MC^4: a tempering algorithm for large-sample network inference

D. Barker[1,2], S. M. Hill[1,3] and S. Mukherjee[3,1]

[1] Centre for Complexity Science, University of Warwick, Coventry, U.K. CV4 7AL
[2] Department of Physics, University of Warwick, Coventry, U.K. CV4 7AL
[3] Department of Statistics, University of Warwick, Coventry, U.K. CV4 7AL

Abstract. Bayesian networks and their variants are widely used for modelling gene regulatory and protein signalling networks. In many settings, it is the underlying network structure itself that is the object of inference. Within a Bayesian framework inferences regarding network structure are made via a posterior probability distribution over graphs. However, in practical problems, the space of graphs is usually too large to permit exact inference, motivating the use of approximate approaches. An MCMC-based algorithm known as MC^3 is widely used for network inference in this setting. We argue that recent trends towards larger sample size datasets, while otherwise advantageous, can, for reasons related to concentration of posterior mass, render inference by MC^3 *harder*. We therefore exploit an approach known as parallel tempering to put forward an algorithm for network inference which we call MC^4. We show empirical results on both synthetic and proteomic data which highlight the ability of MC^4 to converge faster and thereby yield demonstrably accurate results, even in challenging settings where MC^3 fails.

1 Introduction

Modern biochemical technologies are allowing access to ever increasing amounts of data pertaining to cellular processes. As a result, there has been a move away from studying molecular components in isolation towards pathway- and network-oriented approaches. This in turn has driven much work on network models in bioinformatics, machine learning and computational statistics. Probabilistic graphical models [6, 5] have emerged as a key approach. These are stochastic models in which a graph is used to describe relationships between random variables and thereby facilitate representation and inference. Directed graphical models called Bayesian networks (BNs) are widely used in the modelling of gene regulatory and protein signalling networks [4, 1, 16, 19].

A BN consists of a directed acyclic graph (DAG) G which describes conditional independence relationships between variables, and associated parameters. In many bioinformatics settings, it is of interest to make inferences about the DAG itself, a task known as structure learning or network inference.

Within a Bayesian framework, under certain assumptions, it is possible to analytically integrate out parameters to obtain a score which is *proportional*

to the posterior probability of a given graph G. However, since the number of possible DAGs grows super-exponentially [15] it rapidly becomes infeasible to characterise the posterior distribution by simply enumerating all possible DAGs. This has motivated the use of approximate inference methods for network inference. Markov chain Monte Carlo (MCMC) methods [7, 14] in particular are often used in this setting [4, 10].

A widely-used approach is to follow [8] in using a random-walk Metropolis type algorithm in which moves are made in the state space of DAGs via single-edge changes (for details see Section 2 below). This scheme, known as MC^3 (for "Markov Chain Monte Carlo Model Composition"), is asymptotically guaranteed to converge to the desired posterior distribution, but can be slow to do so, and for large or otherwise challenging distributions can fail entirely (we show examples below).

In recent years, there has been a drive towards larger sample-size datasets for network inference. As the cost of array-based assays continues to fall, studies have become wider in scope, covering a greater number of samples. At the same time, single-cell, FACS-based platforms have also become more widely available. Clearly, this trend towards larger datasets is broadly favourable. Yet at the same time, it can render MCMC-based network inference *more* challenging. This is because as the sample size increases, the posterior mass becomes more concentrated around fewer graphs (eventually, by consistency of Bayesian model selection, around members of the correct, data-generating equivalence class). In this setting, the MC^3 scheme can have difficulty discovering these high-scoring graphs, or moving between them. Figure 1 shows an empirical example of this phenomenon. As we increase the sample-size N for a simple, four-node toy-model, the posterior becomes progressively more concentrated on a few graphs. (Note, locality of graphs along the axis in Figure 1 does not represent location in graph space).

