p38 JNK	PLC PKC	Mek PKA Raf
-7.12166	$-31,764.16$	Mek PKA	PLC PKC	Mek PKA PKC
-8.34521	$-31,764.82$	PKC p38 JNK	PLC JNK	Mek PKA PKC
-10.6461	$-31,766.81$	Mek Akt p38	PLC PKC	Mek PKA PKC
-13.7193	$-31,770.11$	Mek Akt PKA	PLC PKC	Mek PKA Raf
-14.0777	$-31,770.77$	PKC p38 JNK	PLC JNK	Mek PKA Raf
-15.0459	$-31,772.18$	Mek Akt PKA	PLC JNK	Mek PKA PKC
-16.8860	$-31,772.76$	Mek Akt p38	PLC PKC	Mek PKA Raf
-18.5716	$-31,774.82$	Mek Akt p38	PLC JNK	Mek PKA PKC

dataset. We partitioned randomly the 5400 data points into ten subsets of equal sizes. We used nine subsets as training data to learn a mean network and calculated the predictive probability of data points in the other subset (test data) given the learned mean network. This procedure was repeated 10 times to test on every subset. We verified the average accuracy of the mean networks constructed from the 10 training datasets with different thresholds. The performance of the MD sampler on the training datasets was comparable to its performance on the full dataset, which implies its robustness to random sampling of input data. The improvement in accuracy (TP/FP) of the MD sampler over the other two methods became more significant, especially compared to the WL algorithm (Table 3, bottom panel). The average predictive probability of the test datasets given the mean networks constructed from the training datasets by each method with $c = 0.9$ is reported in Table 3 [Log pred (mean)], from which we see that the predictive power of the networks constructed by the two MD samplers was much higher than that of the WL algorithm (> 60 in log probability ratio). In addition, we utilized the estimated domain-based representations to calculate the predictive probability of a test data point $\mathbf{y}$ by

$$P(\mathbf{y} \mid \{\hat{G}_k, \hat{\lambda}_k\}) \triangleq \sum_{k=0}^{K^*} \hat{\lambda}_k P(\mathbf{y} \mid \hat{G}_k), \qquad (19)$$

where $P(\mathbf{y} \mid \hat{G}_k)$ is the marginal likelihood of $\mathbf{y}$ given the local network $\hat{G}_k$. This can be regarded as a domain-based approximation to the posterior predictive distribution

$$P(\mathbf{y} \mid \mathbf{y}_{obs}) = \sum_G P(G \mid \mathbf{y}_{obs}) P(\mathbf{y} \mid G),$$

that is, we use estimated probability masses $\hat{\lambda}_k$ and conditional mean networks $\hat{G}_k$ to approximate the posterior predictive probability. The advantage is that there is no need to store a large posterior sample of networks but only an estimated DR. Since Equation (19) captures the variability among different domains, it is expected to outperform the mean network in prediction. In fact, for each method the predictive probability calculated by (19) [Table 3, Log pred (DR)] was indeed significantly greater than the predictive probability calculated given its mean network, especially for the two MD samplers.

In real applications, we are interested in predicting results for a new experimental condition given observed data from other conditions. Thus, we also performed a nine-fold cross-validation where a training dataset was composed of cells from eight experimental conditions and a test dataset only included cells in the other one condition. We applied the MD sampler to construct mean networks and DRs $\{\hat{G}_k, \hat{\lambda}_k\}_{k=1}^{10}$ from the training datasets. The mean networks with $c = 0.9$ included, on average, 13.2 true edges with 10.8 false edges, which was slightly worse than the result from the ten-fold cross-validation. The degraded performance is expected as removing all cells from one experimental perturbation will increase the uncertainty in determining the directionality of the network. The domain-based prediction (19) was compared against the annotated network G^* given in Figure 3, which presumably has the highest predictive power, by evaluating the log-likelihood ratio (LLR) $\log R = \log[P(\mathbf{y} \mid \{\hat{G}_k, \hat{\lambda}_k\})/P(\mathbf{y} \mid G^*)]$, where $\mathbf{y}$ is a test data point. The average

$\log R$ over all test data points was -0.062, and thus the predictive probability for a new observation given the constructed DR is expected to be higher than $94\% (= e^{-0.062})$ of its likelihood given the annotated graph. This demonstrates the high predictive power of the domain-based prediction constructed by our method. As expected, the average LLR of the mean networks over G^* was 26% lower than the average of $\log R$.

7. DISCUSSION

The central idea of this article is to construct domain-based representations with the MD sampler. Related works have been seen in the physics literature under the name of superposition approximation. Please see the article by Wales and Bogdan (2006) for a recent review. Given a Boltzmann distribution $p_B(\mathbf{x}; \tau) \propto \exp[-H(\mathbf{x})/\tau]$, a superposition approach identifies the local minima of $H(\mathbf{x})$, that is, the local modes of $p_B(\mathbf{x}; \tau)$, and approximates $H(\mathbf{x})$ on the attraction domain of a local minimum by quadratic or high-order functions. The approximation is often proposed based on expert knowledge about the physical model under study. Expectations with respect to $p_B(\mathbf{x}; \tau)$ are then estimated by summing over approximations from identified domains. The accuracy of this approach largely depends on the employed approximation to $H(\mathbf{x})$ on a domain and thus may not work well for an arbitrary distribution. The MD sampler differs in that domain-based representations are constructed by Monte Carlo sampling which is able to provide accurate estimation with large-size samples; no expert knowledge about the target distribution is needed. In addition, our method also contains a coherent component for finding local modes, while the superposition approximation works more like a two-step approach.