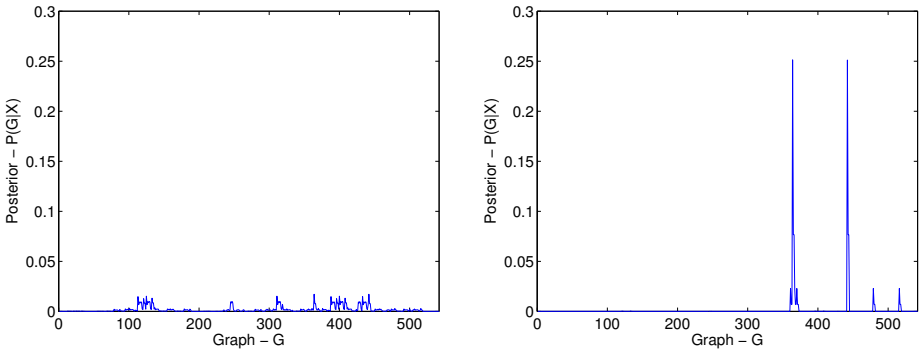


Fig. 1. The posterior distribution of graphs $P(G|\mathbf{X})$ for two lengths of data set N. As N increases the distribution becomes more peaked on a few highly probable graphs. The distributions shown here are over the space of 4-node DAGs; a space small enough to permit enumeration of the true distribution.

This 'peakiness' can be quantified by the information entropy

$$H[P(G|\mathbf{X})] = -\sum_{G\in\mathcal{G}} P(G|\mathbf{X}) \log P(G|\mathbf{X}) \tag{1}$$

where, $P(G|\mathbf{X})$ denotes the posterior probability of a graph G given data $\mathbf{X}$ and $\mathcal{G}$ is the whole space of DAGs. Using Bayes' theorem we can write the posterior as proportional to the product of a marginal likelihood, $P(\mathbf{X}|G)$ and a prior distribution over graphs, $P(G)$;

$$P(G|\mathbf{X}) \propto P(\mathbf{X}|G)P(G). \tag{2}$$

The marginal likelihood is obtained by integrating out model parameters Θ. The likelihood $P(\mathbf{X}|G,\Theta)$ factorises into a product of local terms in which each variable X_i depends only on its parents in graph G, $\pi_G(i)$, and parameters θ_i. That is, $P(\mathbf{X}|G,\Theta) = \prod_i P(X_i|\pi_G(i),\theta_i)$.

Entropy H and shape of the posterior are affected by both marginal likelihood and graph prior. The main focus of the present paper are low-entropy regimes that are characteristic of large-sample problems. We therefore focus only on the effect of increasing sample size and choose a uniform graph prior, $P(G) = |\mathcal{G}|^{-1}$.

H is maximal for a uniform distribution so its maximal possible value is $H_{\max} = \log|\mathcal{G}|$. Consider the information entropy of the distributions on 4-node DAGs shown in Figure 1; as N increases we move away from the uniform distribution which is reflected by H decreasing from $H_{\max} \simeq 6.3$ to a lower value of $\simeq 1.5$ (see Figure 2).

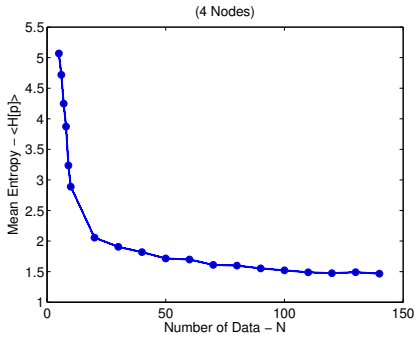


Fig. 2. The information entropy $H[p]$ (averaged over 100 subsamplings of a data set) as a function of length of the data set N. We can see that as we increase N the information entropy H decreases indicating that we are moving further away from the uniform distribution.

Low-entropy regimes of this kind, which are important for the larger datasets that are now becoming available, present special challenges for MCMC. One approach, popular in both statistical physics and Bayesian statistics, is to permit

either longer-range or, via so-called tempering algorithms, "higher temperature" moves around the state space. Here, we show how a form of tempering known as parallel tempering can be used to accelerate convergence of MCMC-based network inference. The approach has a minimum of user-set parameters and is amenable to efficient, parallel implementation, giving effective run-times identical to MC^3. We show comparative results on both synthetic data and data from high-throughput proteomics. Since tempering approaches are often referred to as "Metropolis-coupled", we call our approach "MC^4" for "Model Composition by Metropolis-Coupled Markov Chain Monte Carlo".

The remainder of the paper is organised as follows. We first introduce notation and briefly review MCMC-based network inference. We then introduce the parallel tempering scheme used and illustrate how it can help inference in the relatively low entropy regimes of interest here. We then show empirical results comparing relative performance on both synthetic and proteomic data. We conclude with a discussion of the work presented and ideas for future work.

2 Methods

2.1 Monte Carlo Schemes

The Monte Carlo schemes used here can be thought of as constructing a Markov chain whose state space is the space of DAGs $\mathcal{G}$ and whose (unique) invariant distribution is the posterior distribution $P(G|\mathbf{X})$ of interest. For a more detailed technical exposition of these ideas we refer the interested reader to [14] and references therein.

Given $t_{\max}$ samples our estimate of the probability of a graph G is given by

$$\hat{P}(G|\mathbf{X}) = \frac{1}{t_{\max}} \sum_{t=1}^{t_{\max}} I(g^{(t)} = G) \tag{3}$$

where $g^{(t)}$ is the t^{th} sampled graph and $I(\cdot)$ is the indicator function which equals 1 if its argument is true and 0 otherwise. For MCMC schemes which satisfy certain mild conditions we also have, by standard results:

$$\lim_{t_{\max}\to\infty} \frac{1}{t_{\max}} \sum_{t=1}^{t_{\max}} I(g^{(t)} = G) = P(G|\mathbf{X}). \tag{4}$$

The Markov Chain performs its walk by proposing a new graph from the state space according to some 'proposal distribution' and then subsequently accepting or rejecting the proposed graph according to an 'acceptance probability', thereby ensuring the stationary distribution is the one desired.

MC^3. Here, the proposal distribution Q involves picking so-called 'neighbours' with uniform probability. The neighbourhood $\eta(G)$ of a graph G is defined to be all graphs G' which can be obtained from G by either removing, adding or

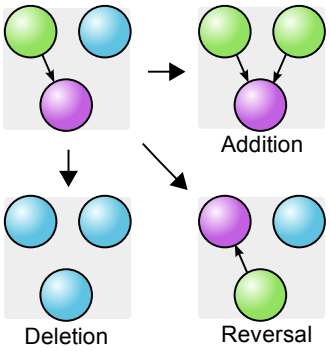


Fig. 3. The neighbourhood $\eta(G)$ of any directed acyclic graph G is defined as all acyclic graphs which are reachable from the current graph with one of the three basic edge operations; addition, deletion and reversal.

flipping a single edge, whilst maintaining acyclicity (see Figure 3). The proposal distribution is then

$$Q(G \to G') = \begin{cases} \frac{1}{|\eta(G)|} & \text{if } G' \in \eta(G) \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

This proposal distribution has only short range support. It is worth noting that this proposal distibution is not symmetric since the sizes of the neighbourhoods, $|\eta|$, are (possibly) different for G and G'. However this is accounted for by a corresponding factor in the acceptance probability below which ensures that detailed balance is satisfied. This in turn is a sufficient condition to guarantee convergence to the correct posterior distribution in the limit. Specifically, the acceptance probability has the form $A = \min\{1, \alpha\}$ where

$$\alpha = \frac{P(\mathbf{X}|G')P(G')}{P(\mathbf{X}|G)P(G)} \frac{Q(G' \to G)}{Q(G \to G')} = \frac{P(\mathbf{X}|G')|\eta(G)|}{P(\mathbf{X}|G)|\eta(G')|}. \tag{6}$$

Here, the prior terms $P(G)$ and $P(G')$ cancel since we are using a uniform prior.