From a computational perspective, the MD sampler integrates Monte Carlo and deterministic optimization. A few other methods also have the two ingredients, such as Monte Carlo optimization (Li and Scheraga 1987), the basin hopping algorithm (Wales and Doye 1997), and conjugate gradient Monte Carlo (CGMC) (Liu, Liang, and Wong 2000). In Monte Carlo optimization and the basin hopping algorithm, the target distribution $p(\mathbf{x})$ is modified to $\tilde{p}(\mathbf{x}) = p(\mathbf{v}_k)$ for all $\mathbf{x} \in D_k$, where D_k is the attraction domain of the mode $\mathbf{v}_k$. Then a Metropolis-type MCMC is used to sample from $\tilde{p}$, in which a local optimization algorithm is employed at each iteration to find $\tilde{p}(\mathbf{X}^t)$ for the current state $\mathbf{X}^t$. These methods have been applied to identification of minimum-energy structures of proteins and other molecules. However, its application to other fields is limited as the modified density $\tilde{p}(\mathbf{x})$ may be improper when the sample space is unbounded. In CGMC, a population of samples is evolved and a line sampling step (Gilks, Roberts, and George 1994) is performed on a sample along a direction pointing to a local mode found by local optimization initiated at another sample. In this way promising proposal may be constructed by borrowing local mode information from other samples. A possible future work on the MD sampler is to utilize a population of samples. Because local modes are recorded, similar proposals as the line sampling can be developed for the MD sampler to further enhance sampling effectiveness. Another future direction is to construct disconnectivity graphs (Becker and Karplus 1997) or trees of sublevel sets (Zhou and Wong 2008) from samples generated by the MD sampler. Since samples have been partitioned

into domains of attraction, we only need to determine the barrier between a pair of domains, defined by $\max_{s \in S} \min_{\mathbf{x} \in s} p(\mathbf{x})$, where S is the collection of all the paths between the two domains. A few candidate approaches toward this direction are under current investigation (Zhou 2011).

APPENDIX: THEORETICAL ANALYSIS

In this appendix we establish the convergence and ergodicity properties of the MD sampler. Our analysis is conducted for a doubly adaptive MCMC, that is, both the target distribution and the proposal may change along the iteration, which includes the MD sampler as a special case. Furthermore, the MWL design (Routine 1) is employed to adjust γ_t.

Assume that the sample space $\mathcal{X}$ is equipped with a countably generated σ-field, $\mathcal{B}(\mathcal{X})$. Let $\{\mathcal{X}_i\}_{i=1}^{\kappa}$ be a partition of $\mathcal{X}$, where each $\mathcal{X}_i$ is nonempty, and $\mathcal{B}(\mathcal{X}_i) = \{A \in \mathcal{B}(\mathcal{X}): A \subseteq \mathcal{X}_i\}$ for $i = 1, \ldots, \kappa$. Let $\boldsymbol{\omega} = (\omega_i)_{1:\kappa} \in \Omega$ and $\boldsymbol{\phi} \in \Phi$ be two vectors of real parameters. Denote the product parameter space by $\Theta = \Omega \times \Phi$ and write $\boldsymbol{\theta} = (\boldsymbol{\omega}, \boldsymbol{\phi}) \in \Theta$. For $\boldsymbol{\omega} \in \Omega$, define working density

$$p_{\boldsymbol{\omega}}(\mathbf{x}) \propto \sum_{i=1}^{\kappa} e^{-\omega_i} p(\mathbf{x}) \mathbf{1}(\mathbf{x} \in \mathcal{X}_i). \qquad (A.1)$$

Let $q(\mathbf{x}, \cdot)$ and $t_{\boldsymbol{\phi}}(\mathbf{x}, \cdot)$, $\boldsymbol{\phi} \in \Phi$, be two transition kernels on $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$. Hereafter, the same notation will be used for a kernel and its density with respect to the Lebesgue measure on $\mathcal{X}$, for example, $q(\mathbf{x}, d\mathbf{y}) \equiv q(\mathbf{x}, \mathbf{y}) \, d\mathbf{y}$. For $j = 0, 1$, define $Q_{j,\boldsymbol{\phi}}(\mathbf{x}, \cdot) = (1-j)q(\mathbf{x}, \cdot) + jt_{\boldsymbol{\phi}}(\mathbf{x}, \cdot)$. Let $K_{j,\boldsymbol{\theta}}$ be the MH transition kernel with $p_{\boldsymbol{\omega}}$ as the target distribution and $Q_{j,\boldsymbol{\phi}}$ as the proposal, that is,

$$K_{j,\boldsymbol{\theta}}(\mathbf{x}, d\mathbf{y}) = S_{j,\boldsymbol{\theta}}(\mathbf{x}, d\mathbf{y}) + \mathbf{1}(\mathbf{x} \in d\mathbf{y}) \left[1 - \int_{\mathcal{X}} S_{j,\boldsymbol{\theta}}(\mathbf{x}, d\mathbf{z}) \right],$$