MC^4. In light of the concerns with MC^3 highlighted in the Introduction above, a natural idea is to consider "higher temperature" moves; a strategy which is widely used in statistical physics. Here, we use an approach known as Parallel Tempering (PT) to this end. PT is an MCMC-approach which aims to help the Markov chain escape local maxima, thus aiding mixing [7, 12]. PT is a natural progression from work done by Marinari and Parisi [9] and Geyer and Thompson [3] on so-called Simulated Tempering (ST). In this statistical application there is no equivalent to the physical temperature, but we can introduce an analogue by writing

$$\alpha = \left(\frac{P(\mathbf{X}|G')P(G')}{P(\mathbf{X}|G)P(G)} \frac{|\eta(G)|}{|\eta(G')|} \right)^{\beta} \tag{7}$$

Here $\beta = T^{-1}$ is the inverse temperature. Clearly as $\beta \to 0$ (infinite temperature) $\alpha = 1$ and so we have the uniform distribution one would expect. Similarly as $\beta \to \infty$ (zero temperature) $\alpha = \infty$ if G' is more likely or $\alpha = 0$ if G is less likely and we recover steepest ascent.

In PT we run a collection of Markov Chains at different temperatures, occasionally swapping the graphs between them. To be more concrete, we have m chains each with an associated temperature T_i (β_i). The temperatures must obey $T_1 = 1$ ($\beta_1 = 1$) and $T_i > T_j$ ($\beta_i < \beta_j$) for $1 \leq j < i \leq m$. The algorithm for updating the chains is

(1) With probability $(1 - p_{\text{swap}})$ conduct a parallel step;
   (a) Update each graph G_i for each chain i using the MH scheme at temperature β_i.
(2) else conduct an exchange step;
   (a) Randomly choose a neighbouring pair of chains (i,j). Propose swapping their graphs G_i with G_j.
   (b) Accept the swap with probability $R = \min\{1, \rho\}$

$$\rho = \frac{(P(\mathbf{X}|G_j)P(G_j))^{\beta_i}\ (P(\mathbf{X}|G_i)P(G_i))^{\beta_j}}{(P(\mathbf{X}|G_i)P(G_i))^{\beta_i}\ (P(\mathbf{X}|G_j)P(G_j))^{\beta_j}}. \tag{8}$$

This scheme satisfies detailed balance in the extended state space thus convergence for each chain is guaranteed to the correct posterior distribution for each temperature [3, 7].

The performace of this scheme depends upon the choice of temperatures and (somewhat more weakly) upon the exchange probability p_{swap}. A guide for choosing suitable temperatures is given in [7] as

$$(\beta_i - \beta_{i+1})\,|\Delta \log P| \approx -\log p_a \tag{9}$$

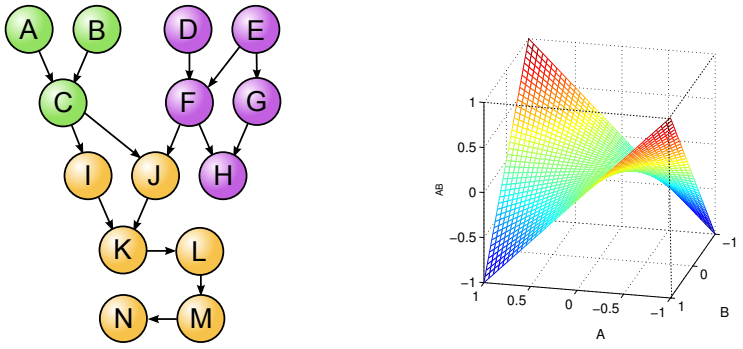
where $|\Delta \log P|$ is the typical difference in the log-posterior and p_a is the desired lower bound for the swapping acceptance probability.

Thanks to modern parallel computing facilities the updating of chains in step (1)(a) can be carried out simultaneously meaning this scheme can be run at effectively the same speed as the traditional MC^3 scheme, so long as the number of available processors is $\geq m$. If this condition is not satisfied we must wait for the chains to update before preceeding.

2.2 Simulation set-up

We run the two schemes on data simulated from the known network shown in Figure 4(a). Since we know the underlying network structure, we are able to assess and compare performance of the schemes.

Continuous data is generated by sampling the root nodes from a zero-mean Gaussian. Child nodes are also Gaussian distributed, but with mean dependent on their parents in the graph. Specifically, when a node has a single parent, the mean is simply its parent's value. If there are two parents the mean of child



(a) Data generating network.

(b) Continuous !XOR function.

Fig. 4. Simulation set-up. (a) Data generating network. (b) The cross terms in the known model (10) approximates the !XOR Boolean function and make it tough to infer the underlying structure (a).

is taken to be a non-linear combination of the parents. Thus the cumulative probabilities are defined as

$$P(A \leq x) = P(B \leq x) = \Phi\left(\frac{x}{\sigma}\right)$$

$$P(C \leq x|A,B) = \Phi\left(\frac{x-(A+B+\gamma AB)}{\sigma}\right) \tag{10}$$

for child node C with parents A and B, where $\Phi(x) = \frac{1}{2}[1+\mathrm{erf}(\frac{x}{\sqrt{2}})]$ is the cumulative distribution function of a standard Gaussian. The non-linear cross term γAB in the mean is chosen in the hope of separating the peaks of the distribution. If A is high and B is low (or vice versa) then C is low, if however A and B are both high (or low) then crucially C is high. This structure (illustrated in Figure 4) approximates the !XOR Boolean function, rendering difficult inference of the relationship between parents A and B and child C.

In order to investigate the effects of sample size N on the two schemes, we consider data sets with $N = 500$ and $N = 5,000$. We consider performance over ten MCMC runs of $t_{max} = 50,000$ iterations each; this gives good indications of convergence using standard diagnostics [2].

This paper concerns MCMC methods and the approaches we discuss apply to essentially any prior specification which yields a closed-form marginal likelihood or one which can be efficiently approximated. In all the experiments here we use a Gaussian model. Specifically, we take $X_i \sim \mathcal{N}(\mathbf{B}_i\beta_i, \sigma^2 I)$ where B_i is a local design matrix (including products over parents) and β_i corresponding regression coefficients. We use conjugate parameter priors [17, 13] $\beta_i \sim \mathcal{N}(\mathbf{0}, N\sigma^2(\mathbf{B}_i^{\mathsf{T}}\mathbf{B}_i)^{-1})$

and $\sigma^2 \propto 1/\sigma^2$ to obtain the following closed form marginal likelihood,

$$p(\mathbf{X}|G) \propto \prod_i (1+N)^{-(2^{|\pi_{G^{(i)}}|}-1)/2} \left({X_i}^\mathsf{T} X_i - \frac{N}{N+1} {X_i}^\mathsf{T} \mathbf{B}_i \left({\mathbf{B}_i}^\mathsf{T} \mathbf{B}_i \right)^{-1} {\mathbf{B}_i}^\mathsf{T} X_i \right)^{-\frac{N}{2}}. \tag{11}$$

2.3 Measuring Convergence

We are interested in the marginal posterior probabilities for individual edges. We collect these probabilities into an "edge probability matrix" $\mathbf{E}$, specifically:

$$\mathbf{E}_{ij} = \sum_{G \in \mathcal{G}} P(G|\mathbf{X}) I(e = (i,j) \in G). \tag{12}$$

We will also index entries in $\mathbf{E}$ by edge, e.g. $\mathbf{E}(e)$. Similarly to equation (3), we estimate the the edge probability of edge e by counting how many times it appears in the sampled graphs $g^{(t)}$,

$$\mathbf{E}^{\mathrm{MC}}_{ij} = \frac{1}{t_{\max}} \sum_{t=1}^{t_{\max}} I(e = (i,j) \in g^{(t)}). \tag{13}$$

If exact edge probabilities are available (as with our proteomic data study below), they can be used in tandem with our estimated edge probabilities to assess convergence. We use two measures of how well our Markov chains are converging; the correlation coefficient ρ between the exact and estimated edge probabilities and the normalised sum of absolute differences

$$S = \frac{1}{\nu} \sum_e |\mathbf{E}^{\mathrm{MC}}(e) - \mathbf{E}(e)| \tag{14}$$

where the sum runs over all possible edges and ν is the number of possible edges.

3 Results

3.1 Simulation results

We assessed performance by thresholding posterior edge probabilities to obtain a set of edges and comparing this set to the true data generating graph edge set. We constructed receiver operating characteristic (ROC) curves for the MC^3 and MC^4 schemes with sample sizes $N = 500$ and $N = 5000$. The ROC curves show the number of false positives (edges obtained after thresholding that are not in the true graph) encountered for a given number of true positives (edges obtained after thresholding that are in the true graph). The curve is parameterised by the threshold level. Figure 5 shows average ROC curves, produced by averaging ROC curves obtained from ten MCMC runs.