where $S_{j,\boldsymbol{\theta}}(\mathbf{x}, d\mathbf{y}) = Q_{j,\boldsymbol{\phi}}(\mathbf{x}, d\mathbf{y}) \min[1, p_{\boldsymbol{\omega}}(\mathbf{y})Q_{j,\boldsymbol{\phi}}(\mathbf{y}, \mathbf{x})/p_{\boldsymbol{\omega}}(\mathbf{x})Q_{j,\boldsymbol{\phi}}(\mathbf{x}, \mathbf{y})]$, representing an accepted move. As $Q_{0,\boldsymbol{\phi}} = q$, $K_{0,\boldsymbol{\theta}}$ and $S_{0,\boldsymbol{\theta}}$ do not depend on $\boldsymbol{\phi}$ and thus reduce to $K_{0,\boldsymbol{\omega}}$ and $S_{0,\boldsymbol{\omega}}$, respectively. Furthermore, if we let $\boldsymbol{\omega} = \mathbf{0}$, then $p_{\boldsymbol{\omega}}(\mathbf{x}) = p(\mathbf{x})$, in which case we simply use K_0 and S_0. Given $\alpha \in [0, 1)$, define a mixture proposal $Q_{\boldsymbol{\phi}} = (1-\alpha)q + \alpha t_{\boldsymbol{\phi}}$, its accepted move $S_{\boldsymbol{\theta}} = (1-\alpha)S_{0,\boldsymbol{\omega}} + \alpha S_{1,\boldsymbol{\theta}}$, and the corresponding MH kernel $K_{\boldsymbol{\theta}} = (1-\alpha)K_{0,\boldsymbol{\omega}} + \alpha K_{1,\boldsymbol{\theta}}$. Table A.1 summarizes the notations, from left to right, for target distributions, proposals, MH kernels, and accepted moves for different scenarios involved in this analysis.

Denote by $Z(\boldsymbol{\omega})$ the normalizing constant of (A.1). Then $Z(\boldsymbol{\omega}) = \sum_{i=1}^{\kappa} Z_i(\omega_i)$, where $Z_i(\omega_i) = e^{-\omega_i} \int_{\mathcal{X}_i} p(\mathbf{x}) \, d\mathbf{x}$. Let U be a nonempty subset of $\{1, \ldots, \kappa\}$ and $\mathcal{X}_U = \bigcup_{i \in U} \mathcal{X}_i$. Given a map $\mathbf{g}: \mathcal{X} \to \Phi$, let $\boldsymbol{\mu}_{\mathbf{g}, U}(\boldsymbol{\omega}) = E[\mathbf{g}(\mathbf{X}) \mid \mathbf{X} \in \mathcal{X}_U]$ with respect to $p_{\boldsymbol{\omega}}$. Define a map $\mathbf{H}: \Theta \times \mathcal{X} \to \Theta$ by

$$\mathbf{H}(\boldsymbol{\theta}, \mathbf{x}) = \left[\left(\mathbf{1}(\mathbf{x} \in \mathcal{X}_i) - 1/\kappa \right)_{1:\kappa}, (\mathbf{g}(\mathbf{x}) - \boldsymbol{\phi}) \mathbf{1}(\mathbf{x} \in \mathcal{X}_U) \right]$$

and the mean field $\mathbf{F}(\boldsymbol{\theta}) = \int_{\mathcal{X}} \mathbf{H}(\boldsymbol{\theta}, \mathbf{x}) p_{\boldsymbol{\omega}}(\mathbf{x}) \, d\mathbf{x}$, that is,

$$\mathbf{F}(\boldsymbol{\theta}) = \left[\left(\frac{Z_i(\omega_i)}{Z(\boldsymbol{\omega})} - \frac{1}{\kappa} \right)_{1:\kappa}, \frac{\sum_{u \in U} Z_u(\omega_u)}{Z(\boldsymbol{\omega})} (\boldsymbol{\mu}_{\mathbf{g}, U}(\boldsymbol{\omega}) - \boldsymbol{\phi}) \right].$$

Consider the equation $\mathbf{F}(\boldsymbol{\theta}) = \mathbf{0}$. Let $\boldsymbol{\omega}^* = [\log Z_i(0)]_{1:\kappa}$ and $\boldsymbol{\phi}^* = \boldsymbol{\mu}_{\mathbf{g}, U}(\boldsymbol{\omega}^*)$. As $\mathbf{F}(\boldsymbol{\theta})$ is invariant to translation of $\boldsymbol{\omega}$ by a scalar: $\boldsymbol{\omega} \to (\boldsymbol{\omega} + \beta) \triangleq (\omega_i + \beta)_{1:\kappa}$, $\beta \in \mathbb{R}$, the solution set to this equation is

$\{\boldsymbol{\theta}^*(\beta) \triangleq (\boldsymbol{\omega}^* + \beta, \boldsymbol{\phi}^*)\} \cap \Theta \triangleq \Theta^*$. Set $\gamma_1 = 1$ and choose an arbitrary point $(\tilde{\mathbf{x}}, \tilde{\boldsymbol{\theta}}) \in \mathcal{X} \times \Theta$ to initialize $(\mathbf{X}^1, \boldsymbol{\theta}^1)$. A doubly adaptive MCMC is employed to find a solution $\boldsymbol{\theta}^*(\beta) \in \Theta^*$ and to estimate $\mu_h(\boldsymbol{\omega}^*) = \int_{\mathcal{X}} h(\mathbf{x}) p_{\boldsymbol{\omega}^*}(\mathbf{x}) \, d\mathbf{x}$ for a function $h: \mathcal{X} \to \mathbb{R}$.