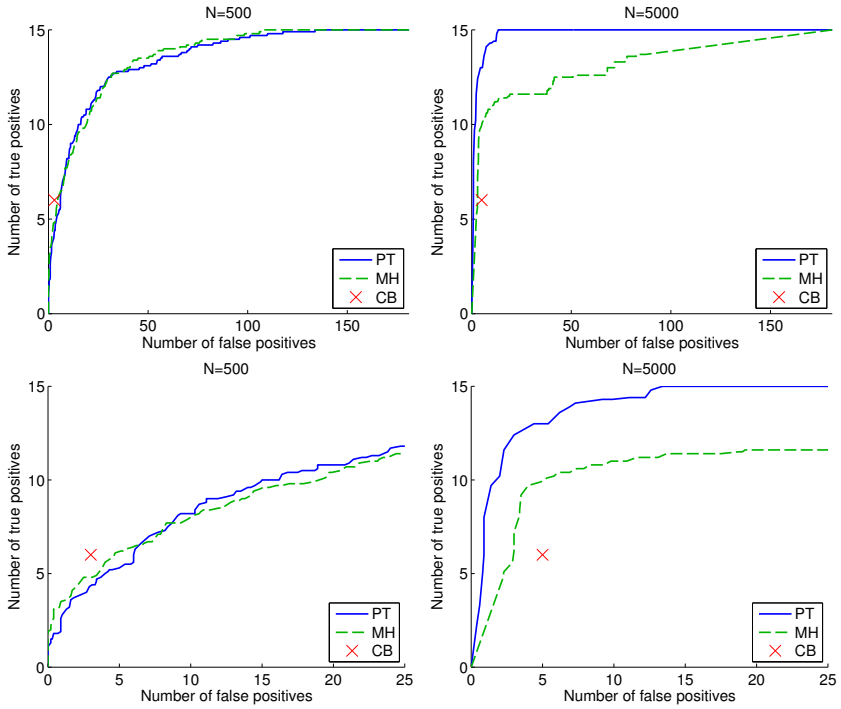


Fig. 5. Receiver operating characteristic (ROC) curves for simulated data. Number of true positive against number of false positive edge calls, produced by thresholding posterior edge probabilities at varying levels and comparing with the known, true data-generating graph. Results shown for MC^4 (blue solid line), MC^3 (green dashed line) and a recent deterministic constraint-based (CB) method for learning DAGs [18] (red cross), for sample sizes $N = 500$ (left) and $N = 5000$ (right). Lower panels show detail of corresponding upper panels. (MCMC results shown are averages over ten iterations).

The area under the ROC curve (AUC) gives a simple measure of performance. Higher AUC values indicate superior accuracy. At the smaller sample size of $N = 500$ we see that MC^4 performs comparably with MC^3. The mean AUC value ($\pm$ standard deviation) for MC^4 is 0.90 ± 0.02 compared with 0.91 ± 0.02 for MC^3. However, at the larger sample size of $N = 5000$, we see a substantial improvement of performance with the MC^4 scheme compared with MC^3. The mean AUC value for MC^4 here is 0.99 ± 0.003, whereas the mean AUC value for MC^3 has decreased to 0.88 ± 0.13. This clearly illustrates that higher sample sizes can have a deleterious effect on the MC^3 scheme, whereas the MC^4 scheme improves dramatically in performance. At smaller sample sizes, with higher entropy posterior distributions, MC^3 performs well with MC^4 not providing any real gains.

We also compared our MCMC methods to a recent deterministic, constraint-based (CB) algorithm for learning DAGs [18] (default settings, significance level

set to 0.05). For the $N = 500$ case, it performs comparably with MC^3 and MC^4 in terms of numbers of true and false positives. However, for the large-sample, $N = 5000$ case, we find that for the same number of false positives, MC^4 returns more than twice as many true positive edges as the CB algorithm.

3.2 Real Data Results

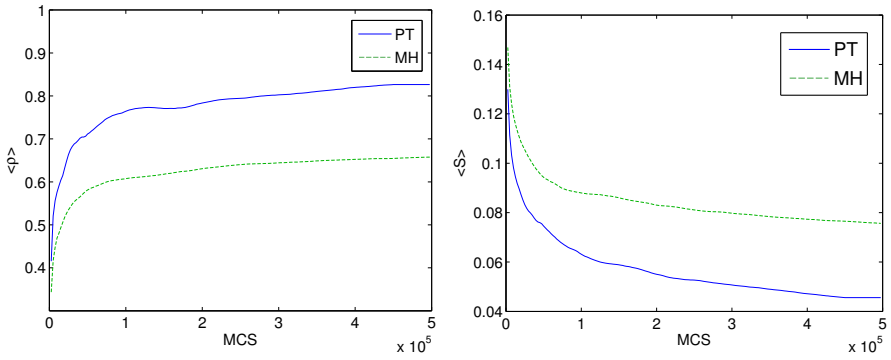


Fig. 6. Average correlation $\langle \rho \rangle$ between the exact edge probabilities and estimated edge probabilities for MC^4 and MC^3 (left) and the average per edge deviation error $\langle S \rangle$ (right). We can see from both measures that, in this real problem, MC^4 is outperforming MC^3 in terms of convergence. The parallel tempering used here had 5 temperatures evenly spaced between 1.0 and 2.0 with an exchange probability $p_{\text{swap}} = 0.1$.

To investigate the performance of MC^4 on challenging experimental data, we make use of proteomic data from an ongoing study of cell signalling (unpublished data). Here the models are dynamic Bayesian networks (DBNs) [11] with 20 nodes (and thus 400 possible edges). The true underlying networks are not known in this case. However, by taking advantage of a certain factorisation of the graph space we can, in this case, calculate the edge probabilities *exactly*. The availability of exact results enables us to properly assess the performance of the MCMC schemes for this problem. We note that in practice, in this particular setting, one should use the exact calculation rather than an MCMC estimate. However, this design provides an ideal opportunity to test the MCMC methods on a large state-space problem based on real data but with gold-standard results available for comparison.

When applied to the real proteomic data we can see that parallel tempering provides a clear advantage over Metropolis-Hastings (see Figure 6). Both measures used to assess convergence are favourable for MC^4; the correlation between the exact edge probabilities and those estimated is closer to 1 for MC^4 than MC^3 and the per edge error as quantified by S is lower for MC^4.

The scatter plots shown in Figure 7 serve to further elucidate this point. If the edges had been inferred perfectly they would lie on the line $x = y$ (representing

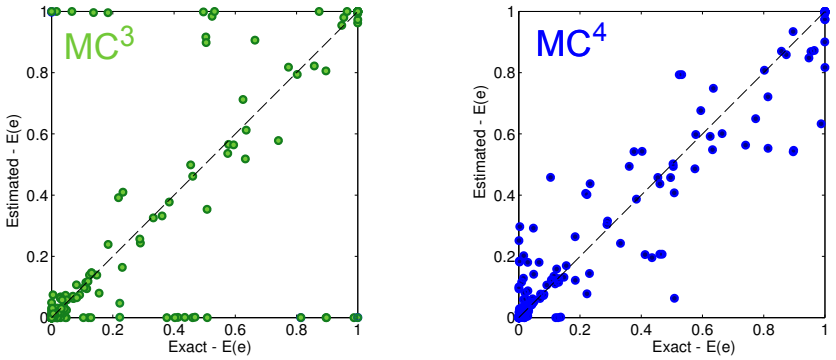


Fig. 7. Scatterplots of the exact edge probabilities versus MCMC estimated edge probabilities after $T = 500,000$ iterations for the real proteomic data for both MC^3 and MC^4 schemes. Notice that in the MC^3 case many of the edges lie in the off-diagonal corners $(0, 1)$ and $(1, 0)$, representing the most dramatic failures of the network inference. Use of the MC^4 scheme has remedied this with the offending edges in MC^3 being pulled significantly closer to the line $x = y$.