Algorithm A.1 (Doubly adaptive MCMC). Choose a fixed $\alpha \in [0, 1)$. For $t = 1, \ldots, n$:

1. If $\boldsymbol{\theta}^t \notin \Theta$, set $\mathbf{X}^{t+1} = \tilde{\mathbf{x}}$ and $\boldsymbol{\theta}^{t+1} = \tilde{\boldsymbol{\theta}}$; otherwise draw $\mathbf{X}^{t+1} \sim K_{\boldsymbol{\theta}^t}(\mathbf{X}^t, \cdot)$ and set $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t + \gamma_t \mathbf{H}(\boldsymbol{\theta}^t, \mathbf{X}^{t+1})$.
2. Determine γ_{t+1} by the MWL design in Routine 1 with $\{\mathcal{X}_i\}$ in place of $\{D_{kj}\}$.

Denote the L_2 norm by $|\cdot|$ and let $d(\mathbf{x}, A) \triangleq \inf_{\mathbf{y} \in A} |\mathbf{x} - \mathbf{y}|$, where $\mathbf{x}, \mathbf{y}$ are vectors and A is a set. Our goal is to establish that $d(\boldsymbol{\theta}^n, \Theta^*) \to 0$ almost surely (with respect to the probability measure of the process $\{\mathbf{X}^t, \boldsymbol{\theta}^t\}$) and that $\{\mathbf{X}^t\}$ is ergodic. Clearly, translation of $\boldsymbol{\omega}^t$ by a scalar does not change the working density $p_{\boldsymbol{\omega}^t}$ or affect the convergence of $\boldsymbol{\theta}^t$ to Θ^*. Thus, the theory for Algorithm A.1 can be applied to the MD sampler with reinitialization (Remark 2). The update of $\boldsymbol{\omega}^t$, up to translation by a scalar, and the update of $\boldsymbol{\phi}^t$ correspond to, respectively, the update of $\mathbf{W}^t$ (7) and the update of $\mathbf{V}_k^t$ (8) for any k in the MD sampler. We state four conditions for establishing the main results.

(C1) The sample space $\mathcal{X}$ is compact, $p(\mathbf{x}) > 0$ for all $\mathbf{x} \in \mathcal{X}$, Θ is bounded, and Θ^* is nonempty. The map $\mathbf{g}$ and the function h are p-integrable and bounded.
(C2) There exist $\delta_q > 0$ and $\epsilon_q > 0$ such that $|\mathbf{x} - \mathbf{y}| \leq \delta_q$ implies that $q(\mathbf{x}, \mathbf{y}) \geq \epsilon_q$ for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$.
(C3) There exist an integer ℓ, $\delta > 0$ and a probability measure ν, such that $\nu(\mathcal{X}_i) > 0$ for $i = 1, \ldots, \kappa$ and $S_0^{\ell}(\mathbf{x}, A) \geq \delta \nu(A)$, $\forall \mathbf{x} \in \mathcal{X}$ and $A \in \mathcal{B}(\mathcal{X})$.
(C4) For all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ and all $\boldsymbol{\phi} \in \Phi$, $t_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{y}) > 0$ and $\log t_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{y})$ has continuous partial derivatives with respect to all the components of $\boldsymbol{\phi}$.

To avoid mathematical complexity, we assume that $\mathcal{X}$ is compact (C1). This assumption does not lose much generality in practice as we may always restrict the sample space to $\{\mathbf{x}: p(\mathbf{x}) \geq \epsilon_p\}$ given a sufficiently small ϵ_p. Due to the compactness of $\mathcal{X}$, any continuous map and function on $\mathcal{X}$ will be bounded. Conditions (C2), (C3) are standard conditions on the fixed proposal $q(\mathbf{x}, \mathbf{y})$ to guarantee irreducibility and aperiodicity of the MH kernel K_0. They are satisfied by all the local moves used in this article. A regularity condition on the adaptive proposal $t_{\boldsymbol{\phi}}$ is specified in (C4). For the mixed jumps in the examples, $\boldsymbol{\phi}$ is either the covariance matrix of a multivariate normal distribution or the cell probability vector of a multinomial distribution, and (C4) is satisfied.

Lemma A.1. Let $\alpha \in [0, 1)$. For any $i, j \in \{1, \ldots, \kappa\}$ and $\boldsymbol{\theta} \in \Theta$, if $e^{\omega_i - \omega_j} \geq c_1 \in (0, 1]$, then $K_{\boldsymbol{\theta}}(\mathbf{x}, A) \geq (1-\alpha)c_1 S_0(\mathbf{x}, A)$, $\forall \mathbf{x} \in \mathcal{X}_i$ and $A \in \mathcal{B}(\mathcal{X}_j)$.

Proof. By definition, $K_{\boldsymbol{\theta}}(\mathbf{x}, A) \geq (1-\alpha)K_{0,\boldsymbol{\omega}}(\mathbf{x}, A) \geq (1-\alpha) \times S_{0,\boldsymbol{\omega}}(\mathbf{x}, A)$ for every $\boldsymbol{\theta} = (\boldsymbol{\omega}, \boldsymbol{\phi})$. For any $\mathbf{x} \in \mathcal{X}_i$ and $\mathbf{y} \in \mathcal{X}_j$,

$$S_{0,\boldsymbol{\omega}}(\mathbf{x}, d\mathbf{y}) = q(\mathbf{x}, d\mathbf{y}) \min \left[1, e^{\omega_i - \omega_j} \frac{p(\mathbf{y})q(\mathbf{y}, \mathbf{x})}{p(\mathbf{x})q(\mathbf{x}, \mathbf{y})} \right]$$

$$\geq c_1 q(\mathbf{x}, d\mathbf{y}) \min \left[1, \frac{p(\mathbf{y})q(\mathbf{y}, \mathbf{x})}{p(\mathbf{x})q(\mathbf{x}, \mathbf{y})} \right] = c_1 S_0(\mathbf{x}, d\mathbf{y}).$$

Thus, $K_{\boldsymbol{\theta}}(\mathbf{x}, A) \geq (1-\alpha)c_1 S_0(\mathbf{x}, A)$ for all $A \in \mathcal{B}(\mathcal{X}_j)$.

Theorem A.1. If (C1)–(C4) hold, then $d(\boldsymbol{\theta}^n, \Theta^*) \xrightarrow{\text{a.s.}} 0$ and

$$\frac{1}{n} \sum_{t=1}^{n} h(\mathbf{X}^t) \xrightarrow{\text{a.s.}} \mu_h(\boldsymbol{\omega}^*), \quad \text{as } n \to \infty. \qquad (A.2)$$

Table A.1. Summary of notations

Mixture:	$p_{\boldsymbol{\omega}}$	$Q_{\boldsymbol{\phi}}$	$K_{\boldsymbol{\theta}}$	$S_{\boldsymbol{\theta}}$
$j = 0$:	$p_{\boldsymbol{\omega}}$	q	$K_{0,\boldsymbol{\omega}}$	$S_{0,\boldsymbol{\omega}}$
$j = 0, \boldsymbol{\omega} = \mathbf{0}$:	p	q	K_0	S_0
$j = 1$:	$p_{\boldsymbol{\omega}}$	$t_{\boldsymbol{\phi}}$	$K_{1,\boldsymbol{\theta}}$	$S_{1,\boldsymbol{\theta}}$

Proof. Lemma A.1 and (C3) imply that $\forall \mathbf{x} \in \mathcal{X}_i$, $K_{\boldsymbol{\theta}}^{\ell}(\mathbf{x}, \mathcal{X}_j) \geq \epsilon_{\nu} \nu(\mathcal{X}_j) > 0$ if $e^{\omega_i - \omega_j} \geq c_1$, where $\epsilon_{\nu} > 0$. By Theorem 4.2 of the article by Atchadé and Liu (2010),

$$\max_{i,j} \limsup_{n \to \infty} |\nu_i^n - \nu_j^n| < \infty, \quad \text{a.s.,}$$

where $\nu_i^n = \sum_{t=1}^{n} \mathbf{1}(\mathbf{X}^t \in \mathcal{X}_i)$. This implies that the $\{\gamma_t\}$ defined by the MWL update (Routine 1) will decrease below any given $\epsilon_{\gamma} > 0$ after a finite number of iterations, that is, $t_c < \infty$, almost surely. Then, Algorithm A.1 becomes a stochastic approximation algorithm with a deterministic sequence of $\{\gamma_t\}$. According to proposition 6.1 and theorem 5.5 in the article of Andrieu, Moulines, and Priouret (2005), we only need to verify the drift conditions (DRI1–3) and assumptions (A1), (A4) given in that work to show the convergence of $\boldsymbol{\theta}^n$.

Verifying the drift conditions. Let $\mathcal{D}$ be any compact subset of Θ, and $\mathcal{D}_1$ and $\mathcal{D}_2$ be the projections of $\mathcal{D}$ into Ω and Φ, respectively. Since $\mathcal{D}_1$ is compact, there is an $\epsilon_{\mathcal{D}} \in (0, 1]$ such that $\min_{i,j} \inf_{\boldsymbol{\omega} \in \mathcal{D}_1} e^{\omega_i - \omega_j} \geq \epsilon_{\mathcal{D}}$. By Lemma A.1 and (C3), there is a $\delta_{\mathcal{D}} > 0$ such that

$$\inf_{\boldsymbol{\theta} \in \mathcal{D}} K_{\boldsymbol{\theta}}^{\ell}(\mathbf{x}, A) \geq \delta_{\mathcal{D}} \nu(A), \qquad \forall \mathbf{x} \in \mathcal{X}, A \in \mathcal{B}(\mathcal{X}), \qquad (\text{A.3})$$

where ℓ and ν are defined in (C3). This gives the minorization condition in (DRI1). Given (C2) and that $p_{\boldsymbol{\omega}}$, $\boldsymbol{\omega} \in \mathcal{D}_1$, is bounded away from 0 and ∞ under (C1), $K_{0,\boldsymbol{\omega}}$ is irreducible and aperiodic for every $\boldsymbol{\omega} \in \mathcal{D}_1$, according to theorem 2.2 of the article by Roberts and Tweedie (1996). Consequently, for every $\boldsymbol{\theta} \in \mathcal{D}$, $K_{\boldsymbol{\theta}}$ is also irreducible and aperiodic as $\alpha < 1$. Let $V(\mathbf{x}) = 1$ for all $\mathbf{x} \in \mathcal{X}$. It is then easy to verify other conditions in (DRI1).

Since both $\mathbf{g}$ and Θ are bounded (C1), there is a $c_2 > 0$ such that for all $\mathbf{x} \in \mathcal{X}$,

$$\sup_{\boldsymbol{\theta} \in \Theta} |\mathbf{H}(\boldsymbol{\theta}, \mathbf{x})| \leq \kappa + |\mathbf{g}(\mathbf{x})| + \sup_{\boldsymbol{\phi} \in \Phi} |\boldsymbol{\phi}| \leq c_2, \qquad (\text{A.4})$$

$$|\mathbf{H}(\boldsymbol{\theta}, \mathbf{x}) - \mathbf{H}(\boldsymbol{\theta}', \mathbf{x})| \leq |\boldsymbol{\phi} - \boldsymbol{\phi}'| \leq c_2 |\boldsymbol{\theta} - \boldsymbol{\theta}'|, \qquad \forall \boldsymbol{\theta}, \boldsymbol{\theta}' \in \Theta. \quad (\text{A.5})$$

These two inequalities imply (DRI2) with $V(\mathbf{x}) = 1$.