$S = 0$), the farther points lie away from this line the worse our estimate of their value is. This means that points lying in the off-diagonal corners, as seen with MC^3, represent dramatic failures of inference. We observe that MC^4 has remedied this defect.

4 Conclusions

We have argued that MCMC-based network inference from larger datasets poses special problems for the widely-used MC^3 algorithm. As experimental designs become more ambitious in scope, better MCMC approaches will become ever more crucial for robust network inference. Motivated by these concerns, and by the increasing importance of inference in larger sample size settings, we proposed a tempering-based approach to network inference which we called MC^4.

We showed that MC^4 was able to outperform MC^3 in experiments on both simulated and real data, in some cases offering dramatic gains. These results support the idea that chains at higher temperatures can help inference in the regimes of interest by moving more freely in the regions of low scoring graphs. By coupling higher temperature chains to the desired target chain at $T = 1$, using exchange moves, we allowed it to move between the peaks while sampling them with the correct frequencies. In conclusion, the MC^4 algorithm put forward here is simple, requires little user-input and is demonstrably effective for network inference.

Acknowledgements. The authors are grateful to three referees for constructive comments which improved the paper; to Robert Goudie, Mario Nicodemi and Nicholas Podd for discussions and to EPSRC for support.

References

1. Friedman, N.: Inferring cellular networks using probabilistic graphical models. Science 303(5659), 799–805 (2004)
2. Gelman, A., Rubin, D.B.: Inference from iterative simulation using multiple sequences. Stat. Sci. 7, 457–472 (1992)
3. Geyer, C., Thompson, E.: Annealing Markov Chain Monte Carlo with Applications to Ancestral Inference. Journal of the American Statistical Association 90(431), 909–920 (Sept 1995)
4. Husmeier, D.: Reverse engineering of genetic networks with Bayesian networks. Biochemical Society Transactions 31(6), 1516–8 (2003)
5. Jordan, M.: Graphical Models. Stat Sci 19, 140–155 (2004)
6. Lauritzen, S.: Graphical Models. O.U.P., Oxford, U.K. (1996)
7. Liu, J.: Monte Carlo Strategies in Scientific Computing. Springer Series in Statistics, Springer, New York, USA (2008)
8. Madigan, D., York, J., Allard, D.: Bayesian Graphical Models for Discrete Data. International Statistical Review/Revue Internationale de Statistique 63(2), 215–232 (1995)
9. Marinari, E., Parisi, G.: Simulated Tempering: a New Monte Carlo Scheme. Europhys. Lett. 19(6), 451–458 (July 1992)
10. Mukherjee, S., Speed, T.: Network Inference Using Informative Priors. PNAS 105(38), 14313–14318 (Sept 2008)
11. Murphy, K.: Dynamic Bayesian Networks: Representation, Inference and Learning. Ph.D. thesis, Computer Science Division, Berkeley CA (2002)
12. Newman, M., Barkema, G.: Monte Carlo Methods in Statistical Physics. O.U.P., Oxford, U.K. (1999)
13. Nott, D.J., Green, P.J.: Bayesian variable selection and the swendsen-wang algorithm. J. Comput. Graph. Stat. 13, 141–157 (2004)
14. Robert, C., Casella, G.: Monte Carlo Statistical Methods. Springer, New York, USA (2004)
15. Robinson, R.: Counting Labeled Acyclic Digraphs, pp. 239–273. New Directions in the Theory of Graphs, Academic Press (1973)
16. Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D.A., Nolan, G.P.: Causal protein-signaling networks derived from multiparameter single-cell data. Science 308(5721), 523–9 (2005)
17. Smith, M., Kohn, R.: Nonparametric regression using Bayesian variable selection. J. Econometrics 75, 317–343 (1996)
18. Xie, X., Geng, Z.: A recursive method for structural learning of directed acyclic graphs. J. Mach. Learn. Res. 9, 459–483 (2008)
19. Yu, J., Smith, A., Wang, P.P., Hartemink, A.J., Jarvis, E.D.: Advances to Bayesian network inference for generating causal networks from observational biological data. Bioinformatics 20(18), 3594–3603 (2004)