Condition (DRI3) can be verified by the same argument used by Liang, Liu, and Carroll (2007) once we find a constant $c_3 > 0$ such that

$$\left| \frac{\partial S_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y})}{\partial \theta_i} \right| \leq c_3 Q_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{y}), \qquad (\text{A.6})$$

for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$, $\boldsymbol{\theta} \in \mathcal{D}$, and all i, where θ_i is the ith component of $\boldsymbol{\theta} = (\boldsymbol{\omega}, \boldsymbol{\phi})$. Denote by ϕ_j the jth component of $\boldsymbol{\phi}$. Straightforward calculation leads to $|\partial S_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}) / \partial \omega_i| \leq Q_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{y})$ and

$$\frac{\partial S_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y})}{\partial \phi_j} = \begin{cases} R_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y})[\partial \log t_{\boldsymbol{\phi}}(\mathbf{y}, \mathbf{x}) / \partial \phi_j] \alpha t_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{y}), \\ \quad \text{if } R_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}) < 1 \\ [\partial \log t_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{y}) / \partial \phi_j] \alpha t_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{y}), \\ \quad \text{otherwise,} \end{cases}$$

where $R_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}) = p_{\boldsymbol{\omega}}(\mathbf{y}) t_{\boldsymbol{\phi}}(\mathbf{y}, \mathbf{x}) / p_{\boldsymbol{\omega}}(\mathbf{x}) t_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{y})$. Condition (C4) with the compactness of $\mathcal{D}_2$ and $\mathcal{X}$ guarantees that

$$\sup_{\mathbf{x}, \mathbf{y} \in \mathcal{X}} \sup_{\boldsymbol{\phi} \in \mathcal{D}_2} |\partial \log t_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{y}) / \partial \phi_j| < \infty.$$

As $\alpha t_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{y}) \leq Q_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{y})$, (A.6) holds and (DRI3) is verified.

Verifying assumptions (A1), (A4). It is assumed in assumption (A1) the existence of a global Lyapunov function for $\mathbf{F}(\boldsymbol{\theta})$. Let $L(\boldsymbol{\theta}) = \frac{c_4}{2} \sum_{i=1}^{\kappa} (Z_i(\omega_i) - \bar{Z}(\boldsymbol{\omega}))^2 + \frac{1}{2} |\boldsymbol{\phi} - \boldsymbol{\mu}_{\mathbf{g}, U}(\boldsymbol{\omega})|^2$, where $\bar{Z}(\boldsymbol{\omega}) = Z(\boldsymbol{\omega}) / \kappa$. Using straightforward algebra one can show that

$$-Z \langle \nabla L, \mathbf{F} \rangle = \sum_{i=1}^{\kappa} c_4 Z_i \cdot (\Delta Z_i)^2$$

$$+ (\Delta \boldsymbol{\phi})^{\mathsf{T}} \sum_{u \in U} \Delta Z_u \frac{\partial \boldsymbol{\mu}_{\mathbf{g}, U}(\boldsymbol{\omega})}{\partial \omega_u} + \sum_{u \in U} Z_u |\Delta \boldsymbol{\phi}|^2,$$

where $\Delta Z_i = Z_i(\omega_i) - \bar{Z}(\boldsymbol{\omega})$, $\Delta \boldsymbol{\phi} = \boldsymbol{\phi} - \boldsymbol{\mu}_{\mathbf{g}, U}(\boldsymbol{\omega})$, and the arguments ($\omega_i$ and $\boldsymbol{\omega}$) in $Z_i(\omega_i)$ and $Z(\boldsymbol{\omega})$ have been dropped. Since Θ is bounded, $Z_i(\omega_i) > \epsilon_{\Omega} > 0$ for all i. Because $\mathbf{g}$ is p-integrable, $\boldsymbol{\mu}_{\mathbf{g}, i} \triangleq \boldsymbol{\mu}_{\mathbf{g}, \{i\}}(\mathbf{0})$ is bounded for all i and

$$\left| \frac{\partial \boldsymbol{\mu}_{\mathbf{g}, U}(\boldsymbol{\omega})}{\partial \omega_u} \right| \leq |\boldsymbol{\mu}_{\mathbf{g}, u}| + \max_{i \in U} |\boldsymbol{\mu}_{\mathbf{g}, i}| \leq 2 \max_{1 \leq i \leq \kappa} |\boldsymbol{\mu}_{\mathbf{g}, i}| < \infty.$$

Thus, choosing a sufficiently large c_4 ensures that $\langle \nabla L(\boldsymbol{\theta}), \mathbf{F}(\boldsymbol{\theta}) \rangle \leq 0$ for any $\boldsymbol{\theta} \in \Theta$ with equality if and only if $\boldsymbol{\theta} \in \Theta^*$. Furthermore, $\{\boldsymbol{\theta} \in \Theta : L(\boldsymbol{\theta}) \leq C_L\}$ is compact for some $C_L > 0$ and the closure of $L(\Theta^*)$ has an empty interior. Thus, all the conditions in assumption (A1) are satisfied. Since $|\boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^t| \leq \gamma_t \sup_{\boldsymbol{\theta}, \mathbf{x}} |\mathbf{H}(\boldsymbol{\theta}, \mathbf{x})| \leq c_2 \gamma_t$ (A.4) and $\gamma_t = 1/(t + \xi)$ for $t > t_c$, verifying assumption (A4) is immediate. This completes the proof of the convergence of $\boldsymbol{\theta}^n$.

The result (A.2) can be established similarly as the proof of proposition 6.2 in the work of Atchadé and Liu (2010). We only give an outline here. The drift conditions imply that for any $\boldsymbol{\theta} \in \mathcal{D}$, there exist $h_{\boldsymbol{\theta}}(\mathbf{x})$, $c_5 > 0$, and $b \in (0, 1]$ such that $h_{\boldsymbol{\theta}} - K_{\boldsymbol{\theta}} h_{\boldsymbol{\theta}} = h - \mu_h(\boldsymbol{\omega})$ and

$$\sup_{\boldsymbol{\theta} \in \mathcal{D}} (\|h_{\boldsymbol{\theta}}\| + \|K_{\boldsymbol{\theta}} h_{\boldsymbol{\theta}}\|) < \infty,$$

$$\|h_{\boldsymbol{\theta}} - h_{\boldsymbol{\theta}'}\| + \|K_{\boldsymbol{\theta}} h_{\boldsymbol{\theta}} - K_{\boldsymbol{\theta}'} h_{\boldsymbol{\theta}'}\| < c_5 |\boldsymbol{\theta} - \boldsymbol{\theta}'|^b, \qquad \forall \boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathcal{D},$$

where $K_{\boldsymbol{\theta}} h_{\boldsymbol{\theta}}(\mathbf{x}) = \int_{\mathcal{X}} K_{\boldsymbol{\theta}}(\mathbf{x}, d\mathbf{y}) h_{\boldsymbol{\theta}}(\mathbf{y})$ and for $f : \mathcal{X} \to \mathbb{R}$, $\|f\| = \sup_{\mathbf{x} \in \mathcal{X}} |f(\mathbf{x})|$. See Proposition 6.1 and assumption (A3) in the work of Andrieu, Moulines, and Priouret (2005). Then, following an essentially identical proof to that of lemma 6.6 in the work of Atchadé and Liu (2010), we can show that $\sum_{t=1}^{\infty} t^{-1}[h(\mathbf{X}^{t+1}) - \mu_h(\boldsymbol{\omega}^t)]$ has a finite limit almost surely. Since $\mu_h(\boldsymbol{\omega}^t) \xrightarrow{\text{a.s.}} \mu_h(\boldsymbol{\omega}^*)$ as $t \to \infty$, Kronecker's lemma applied to the above infinite sum leads to the desired result.

SUPPLEMENTARY MATERIALS

Supplement: The supplementary document contains (1) more discussion on the weighted estimator in Section 3.3 and (2) data simulation details in Section 5.3. (MultiDomainSuppl_R5.pdf)

REFERENCES

Andrieu, C., and Moulines, E. (2006), "On the Ergodicity Properties of Some Adaptive MCMC Algorithms," *Annals of Applied Probability*, 16, 1462–1505. [1320]

Andrieu, C., Moulines, E., and Priouret, P. (2005), "Stability of Stochastic Approximation Under Verifiable Conditions," *SIAM Journal of Control and Optimization*, 44, 283–312. [1320,1329]

Atchadé, Y. F., and Liu, J. S. (2010), "The Wang–Landau Algorithm in General State Spaces: Applications and Convergence Analysis," *Statistica Sinica*, 20, 209–233. [1317,1320,1321,1329]

Becker, O. M., and Karplus, M. (1997), "The Topology of Multidimensional Potential Energy Surface: Theory and Application to Peptide Structure and Kinetics," *Journal of Chemical Physics*, 106, 1495–1517. [1327]

Chen, M. H., Shao, Q., and Ibrahim, J. G. (2001), *Monte Carlo Methods in Bayesian Computation*. New York: Springer. [1317]

Cooper, G. F., and Herskovits, E. (1992), "A Bayesian Method for the Induction of Probabilistic Networks From Data," *Machine Learning*, 9, 309–347. [1323]

Cooper, G. F., and Yoo, C. (1999), "Causal Discovery From a Mixture of Experimental and Observational Data," in *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*, San Francisco, CA: Morgan Kaufmann, pp. 116–125. [1323]

Ellis, B., and Wong, W. H. (2008), "Learning Causal Bayesian Network Structures From Experimental Data," *Journal of the American Statistical Association*, 103, 778–789. [1318,1323,1326]

Friedman, N., and Koller, D. (2003), "Being Bayesian About Network Structure: A Bayesian Approach to Structure Discovery in Bayesian Networks," *Machine Learning*, 50, 95–126. [1318]

Gelfand, A. E., and Smith, A. F. M. (1990), "Sampling-Based Approaches to Calculating Marginal Densities," *Journal of the American Statistical Association*, 85, 398–409. [1317]

Geman, S., and Geman, D. (1984), "Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, 721–741. [1317]

Geyer, C. J. (1991), "Markov Chain Monte Carlo Maximum Likelihood," in *Computing Science and Statistics: Proc. 23rd Symp. Interface*, ed. E. M. Keramidas, Fairfax Station, VA, pp. 156–163. [1319]

Gilks, W. R., Roberts, R. O., and George, E. I. (1994), "Adaptive Direction Sampling," *The Statistician*, 43, 179–189. [1327]

Gordon, V. S., and Whitley, D. (1993), "Serial and Parallel Genetic Algorithms as Function Optimizers," in *Proc. 5th Int. Conf. Genetic Algorithms*, San Mateo, CA, pp. 177–183. [1322]

Haario, H., Saksman, E., and Tamminen, J. (2001), "An Adaptive Metropolis Algorithm," *Bernoulli*, 7, 223–242. [1319]

Hastings, W. K. (1970), "Monte Carlo Sampling Methods Using Markov Chains and Their Applications," *Biometrika*, 57, 97–109. [1317]

Hindley, A., and Kolch, W. (2007), "Raf-1 and B–Raf Promote Protein Kinase C θ Interaction With BAD," *Cellular Signalling*, 19, 547–555. [1326]

Holland, P. W. (1988), "Causal Inference, Path Analysis, and Recursive Structural Equations Models," in *Sociological Methodology*, ed. C. C. Clogg, Washington, DC: American Sociological Association, pp. 449–484. [1323]

Li, Z., and Scheraga, H. A. (1987), "Monte Carlo-Minimization Approach to the Multiple-Minima Problem in Protein Folding," *Proceedings of the National Academy of Sciences, USA*, 84, 6611–6615. [1327]

Liang, F. (2005), "Generalized Wang–Landau Algorithm for Monte Carlo Computation," *Journal of the American Statistical Association*, 100, 1311–1327. [1317]

—— (2009), "On the Use of Stochastic Approximation Monte Carlo for Monte Carlo Integration," *Statistics and Probability Letters*, 79, 581–587. [1321]

Liang, F., and Zhang, J. (2009), "Learning Bayesian Networks for Discrete Data," *Computational Statistics and Data Analysis*, 53, 865–876. [1318]

Liang, F., Liu, C., and Carroll, J. (2007), "Stochastic Approximation in Monte Carlo Computation," *Journal of the American Statistical Association*, 102, 305–320. [1317,1320,1329]

Liu, J. S. (2008), *Monte Carlo Strategies in Scientific Computing* (2nd ed.), New York: Springer. [1317]

Liu, J. S., Liang, F., and Wong, W. H. (2000), "The Multiple-Try Method and Local Optimization in Metropolis Sampling," *Journal of the American Statistical Association*, 95, 121–134. [1327]

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953), "Equations of State Calculation by Fast Computing Machines," *Journal of Chemical Physics*, 21, 1087–1091. [1317]

Neyman, J. (1990), "Sur les applications de la thar des probabilities aux experience Agaricales: Essay des principle" [English translation of excerpts by D. Dabrowska and T. Speed], *Statistical Science*, 5, 463–472. [1323]

Pearl, J. (1993), "Graphical Models: Causality and Intervention," *Statistical Science*, 8, 266–273. [1323]

—— (2000), "The Logic of Counterfactuals in Causal Inference," *Journal of the American Statistical Association*, 95, 428–435. [1323]

Roberts, G. O., and Tweedie, R. L. (1996), "Geometric Convergence and Central Limit Theorems for Multidimensional Hastings and Metropolis Algorithms," *Biometrika*, 83, 95–110. [1329]

Robins, J. (1986), "A New Approach to Causal Inference in Mortality Studies With Sustained Exposure Periods: Application to Control of the Healthy Worker Survivor Effect," *Math Modeling*, 7, 1393–1512. [1323]

—— (1987), "A Graphical Approach to the Identification and Estimation of Causal Parameters in Mortality Studies With Sustained Exposure Periods," *Journal of Chronic Diseases*, 40, 1395–1615. [1323]

Rubin, D. (1978), "Bayesian Inference for Causal Effects: The Role of Randomization," *The Annals of Statistics*, 6, 34–58. [1323]

Sachs, K., Perez, O., Peer, D., Lauffenburger, D. A., and Nolan, G. P. (2005), "Causal Protein-Signalling Networks Derived From Multiparameter Single-Cell Data," *Science*, 308, 523–529. [1325]

Spirtes, P., Glymour, C., and Scheines, R. (1993), *Causation, Prediction, and Search* (2nd ed.), Cambridge, MA: MIT Press. [1323]

Tanner, M. A., and Wong, W. H. (1987), "The Calculation of Posterior Distributions by Data Augmentation," *Journal of the American Statistical Association*, 82, 528–550. [1317]

Wang, F., and Landau, D. P. (2001), "Efficient Multiple-Range Random-Walk Algorithm to Calculate the Density of States," *Physical Review Letters*, 86, 2050–2053. [1317,1320]

Wales, D. J., and Bogdan, T. V. (2006), "Potential Energy and Free Energy Landscapes," *Journal of Physical Chemistry B*, 110, 20765–20776. [1327]

Wales, D. J., and Doye, J. P. K. (1997), "Global Optimization by Basin-Hopping and the Lowest Energy Structures of Lennard–Jones Clusters Containing up to 110 Atoms," *Journal of Physical Chemistry A*, 101, 5111–5116. [1327]

Zhou, Q. (2011), "Random Walk Over Basins of Attraction to Construct Ising Energy Landscapes," *Physical Review Letters*, 106, 180602. [1328]

Zhou, Q., and Wong, W. H. (2008), "Reconstructing the Energy Landscape of a Distribution From Monte Carlo Samples," *Annals of Applied Statistics*, 2, 1307–1331. [1327]